Genstat®

Procedure library

www.vsni.co.uk

VSNi

Genstat® Reference Manual (Release 22)

Part 3: Procedures

This Manual describes the procedures in Procedure Library PL30, which accompanies Release 22 of Genstat. The Library is controlled by an Editorial Committee composed as follows:

R.W. Payne (Chairman)
P. Brain
R.C. Butler
D.B. Baird
P.W. Goedhart
M.C. Hannah
P.W. Lane
S.D. Langton
A.R.G. McLachlan
D.A. Murray
J.M. Potts

Genstat Release 22 was developed by VSN International Ltd, in collaboration with practising statisticians at Rothamsted and other organisations in Britain, Australia, New Zealand and The Netherlands.

Published by: VSN International, 2 Amberside, Wood Lane, Hemel Hempstead, Hertfordshire HP2 4TP, UK
E-mail: info@genstat.co.uk
Website: http://www.genstat.co.uk/

First published 1989, as the *Genstat 5 Procedure Library Manual Release 1.3[2]*
This edition published 2022, for Genstat Release 22

Citation: VSN International (2022). *Genstat Reference Manual (Release 22), Part 3 Procedures*. VSN International, Hemel Hempstead, UK.

Contents

Conventions

Genstat system words are shown in the Courier typeface e.g. `CALCULATE`. In the general form of each statement, elements of the language to be substituted by the user are in italics, e.g. *variate*. New procedures in Release 21, or options and parameters of existing procedures that have been modified in Release 21, are marked by the symbol †.

Introduction

The Genstat Procedure Library contains a substantial collection of procedures which provide many useful extensions facilities provided by the standard Genstat commands (or *directives*). The Library is always attached whenever Genstat is run, and procedures are accessed automatically when required. Thus, a Library procedure can be used in exactly the same way as a Genstat directive.

The procedures have been written not only by the developers of Genstat, but also by the users from many different countries and application areas. They thus provide many useful enhancements to the standard features of Genstat. The Library has an Editorial Committee, to ensure that the procedures are all reliable, useful and well documented.

This Manual is for Release PL30 of the library which is distributed with Release 22 of Genstat. There are some changes in existing procedures in this release. The options and parameters that have been modified are marked by the symbol †. New procedures are marked in the same way. Information about the changes, and a list of new procedures, can also be obtained by running the procedure `NOTICE` with option `PRINT=release`.

Each procedure has an example which can be accessed using the `LIBEXAMPLE` procedure. These examples can easily be run to show the output that can be obtained: e.g. to run the example for the procedure `GEE` (generalized estimating equations) you can put:

```
LIBEXAMPLE 'GEE'; EXAMPLE=EXGEE
##EXGEE
```

Alternatively the example can be written to a file using the `PRINT` directive.

```
OPEN 'exgee.gen'; CHANNEL=2; FILETYPE=output
PRINT [CHANNEL=2] EXGEE; JUSTIFICATION=left; SKIP=0
```

This file can then be edited, if required, before running the example.

Examples and source code of procedures can also be accessed from the Help menu (on the Menu bar) in Genstat *for Windows*.

If you have written a Genstat procedure which you feel may be of use to others and wish to submit it to be considered for inclusion in the library, instructions for Authors can be obtained by running procedure `NOTICE` with option `PRINT=instructions`.

List of procedures in Library PL30

ABIVARIATE produces graphs and statistics for bivariate analysis of variance.
ABLUPS calculates BLUPs for block terms in an ANOVA analysis.
ABOXCOX estimates the power λ in a Box-Cox transformation, that maximizes the partial log-likelihood in ANOVA.
ACANONICAL determines the orthogonal decomposition of the sample space for a design, using an analysis of the canonical relationships between the projectors derived from two or more model formulae.
ACDISPLAY provides further output from an analysis by ACANONICAL.
ACHECK checks assumptions for an ANOVA analysis.
ACKEEP saves information from an analysis by ACANONICAL.
ACONFIDENCE calculates simultaneous confidence intervals for ANOVA means.
ADETECTION calculates the minimum size of effect or contrast detectable in an analysis of variance.
ADPOLYNOMIAL plots single-factor polynomial contrasts fitted by ANOVA.
ADSPREADSHEET puts the data and plan of an experimental design into Genstat spreadsheets.
AEFFICIENCY calculates efficiency factors for experimental designs.
AFALPHA generates alpha designs.
AFAUGMENTED forms an augmented design.
AFCARRYOVER forms factors to represent carry-over effects in cross-over trials.
AFCOVARIATES defines covariates from a model formula for ANOVA.
AFCYCLIC generates block and treatment factors for cyclic designs.
AFDISCREPANCY calculates the discrepancy of a design.
AFFYMETRIX estimates expression values for Affymetrix slides.
AFIELDRESIDUALS display residuals in field layout.
AFLABELS forms a variate of unit labels for a design.
AFMEANS forms tables of means classified by ANOVA treatment factors.
AFNONLINEAR forms D-optimal designs to estimate the parameters of a nonlinear or generalized linear model.
AFORMS prints data forms for an experimental design.
AFPREP searches for an efficient partially-replicated design.
AFRCRESOLVABLE forms doubly resolvable row-column designs, with output.
AFUNITS forms a factor to index the units of the final stratum of a design.
AGALPHA forms alpha designs by standard generators for up to 100 treatments.
AGBIB generates balanced incomplete block designs.
AGBOXBEHNKEN generates Box-Behnken designs.
AGCENTRALCOMPOSITE generates central composite designs.
AGCROSSOVERLATIN generates Latin squares balanced for carry-over effects.
AGCYCLIC generates cyclic designs from standard generators.
AGDESIGN generates generally balanced designs.
AGFACTORIAL generates minimum aberration block or fractional factorial designs.
AGFRACTION generates fractional factorial designs.
AGHIERARCHICAL generates orthogonal hierarchical designs.
AGINDUSTRIAL helps to select and generate effective designs for use in industrial experiments.
AGLATIN generates mutually orthogonal Latin squares.
AGLOOP generates loop designs e.g. for time-course microarray experiments.
AGMAINEFFECT generates designs to estimate main effects of two-level factors.
AGNATURALBLOCK forms 1- and 2-dimensional designs with blocks of natural size.
AGNEIGHBOUR generates neighbour-balanced designs.
AGNONORTHOGONALDESIGN generates non-orthogonal multi-stratum designs.
AGQLATIN generates complete and quasi-complete Latin squares.
AGRAPH plots tables of means from ANOVA.

AGREFERENCE generates reference-level designs e.g. for microarray experiments.
AGSEMILATIN generates semi-Latin squares.
AGSPACEFILLINGDESIGN generates space filling designs.
AGSQLATTICE generates square lattice designs.
AKAIKEHISTOGRAM prints histograms with improved definition of groups.
AKEY generates values for treatment factors using the design key method.
ALIAS finds out information about aliased model terms in analysis of variance.
ALIGNCURVE forms an optimal warping to align an observed series of observations with a standard series.
ALLDIFFERENCES shows all pairwise differences of values in a variate or table.
ALLPAIRWISE performs a range of all pairwise multiple comparison tests.
AMCOMPARISON performs pairwise multiple comparison tests for ANOVA means.
AMDUNNETT forms Dunnett's simultaneous confidence interval around a control.
AMERGE merges extra units into an experimental design.
AMMI allows exploratory analysis of genotype × environment interactions.
AMTDISPLAY displays further output for a multi-tiered design analysed by AMTIER.
AMTIER analyses a multi-tiered design with up to 3 structures.
AMTKEEP saves information from the analysis of a multitiered design by AMTIER.
ANTMVESTIMATE estimates missing values in repeated measurements.
ANTORDER assesses order of ante-dependence for repeated measures data.
ANTTEST calculates overall tests based on a specified order of ante-dependence.
AN1ADVICE aims to give useful advice if a design that is thought to be balanced fails to be analysed by ANOVA.
AONEWAY performs one-way analysis of variance.
AOVANYHOW performs analysis of variance using ANOVA, regression or REML as appropriate.
AOVDISPLAY provides further output from an analysis by AOVANYHOW.
APERMTEST does random permutation tests for analysis-of-variance tables.
APAPADAKIS analysis of variance with an added Papadakis covariate, formed from neighbouring residuals.
APLOT plots residuals from an ANOVA analysis.
APOLYNOMIAL forms equations for single-factor polynomial contrasts fitted by ANOVA.
APOWER calculates the power (probability of detection) for terms in an aov.
APPEND appends a list of vectors of compatible types.
APRODUCT forms a new experimental design from the product of two designs.
ARANDOMIZE randomizes and prints an experimental design.
ARCSPLITPLOT adds extra treatments onto the replicates of a resolvable row-column design, and generates factors giving the row and column locations of the plots within the design.
AREPMEASURES produces an analysis of variance for repeated measurements.
ARESULTSUMMARY provides a summary of results from an ANOVA analysis.
ARETRIEVE retrieves an ANOVA save structure from an external file.
ASAMPLESIZE finds the replication to detect a treatment effect or contrast.
ASCREEN performs screening tests for designs with orthogonal block structure.
ASPREADSHEET saves results from an analysis of variance in a spreadsheet.
ASTATUS provides information about the settings of ANOVA models and variates.
ASTORE stores an ANOVA save structure in an external file.
ASWEEP performs sweeps for model terms in an analysis of variance.
AUDISPLAY produces further output for an unbalanced design (after AUNBALANCED).
AUGRAPH plots tables of means from AUNBALANCED.
AUKEEP saves output from analysis of an unbalanced design (by AUNBALANCED).
AUNBALANCED performs analysis of variance for unbalanced designs.
AUMCOMPARISON performs pairwise multiple comparison tests for means from an unbalanced

analysis of variance, performed previously by AUNBALANCED.
AUPREDICT forms predictions from an unbalanced design (after AUNBALANCED).
AUSPREADSHEET saves results from an analysis of an unbalanced design (by AUNBALANCED) in a spreadsheet.
AU2RDA saves results from an unbalanced analysis of variance, by AUNBALANCED, in R data frames.
AYPARALLEL does the same analysis of variance for several y-variates, and collates the output.
A2DISPLAY provides further output following an analysis of variance by A2WAY.
A2KEEP copies information from an A2WAY analysis into Genstat data structures.
A2PLOT plots effects from two-level designs with robust s.e. estimates.
A2RDA saves results from an analysis of variance in R data frames.
A2RESULTSUMMARY provides a summary of results from an analysis by A2WAY.
A2WAY performs analysis of variance of a balanced or unbalanced design with up to two treatment factors.
BACKTRANSFORM calculates back-transformed means with approximate standard errors and confidence intervals.
BAFFYMETRIX estimates expression values from an Affymetrix CED and CDF file.
BANK calculates the optimum aspect ratio for a graph.
BASELINE estimates a baseline for a series of numbers whose minimum value is drifting.
BBINOMIAL estimates the parameters of the beta binomial distribution.
BCDISPLAY displays a classification tree.
BCFDISPLAY displays information about a random classification forest.
BCFIDENTIFY identifies specimens using a random classification forest.
BCFOREST constructs a random classification forest.
BCIDENTIFY identifies specimens using a classification tree.
BCKEEP saves information from a classification tree.
BCLASSIFICATION constructs a classification tree.
BCONSTRUCT constructs a tree.
BCVALUES forms values for nodes of a classification tree.
BGIMPORT imports MCMC output in CODA format produced by WinBUGS or OpenBUGS.
BGPLOT produces plots for output and diagnostics from MCMC simulations.
BGRAPH plots a tree.
BGXGENSTAT runs WinBUGS or OpenBUGS from Genstat in batch mode using scripts.
BIPLOT produces a biplot from a set of variates.
BJESTIMATE fits an ARIMA model, with forecast and residual checks.
BJFORECAST plots forecasts of a time series using a previously fitted ARIMA.
BJIDENTIFY displays time series statistics useful for ARIMA model selection.
BKDISPLAY displays an identification key.
BKEY constructs an identification key.
BKIDENTIFY identifies specimens using a key.
BKKEEP saves information from an identification key.
BLANDALTMAN produces Bland-Altman plots to assess the agreement between two variates.
BNTEST calculates one- and two-sample binomial tests.
BOOTSTRAP produces bootstrapped estimates, standard errors and distributions.
BOXPLOT draws box-and-whisker diagrams or schematic plots.
BPCONVERT converts bit patterns between integers, pointers of set bits and textual descriptions.
BPRINT displays a tree.
BPRUNE prunes a tree using minimal cost complexity.
BRDISPLAY displays a regression key.
BREGRESSION constructs a regression tree.
BRFDISPLAY displays information about a random regression forest.

BRFOREST constructs a random regression forest.
BRFPREDICT makes predictions using a random regression forest.
BRKEEP saves information from a regression tree.
BRPREDICT makes predictions using a regression tree.
BRVALUES forms values for nodes of a regression tree.
CABIPLOT plots results from correspondence analysis or multiple correspondence analysis.
CANCORRELATION does canonical correlation analysis.
CASSOCIATION calculates measures of association for circular data.
CATRENDTEST calculates the Cochran-Armitage chi-square test for trend.
CCA performs canonical correspondence analysis.
CCOMPARE tests whether samples from circular distributions have a common mean direction or have identical distributions.
CDESCRIBE calculates summary statistics and tests of circular data.
CDNAUGMENTEDDESIGN constructs an augmented block design, using CycDesigN if the controls are in an incomplete-block design.
CDNBLOCKDESIGN constructs a block design using CycDesigN.
CDNPREP constructs a multi-location partially-replicated design using CycDesigN.
CDNROWCOLUMNDESIGN constructs a row-column design using CycDesigN.
CENSOR pre-processes censored data before analysis by ANOVA.
CHECKARGUMENT checks the arguments of a procedure.
CHIPERMTEST performs a random permutation test for a two-dimensional contingency table.
CHISQUARE calculates chi-square statistics for one- and two-way tables.
CINTERACTION clusters rows and columns of a two-way interaction table.
CLASSIFY obtains a starting classification for non-hierarchical clustering.
CMHTEST performs the Cochran-Mantel-Haenszel test.
CONCORD is a synonym for KCONCORDANCE.
CONFIDENCE calculates simultaneous confidence intervals.
CONVEXHULL finds the points of a single or a full peel of convex hulls.
CORANALYSIS does correspondence analysis, or reciprocal averaging.
CORRESP is a synonym for CORANALYSIS.
COVDESIGN produces experimental designs efficient under analysis of covariance.
CRBIPLOT plots correlation or distance biplots after RDA, or ranking biplots after CCA.
CRTRIPLOT plots ordination biplots or triplots after CCA or RDA.
CSPRO reads a data set from a CSPro survey data file and dictionary, and loads it into Genstat or puts it into a spreadsheet file.
CUMDISTRIBUTION fits frequency distributions to accumulated counts.
CVAPLOT plots the mean and unit scores from a canonical variates analysis.
CVASCORES calculates scores for individual units in canonical variates analysis.
DARROW adds arrows to an existing plot.
DAYLENGTH calculates daylengths at a given period of the year.
DBARCHART produces bar charts for one or two-way tables.
DBCOMMAND runs an SQL command on an ODBC database.
DBEXPORT updates an ODBC database table using data from Genstat.
DBIMPORT loads data into Genstat from an ODBC database.
DBINFORMATION loads information on the tables and columns in an ODBC database.
DBIPLOT plots a biplot from an analysis by PCP, CVA or PCO.
DCIRCULAR plots circular data.
DCLUSTERLABELS labels clusters in a single-page dendrogram plotted by DDENDROGRAM.
DCOLOURS forms a band of graduated colours for graphics.
DCOMPOSITIONAL plots 3-part compositional data within a barycentric triangle.
DCORRELATION plots a correlation matrix.

DCOVARIOGRAM plots 2-dimensional auto- and cross-variograms.
DDEEXPORT sends data or commands to a Dynamic Data Exchange server.
DDEIMPORT gets data from a Dynamic Data Exchange (DDE) server.
DDENDROGRAM draws dendrograms with control over structure and style.
DDESIGN plots the plan of an experimental design.
DECIMALS sets the number of decimals for a structure, using its round-off.
DELLIPSE draws a 2-dimensional scatter plot with confidence, prediction and/or equal-frequency ellipses superimposed.
DEMC performs Bayesian computing using the Differential Evolution Markov Chain algorithm.
DERRORBAR adds error bars to a graph.
DESCRIBE saves and/or prints summary statistics for variates.
DESIGN helps to select and generate effective experimental designs.
DFOURIER performs a harmonic analysis of a univariate time series.
DFRTEXT adds text to a graphics frame.
DFUNCTION plots a function.
DHELP provides information about Genstat graphics.
DHSCATTERGRAM plots an h-scattergram.
DIALLEL analyses full and half diallel tables with parents.
DILUTION calculates Most Probable Numbers from dilution series data.
DIRECTORY prints or saves a list of files with names matching a specified mask.
DISCRIMINATE performs discriminant analysis.
DKALMAN plots results from an analysis by KALMAN.
DKEY adds a key to a graph.
DKSTPLOT produces diagnostic plots for space-time clustering.
DMADENSITY plots the empirical CDF or PDF (kernel smoothed) by groups.
DMASS plots discrete data like mass spectra, discrete probability functions.
DMSCATTER produces a scatter-plot matrix for one or two sets of variables.
DMST gives a high resolution plot of an ordination with minimum spanning tree.
DOTPLOT produces a dot-plot using line-printer or high-resolution graphics.
DOTHISTOGRAM plots dot histograms.
DPARALLEL displays multivariate data using parallel coordinates.
DPOLYGON draws polygons using high-resolution graphics.
DPROBABILITY plots probability distributions, and estimates their parameters.
DPSPECTRALPLOT calculates an estimate of the spectrum of a spatial point pattern.
DPTMAP draws maps for spatial point patterns using high-resolution graphics.
DPTREAD adds points interactively to a spatial point pattern.
DQMAP displays a genetic map.
DQMKSCORES plots a grid of marker scores for genotypes and indicates missing data.
DQMQTLSCAN plots the results of a genome-wide scan for QTL effects in multi-environment trials.
DQRECOMBINATIONS plots a matrix of recombination frequencies between markers.
DQSQTLSCAN plots the results of a genome-wide scan for QTL effects in single-environment trials.
DREFERENCELINE adds reference lines to a graph.
DRESIDUALS plots residuals.
DREPMEASURES plots profiles and differences of profiles for repeated measures data.
DRPOLYGON reads a polygon interactively from the current graphics device.
DSCATTER produces a scatter-plot matrix using high-resolution graphics.
DSEPARATIONPLOT creates a separation plot for visualising the fit of a model with a dichotomous (i.e. binary) or polytomous (i.e. multi-categorical) outcome.
DSPIDERWEB displays spider-web and star plots.

DSTTEST plots power and significance for t-tests, including equivalence tests.
DTABLE plots tables.
DTEXT adds text to a graph.
DTIMEPLOT produces horizontal bars displaying a continuous time record.
DVARIOGRAM plots fitted models to an experimental variogram.
DXDENSITY produces one-dimensional density (or violin) plots.
DXYDENSITY produces density plots for large data sets.
DXYGRAPH draws two-dimensional graphs with marginal distribution plots alongside the y- and x-axes.
DYPOLAR produces polar plots.
ECABUNDANCEPLOT produces rank/abundance, *ABC* and *k*-dominance plots.
ECACCUMULATION plots species accumulation curves for samples or individuals.
ECANOSIM performs an analysis of similarities (ANOSIM).
ECDIVERSITY calculates measures of diversity with jackknife or bootstrap estimates.
ECFIT fits models to species abundance data.
ECNICHE generates relative abundance of species for niche-based models.
ECNPESTIMATE calculates nonparametric estimates of species richness.
ECRAREFACTION calculates individual or sample-based rarefaction.
EDDUNNETT calculates equivalent deviates for Dunnett's simultaneous confidence interval around a control.
EDFTEST performs empirical-distribution-function goodness-of-fit tests.
EXAMPLE obtains and runs a Genstat example program.
EXPORT outputs data structures in foreign file formats, including Excel, Quattro, dBase, SPlus, Gauss, MatLab and Instat, or as plain or comma-delimited text.
EXTRABINOMIAL fits the models of Williams (1982) to overdispersed proportions.
FACAMEND permutes the levels and labels of a factor.
FACCOMBINATIONS forms a factor to indicate observations with identical combinations of values of a set of variates, texts or factors.
FACDIVIDE represents a factor by factorial combinations of a set of factors.
FACEXCLUDEUNUSED redefines the levels and labels of a factor to exclude those that are unused.
FACGETLABELS obtains the labels for a factor if it has been defined with labels, or constructs labels from its levels otherwise.
FACLEVSTANDARDIZE standardizes the levels or labels of a list of factors.
FACMERGE merges levels of factors.
FACPRODUCT forms a factor with a level for every combination of other factors.
FACSORT sorts the levels of a factor according to an index vector.
FACUNIQUE redefines a factor so that its levels and labels are unique.
FALIASTERMS forms information about aliased model terms in analysis of variance.
FBASICCONTRASTS breaks a model term down into its basic contrasts.
FBETWEENGROUPVECTORS forms variates and classifying factors containing within-group summaries to use e.g. in a between-group analysis.
FCOMPLEMENT forms the complement of an incomplete block design.
FCONTRASTS modifies a model formula to contain contrasts of factors.
FCORRELATION forms the correlation matrix for a list of variates.
FDESIGNFILE forms a backing-store file of information for AGDESIGN.
FDIALLEL forms the components of a diallel model for REML or regression.
FDISTINCTFACTORS checks sets of factors to remove any that define duplicate classifications.
FDRBONFERRONI estimates false discovery rates by a Bonferroni-type procedure.
FDRMIXTURE estimates false discovery rates using mixture distributions.
FEXACT2X2 does Fisher's exact test for 2×2 tables.
FFRAME forms multiple windows in a plot-matrix for high-resolution graphics.

FFREERESPONSEFACTOR forms multiple-response factors from free-response data.
FHADAMARDMATRIX forms Hadamard matrices.
FHAT calculates an estimate of the F nearest-neighbour distribution function.
FIELLER calculates effective doses or relative potencies.
FILEREAD reads data from a file.
FITINDIVIDUALLY fits regression models one term at a time.
FITMULTINOMIAL fits generalized linear models with multinomial distribution.
FITNONNEGATIVE is a synonym for RNONNEGATIVE.
FITPARALLEL is a synonym for RPARALLEL.
FITSCHNUTE is a synonym for RSCHNUTE.
FMEGAENVIRONMENTS forms mega-environments based on winning genotypes from an AMMI-2 model.
FMFACTORS forms a pointer of factors representing a multiple-response.
FNCORRELATION calculates correlations from variances and covariances, together with their variances and covariances.
FNLINEAR estimates linear functions of random variables, and calculates their variances and covariances.
FNPOWER estimates products of powers of two random variables, and calculates their variances and covariances.
FOCCURRENCES counts how often each pair of treatments occurs in the same block.
FPARETOSET forms the Pareto optimal set of non-dominated groups.
FPLOTNUMBER forms plot numbers for a row-by-column design.
FPROJECTIONMATRIX forms a projection matrix for a set of model terms.
FREGULAR expands vectors onto a regular two-dimensional grid.
FRESTRICTEDSET forms vectors with the restricted subset of a list of vectors.
FRIEDMAN performs Friedman's non-parametric analysis of variance.
FROWCANONICALMATRIX puts a matrix into row canonical, or reduced row echelon, form.
FRTPRODUCTDESIGNMATRIX forms summation, or relationship, matrices for model terms.
FSPREADSHEET creates a Genstat Spreadsheet file (GSH) from specified data structures.
FSTRING forms a single string from a list of strings in a text.
FTEXT forms a text structure from a variate.
FUNIQUEVALUES redefines a variate or text so that its values are unique.
FVCOVARIANCE forms the variance-covariance matrix for a list of variates.
FVSTRING forms a string listing the identifiers of a set of data structures.
FWITHINTERMS forms factors to define terms representing the effects of one factor within another factor.
FZERO gives the F function expectation under complete spatial randomness.
F2DRESIDUALVARIOGRAM calculates and plots a 2-dimensional variogram from a 2-dimensional array of residuals.
GALOIS forms addition and multiplication tables for a Galois finite field.
GBGRIDCONVERSION converts GB grid references to or from latitudes and longitudes or to or from UTM coordinates.
GRIBIMPORT reads data from a GRIB2 meteorological data file, and loads it or converts it to a spreadsheet file.
GEE fits models to longitudinal data by generalized estimating equations.
GENPROCRUSTES performs a generalized Procrustes analysis.
GESTABILITY calculates stability coefficients for genotype-by-environment data.
GETNAME forms the name of a structure according to its IPRINT attribute.
GETRGB gets the RGB values of the standard graphics colours.
GETTEMPFOLDER gets gets the location of the folder used by Genstat for temporary files.
GGEBIPLOT plots displays to assess genotype+genotype-by-environment variation.

GHAT calculates an estimate of the G nearest-neighbour distribution function.
GINVERSE calculates the generalized inverse of a matrix.
GLDISPLAY displays further output from a GLMM analysis.
GLKEEP saves results from a GLMM analysis.
GLM analyses non-standard generalized linear models.
GLMM fits a generalized linear mixed model.
GLPERMTEST does random permutation tests for generalized linear mixed models.
GLPLOT plots residuals from a GLMM analysis.
GLPREDICT forms predictions from a GLMM analysis.
GLRTEST calculates likelihood tests to assess random terms in a generalized linear mixed model.
GPREDICTION produces genomic predictions (breeding values) using phenotypic and molecular marker information.
GRANDOM generates pseudo-random numbers from probability distributions.
GRCSR generates completely spatially random points in a polygon.
GREJECTIONSAMPLE generates random samples using rejection sampling.
GRLABEL randomly labels two or more spatial point patterns.
GRMNOMIAL generates multinomial pseudo-random numbers.
GRMULTINORMAL generates multivariate normal pseudo-random numbers.
GRTHIN randomly thins a spatial point pattern.
GRTORSHIFT performs a random toroidal shift on a spatial point pattern.
GSTATISTIC calculates the gamma statistic of agreement for ordinal data.
G2AEXPORT forms a dbase file to transfer ANOVA output to Agronomix Generation II.
G2AFACTORS redefines block and treatment variables as factors.
G2VEXPORT forms a dbase file to transfer REML output to Agronomix Generation II.
HANOVA does hierarchical analysis of variance or covariance for unbalanced data.
HBOOTSTRAP performs bootstrap analyses to assess the reliability of clusters from hierarchical cluster analysis.
HCOMPAREGROUPINGS compares groupings generated, for example, from cluster analyses.
HEATUNITS calculates accumulated heat units of a temperature dependent process.
HFAMALGAMATIONS forms an amalgamations matrix from a minimum spanning tree.
HFCLUSTERS forms a set of clusters from an amalgamations matrix.
HGANALYSE analyses data using a hierarchical or double hierarchical generalized linear model.
HGDISPLAY displays results from a hierarchical or double hierarchical generalized linear model.
HGDRANDOMMODEL defines the random model in a hierarchical generalized linear model for the dispersion model of a double hierarchical generalized linear model.
HGFIXEDMODEL defines the fixed model for a hierarchical or double hierarchical generalized linear model.
HGFTEST calculates likelihood tests for fixed terms in a hierarchical generalized linear model
HGGRAPH draws a graph to display the fit of an HGLM or DHGLM analysis.
HGKEEP saves information from a hierarchical or double hierarchical generalized linear model analysis.
HGNONLINEAR defines nonlinear parameters for the fixed model of a hierarchical generalized linear model.
HGPLOT produces model-checking plots for a hierarchical or double hierarchical generalized linear model.
HGPREDICT forms predictions from a hierarchical or double hierarchical generalized linear model.
HGRANDOMMODEL defines the random model for a hierarchical or double hierarchical generalized linear model.
HGRTEST calculates likelihood tests for random terms in a hierarchical generalized linear model.
HGSTATUS displays the current HGLM model definitions.

HGWALD prints or saves Wald tests for fixed terms in an HGLM.
HPCLUSTERS prints a set of clusters.
IDENTIFY identifies an unknown specimen from a defined set of objects.
IFUNCTION estimates implicit and/or explicit functions of parameters.
IMPORT reads data from a foreign file format, and loads it or converts it to a spreadsheet file.
INSIDE determines whether points lie within a specified polygon.
JACKKNIFE produces Jackknife estimates and standard errors.
JOIN joins or merges two sets of vectors together, based on classifying keys.
KALMAN calculates estimates from the Kalman filter.
KAPLANMEIER calculates the Kaplan-Meier estimate of the survivor function.
KAPPA calculates a kappa coefficient of agreement for nominally scaled data.
KCONCORDANCE calculates Kendall's Coefficient of Concordance.
KCROSSVALIDATION computes cross validation statistics for punctual kriging.
KCSRENVELOPES simulates K function bounds under complete spatial randomness.
KERNELDENSITY uses kernel density estimation to estimate a sample density.
KHAT calculates an estimate of the K function.
KLABENVELOPES gives bounds for K function differences under random labelling.
KNEARESTNEIGHBOURS classifies items or predicts their responses by examining their *k* nearest neighbours.
KOLMOG2 performs a Kolmogorov-Smirnoff two-sample test.
KRUSKAL carries out a Kruskal-Wallis one-way analysis of variance.
KSED calculates the standard error for K function differences under random labelling.
KSTHAT calculates an estimate of the K function in space, time and space-time.
KSTMCTEST performs a Monte-Carlo test for space-time interaction.
KSTSE calculates the standard error for the space-time K function.
KTAU calculates Kendall's rank correlation coefficient τ.
KTORENVELOPES gives bounds for the bivariate K function under independence.
K12HAT calculates an estimate of the bivariate K function.
LCONCORDANCE calculates Lin's concordance correlation coefficient.
LIBEXAMPLE accesses examples and source code of library procedures.
LIBFILENAME supplies the names of information files for library procedures.
LIBHELP provides help information about library procedures.
LIBSOURCE obtains the source code of a Genstat procedure.
LIBVERSION provides the name of the current Genstat Procedure Library.
LINDEPENDENCE finds the linear relations associated with matrix singularities.
LORENZ plots the Lorenz curve and calculates the Gini and asymmetry coefficients.
LRIDGE does logistic ridge regression.
LRVSCREE prints a scree diagram and/or a difference table of latent roots.
LSIPLOT plots least significant intervals, saved from SEDLSI.
LSPLINE calculates design matrices to fit a natural polynomial or trigonometric L-spline as a linear mixed model.
LVARMODEL analyses a field trial using the Linear Variance Neighbour model.
MAANOVA does analysis of variance for a single-channel microarray design.
MABGCORRECT performs background correction of Affymetrix slides.
MACALCULATE corrects and transforms two-colour microarray differential expressions.
MADESIGN assesses the efficiency of a two-colour microarray design.
MAEBAYES modifies t-values by an empirical Bayes method.
MAESTIMATE estimates treatment effects from a two-colour microarray design.
MAHISTOGRAM plots histograms of microarray data.
MANNWHITNEY performs a Mann-Whitney U test.
MANOVA performs multivariate analysis of variance and covariance.

MANTEL assesses the association between similarity matrices.
MAPCLUSTER clusters probes or genes with microarray data.
MAPLOT produces two-dimensional plots of microarray data.
MAREGRESSION does regressions for single-channel microarray data.
MARMA calculates Affymetrix expression values.
MAROBUSTMEANS does a robust means analysis for Affymetrix slides.
MASCLUSTER clusters microarray slides.
MASHADE produces shade plots to display spatial variation of microarray data.
MAVDIFFERENCE applies the average difference algorithm to Affymetrix data.
MAVOLCANO produces volcano plots of microarray data.
MA2CLUSTER performs a two-way clustering of microarray data by probes (or genes) and slides.
MCNEMAR performs McNemar's test for the significance of changes.
MCOMPARISON performs pairwise multiple comparison tests within a table of means.
MCORANALYSIS does multiple correspondence analysis.
MCROSSPECTRUM performs a spectral analysis of a multiple time series.
MC1PSTATIONARY gives the stationary probabilities for a 1st-order Markov chain.
MEDIANTETRAD gives robust identification of multiple outliers in 2-way tables.
META combines estimates from individual trials.
MICHAELISMENTEN fits the Michaelis-Menten equation for substrate concentration versus time data.
MINFIELDWIDTH calculates minimum field widths for printing data structures.
MINIMIZE finds the minimum of a function calculated by a procedure.
MIN1DIMENSION finds the minimum of a function in one dimension.
MMPREDICT predicts the Michaelis-Menten curve for a particular set of parameter values.
MNORMALIZE normalizes two-colour microarray data.
MOVINGAVERAGE calculates and plots the moving average of a time series.
MPOLISH performs a median polish of two-way data.
MPOWER forms integer powers of a square matrix.
MTABULATE forms tables classified by multiple-response factors.
MULTMISSING estimates missing values for units in a multivariate data set.
MSEKERNEL2D estimates the mean square error for a kernel smoothing.
MVAOD does an analysis of distance of multivariate data.
MVARIOGRAM fits models to an experimental variogram.
MVFILL replaces missing values in a vector with the previous non-missing value.
NCONVERT converts integers between base 10 and other bases.
NCSPLINE calculates natural cubic spline basis functions (for use e.g. in REML).
NLAR1 fits curves with an AR1 or a power-distance correlation model.
NLCONTRASTS fits nonlinear contrasts to quantitative factors in ANOVA.
NORMTEST performs tests of univariate and/or multivariate Normality.
NOTICE provides news and other information about Genstat.
OPLS performs orthogonal partial least squares regression.
ORTHPOLYNOMIAL calculates orthogonal polynomials.
PAIRTEST performs t-tests for pairwise differences.
PARTIALCORRELATIONS calculates partial correlations for a list of variates.
PCOPROCRUSTES performs a multiple Procrustes analysis.
PDESIGN prints or stores treatment combinations tabulated by the block factors.
PDUPLICATE duplicates a pointer, with all its components.
PEAKFINDER finds the locations of peaks in an observed series.
PENSPLINE calculates design matrices to fit a penalized spline as a linear mixed model.
PERCENT expresses the body of a table as percentages of one of its margins.
PERIODTEST gives periodogram-based tests for white noise in time series.

PERMUTE forms all possible permutations of the integers 1...*n*.
PFACLEVELS prints levels and labels of factors.
PLINK prints a link to a graphics file into an HTML file.
PLS fits a partial least squares regression model.
PNTEST calculates one- and two-sample Poisson tests.
POSSEMIDEFINITE calculates a positive semi-definite approximation of a non-positive semi-definite symmetric matrix.
PPAIR displays results of t-tests for pairwise differences in compact diagrams.
PRCORRELATION calculates probabilities for product moment correlations.
PRDOUBLEPOISSON calculates the probability density for the double Poisson distribution.
PREWHITEN filters a time series before spectral analysis.
PRIMEPOWER decomposes a positive integer into its constituent prime powers.
PRKTAU calculates probabilities for Kendall's rank correlation coefficient τ.
PRMANNWHITNEYU calculates probabilities for the Mann-Whitney U statistic.
PROBITANALYSIS fits probit models allowing for natural mortality and immunity.
PRSPEARMAN calculates probabilities for Spearman's rank correlation statistic.
PRWILCOXON calculates probabilities for the Wilcoxon signed-rank statistic.
PSPLINE calculates design matrices to fit a P-spline as a linear mixed model.
PTAREAPOLYGON calculates the area of a polygon.
PTBOX generates a bounding or surrounding box for a spatial point pattern.
PTCLOSEPOLYGON closes open polygons.
PTDESCRIBE gives summary and second order statistics for a point process.
PTGRID generates a grid of points in a polygon.
PTINTENSITY calculates the overall density for a spatial point pattern.
PTKERNEL2D performs kernel smoothing of a spatial point pattern.
PTK3D performs kernel smoothing of space-time data.
PTREMOVE removes points interactively from a spatial point pattern.
PTROTATE rotates a point pattern.
PTSINPOLYGON returns points inside or outside a polygon.
QBESTGENOTYPES sorts individuals of a segregating population by their genetic similarity with a target genotype, using the identity by descent (IBD) information at QTL positions.
QCANDIDATES selects QTLs on the basis of a test statistic profile along the genome.
QCOCHRAN performs Cochran's *Q* test for differences between related-samples.
QDESCRIBE calculates descriptive statistics of molecular markers.
QDISCRIMINATE performs quadratic discrimination between groups i.e. allowing for different variance-covariance matrices.
QEIGENANALYSIS uses principal components analysis and the Tracy-Widom statistic to find the number of significant principal components to represent a set of variables.
QEXPORT exports genotypic data for QTL analysis.
QFACTOR allows the user to decide to convert texts or variates to factors.
QFLAPJACK creates a Flapjack project file from genotypic and phenotypic data.
QGSELECT obtains a representative selection of genotypes by means of genetic distance sampling or genetic distance optimization.
QIBDPROBABILITIES reads molecular marker data and calculates IBD probabilities.
QIMPORT imports genotypic and phenotypic data for QTL analysis.
QKINSHIPMATRIX forms a kinship matrix from molecular markers.
QLDDECAY estimates linkage disequilibrium (LD) decay along a chromosome.
QLINKAGEGROUPS forms linkage groups using marker data from experimental populations.
QLIST gets the user to select a response interactively from a list.
QMAP constructs genetic linkage maps using marker data from experimental populations.
QMASSOCIATION performs multi-environment marker-trait association analysis in a genetically

diverse population using bi-allelic and multi-allelic markers.
QMATCH matches different data structures to be used in QTL estimation.
QMBACKSELECT performs a QTL backward selection for loci in multi-environment trials or multiple populations.
QMESTIMATE calculates QTL effects in multi-environment trials or multiple populations.
QMKDIAGNOSTICS generates descriptive statistics and diagnostic plots of molecular marker data.
QMKRECODE recodes marker scores into separate alleles.
QMKSELECT obtains a representative selection of markers by means of genetic distance sampling or genetic distance optimization.
QMQTLSCAN performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in multi-environment trials or multiple populations.
QMTBACKSELECT performs a QTL backward selection for loci in multi-trait trials.
QMTESTIMATE calculates QTL effects in multi-trait trials.
QMTQTLSCAN performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in multi-trait trials.
QMVAF calculates percentage variance accounted for by QTL effects in a multi-environment analysis.
QMVESTIMATE replaces missing molecular marker scores using conditional genotypic probabilities.
QMVREPLACE replaces missing marker scores with the mode scores of the most similar genotypes.
QNORMALIZE performs quantile normalization.
QRECOMBINATIONS calculates the expected numbers of recombinations and the recombination frequencies between markers.
QREPORT creates an HTML report from QTL linkage or association analysis results.
QSASSOCIATION performs marker-trait association analysis in a genetically diverse population using bi-allelic and multi-allelic markers.
QSBACKSELECT performs a QTL backward selection for loci in single-environment trials.
QSELECTIONINDEX calculates (molecular) selection indexes by using phenotypic information and/or molecular scores of multiple traits.
QSESTIMATE calculates QTL effects in single-environment trials.
QSIMULATE simulates marker data and QTL effects for single and multiple environment trials.
QSQTLSCAN performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in single-environment trials.
QTHRESHOLD calculates a threshold to identify a significant QTL.
QUANTILE calculates quantiles of the values in a variate.
QUESTION obtains a response using a Genstat menu.
RADIALSPLINE calculates design matrices to fit a radial-spline surface as a linear mixed model.
RANK produces ranks, from the values in a variate, allowing for ties.
RAR1 fits regressions with an AR1 or a power-distance correlation model.
RBRADLEYTERRY fits the Bradley-Terry model for paired-comparison preference tests.
RCATENELSON performs a Cate-Nelson graphical analysis of bivariate data.
RCHECK checks the fit of a linear or generalized linear regression.
RCIRCULAR does circular regression of mean direction for an angular response.
RCOMPARISONS calculates comparison contrasts amongst regression means.
RCURVECOMMONNONLINEAR refits a standard curve with common nonlinear parameters across groups to provide s.e.'s for linear parameters.
RDA performs redundancy analysis.
RDESTIMATES plots one- or two-way tables of regression estimates.
REPPERIODOGRAM gives periodogram-based analyses for replicated time series.
RFFAMOUNT fits harmonic models to mean rainfall amounts for a Markov model.

RESHAPE reshapes a data set with classifying factors for rows and columns, into a reorganized data set with new identifying factors.
RFFPROBABILITY fits harmonic models to rainfall probabilities for a Markov model.
RFINLAYWILKINSON performs Finlay and Wilkinson's joint regression analysis of genotype-by-environment data.
RFSUMMARY forms summaries for a Markov model from rainfall data.
RGRAPH draws a graph to display the fit of a regression model.
RIDGE produces ridge regression and principal component regression analyses.
RJOINT does modified joint regression analysis for variety-by-environment data.
RLASSO performs lasso using iteratively reweighted least-squares.
RLFUNCTIONAL fits a linear functional relationship model.
RLIFETABLE calculates the life-table estimate of the survivor function.
RMGLM fits a model where different units follow different generalized linear models.
RMULTIVARIATE performs multivariate linear regression with accumulated tests.
RNEGBINOMIAL fits a negative binomial generalized linear model estimating the aggregation parameter.
RNONNEGATIVE fits a generalized linear model with nonnegativity constraints.
ROBSSPM forms robust estimates of sum-of-squares-and-products matrices.
RPAIR gives t-tests for all pairwise differences of means from a regression or generalized linear model.
RPARALLEL carries out analysis of parallelism for nonlinear functions.
RPERMTEST does random permutation tests for regression or generalized-linear-model analyses.
RPHCHANGE modifies a proportional hazards model fitted by RPHFIT.
RPHDISPLAY prints output for a proportional hazards model fitted by RPHFIT.
RPHFIT fits the proportional hazards model to survival data as a generalized linear model.
RPHKEEP saves information from a proportional hazards model fitted by RPHFIT.
RPHVECTORS forms vectors for fitting proportional hazards data as a generalized linear model.
RPOWER calculates the power (probability of detection) for regression models.
RPROPORTIONAL fits the proportional hazards model to survival data as a generalized linear model.
RQLINEAR fits and plots quantile regressions for linear models.
RQNONLINEAR fits and plots quantile regressions for nonlinear models.
RQSMOOTH fits and plots quantile regressions for loess or spline models.
RQUADRATIC fits a quadratic surface and estimates its stationary point.
RRETRIEVE retrieves a regression save structure from an external file.
RSCHNUTE fits a general 4 parameter growth model to a non-decreasing Y-variate.
RSCREEN performs screening tests for generalized or multivariate linear models.
RSEARCH helps search through models for a regression or generalized linear model.
RSPREADSHEET puts results from a regression, generalized linear or nonlinear model into Genstat spreadsheets.
RSTEST compares groups of right-censored survival data by nonparametric tests.
RSTORE stores a regression save structure in an external file.
RSURVIVAL models survival times of exponential, Weibull, extreme-value, log-logistic or lognormal distributions.
RTCOMPARISONS calculates comparison contrasts within a multi-way table of means.
RUGPLOT draws "rugplots" to display the distribution of one or more samples.
RUNTEST performs a test of randomness of a sequence of observations.
RWALD calculates Wald and F tests for dropping terms from a regression.
RXGENSTAT submits a set of commands externally to R and reads the output.
RYPARALLEL fits the same regression model to several response variates, and collates the output.
R0INFLATED fits zero-inflated regression models to count data with excess zeros.

R0KEEP saves information from a zero-inflated regression model for count data with excess zeros fitted by R0INFLATED.
R2LINES fits two-straight-line (broken-stick) models to data.
SAGRAPES produces statistics and graphs for checking sensory panel performance.
SAMPLE samples from a set of units, possibly stratified by factors.
SBNTEST calculates the sample size for binomial tests.
SCORRELATION calculates the sample size to detect specified correlations.
SDISCRIMINATE selects the best set of variates to discriminate between groups.
SEDLSI calculates least significant intervals.
SED2ESE calculates effective standard errors that give good approximate sed's.
SETDEVICE opens a graphical file and specifies the device number on basis of its extension.
SETNAME sets the identifier of a data structure to be one specified in a text.
SIGNTEST performs a one or two sample sign test.
SIMPLEX searches for the minimum of a function using the Nelder-Mead algorithm.
SKEWSYMMETRY provides an analysis of skew-symmetry for an asymmetric matrix.
SLCONCORDANCE calculates the sample size for Lin's concordance coefficient.
SMANNWHITNEY calculates sample sizes for the Mann-Whitney test.
SMCNEMAR calculates sample sizes for McNemar's test.
SMOOTHSPECTRUM forms smoothed spectrum estimates for univariate time series.
SOM declares a self-organizing map.
SOMADJUST performs adjustments to the weights of a self-organizing map.
SOMDESCRIBE summarizes values of variables at nodes of a self-organizing map.
SOMESTIMATE estimates the weights for self-organizing maps.
SOMIDENTIFY allocates samples to nodes of a self-organizing map.
SOMPREDICT makes predictions using a self-organizing map.
SPCAPABILITY calculates capability statistics.
SPCCHART plots c or u charts representing numbers of defective items.
SPCOMBINE combines spreadsheet and data files, without reading them into Genstat.
SPCUSUM prints CUSUM tables for controlling a process mean.
SPEARMAN calculates Spearman's rank correlation coefficient.
SPEWMA plots exponentially weighted moving-average control charts.
SPLINE calculates a set of basis functions for M-, B- or I-splines.
SPNTEST calculates the sample size for a Poisson test.
SPPCHART plots p or np charts for binomial testing for defective items.
SPRECISION calculates the sample size to obtain a specified precision.
SPSHEWHART plots control charts for mean and standard deviation or range.
SPSYNTAX puts details about the syntax of commands into a spreadsheet.
SSIGNTEST calculates the sample size for a sign test.
STACK combines several data sets by "stacking" the corresponding vectors.
STANDARDIZE standardizes columns of a data matrix to have mean zero and variance one.
STEEL performs Steel's many-one rank test.
STEM produces a simple stem-and-leaf chart.
STTEST calculates the sample size for t-tests (including equivalence tests).
SUBSET forms vectors containing subsets of the values in other vectors.
SVBOOT bootstraps data from random surveys.
SVCALIBRATE performs generalized calibration of survey data.
SVGLM fits generalized linear models to survey data.
SVHOTDECK performs hot-deck and model-based imputation for survey data.
SVMERGE merges strata prior to survey analysis.
SVMFIT fits a support vector machine.
SVMPREDICT forms the predictions using a support vector machine.

SVREWEIGHT modifies survey weights, adjusting other weights to ensure that their overall sum remains unchanged.
SVSAMPLE constructs stratified random samples.
SVSTRATIFIED analyses stratified random surveys by expansion or ratio raising.
SVTABULATE tabulates data from random surveys, including multistage surveys and surveys with unequal probabilities of selection.
SVWEIGHT forms survey weights.
TABINSERT inserts the contents of a sub-table into a table.
TABMODE forms summary tables of modes of values.
TABSORT sorts tables so their margins are in ascending or descending order.
TABTABLE opens a tabbed-table spreadsheet in the Genstat client.
TALLY forms a simple tally table of the distinct values in a vector.
TCOMBINE combines several tables into a single table.
TENSORSPLINE calculates design matrices to fit a tensor-spline surface as a linear mixed model.
TEQUIVALENCE performs equivalence, non-inferiority and non-superiority tests.
THINPLATE calculates the basis functions for thin-plate splines.
TOBIT performs a Tobit linear mixed model analysis on data with fixed-threshold censoring.
TRELLIS does a trellis plot.
TTEST performs a one- or two-sample t-test.
TUKEYBIWEIGHT estimates means using the Tukey biweight algorithm.
TVARMA fits a vector autoregressive moving average (VARMA) model.
TVFORECAST forecasts future values from a vector autoregressive moving average (VARMA) model.
TVGRAPH plots a vector autoregressive moving average (VARMA) model.
TXPAD pads strings of a text structure with extra characters so that their lengths are equal.
TXPROGRESSION forms a text containing a progression of strings.
TXSPLIT splits a text into individual texts, at positions on each line marked by separator character(s).
T%CONTROL expresses tables as percentages of control cells.
UNSTACK splits vectors into individual vectors according to levels of a factor.
UTMCONVERSION converts between geographical latitude and longitude coordinates and UTM eastings and northings.
VABLOCKDESIGN analyses an incomplete-block design by REML, allowing automatic selection of random and spatial covariance models.
VAIC calculates the Akaike and Schwarz (Bayesian) information coefficients for REML.
VALINEBYTESTER provides combinabilities and deviances for a line-by-tester trial analysed by VABLOCKDESIGN or VAROWCOLUMNDESIGN.
VALLSUBSETS fits all subsets of the fixed terms in a REML analysis.
VAMETA performs a REML meta analysis of a series of trials.
VAOPTIONS defines options for the fitting of models by VARANDOM and associated procedures.
VARANDOM finds the best REML random model from a set of models defined by VFMODEL.
VARECOVER recovers when REML, is unable to fit a model, by simplifying the random model.
VAROWCOLUMNDESIGN analyses a row-and-column design by REML, with automatic selection of the best random and spatial covariance model.
VASDISPLAY displays further output from an analysis by VASERIES.
VASERIES analyses a series of trials with incomplete-block or row-and-column designs by REML, automatically selecting the best random models.
VASKEEP copies information from an analysis by VASERIES into Genstat data structures.
VASMEANS saves experiment × treatment means from analysis of a series of trials by VASERIES.
VAYPARALLEL does the same REML analysis for several y-variates, and collates the output.
VBOOTSTRAP performs a parametric bootstrap of the fixed effects in a REML analysis.

VCHECK checks standardized residuals from a REML analysis.
VCRITICAL uses a parametric bootstrap to estimate critical values for a fixed term in a REML analysis.
VDEFFECTS plots one- or two-way tables of effects estimated in a REML analysis.
VDFIELDRESIDUALS display residuals from a REML analysis in field layout.
VEQUATE equates values across a set of data structures.
VFIXEDTESTS saves fixed tests from a REML analysis.
VFLC performs an F-test of random effects in a linear mixed model based on linear combinations of the responses, i.e. an FLC test.
VFMODEL forms a model-definition structure for a REML analysis.
VFPEDIGREE checks and prepares pedigree information from several factors, for use by VPEDIGREE and REML.
VFRESIDUALS obtains residuals, fitted values and their standard errors from a REML analysis.
VFSTRUCTURE adds a covariance-structure definition to a REML model-definition structure.
VFUNCTION calculates functions of variance components from a REML analysis.
VGESELECT selects the best variance-covariance model for a set of environments.
VGRAPH plots tables of means from REML.
VHERITABILITY calculates generalized heritability for a random term in a REML analysis.
VHOMOGENEITY tests homogeneity of variances and variance-covariance matrices.
VINTERPOLATE performs linear & inverse linear interpolation between variates.
VLINEBYTESTER analyses a line-by-tester trial by REML.
VLSD prints approximate least significant differences for REML means.
VMATRIX copies values and row/column labels from a matrix to variates or texts.
VMCOMPARISON performs pairwise comparisons between REML means.
VMETA performs a multi-treatment meta analysis using summary results from individual experiments.
VMODEL specifies the model for a REML analysis using a model-definition structure defined by VFMODEL.
VNEARESTNEIGHBOUR analyses a field trial using nearest neighbour analysis.
VORTHPOLYNOMIAL calculates orthogonal polynomials over time for repeated measures.
VPERMTEST does random permutation tests for the fixed effects in a REML analysis.
VPLOT plots residuals from a REML analysis.
VPOWER uses a parametric bootstrap to estimate the power (probability of detection) for terms in a REML analysis.
VRACCUMULATE forms a summary accumulating the results of a sequence of REML random models.
VRADD adds terms from a REML fixed model into a Genstat regression.
VRCHECK checks effects of a random term in a REML analysis.
VRDISPLAY displays output for a REML fixed model fitted in a Genstat regression.
VRDROP drops terms in a REML fixed model from a Genstat regression.
VREGRESS performs regression across variates.
VRFIT fits terms from a REML fixed model in a Genstat regression.
VRKEEP saves output for a REML fixed model fitted in a Genstat regression.
VRMETAMODEL forms the random model for a REML meta analysis.
VRPERMTEST performs permutation tests for random terms in REML analysis.
VRSETUP sets up Genstat regression to assess terms from a REML fixed model.
VRSWITCH adds or drops terms from a REML fixed model in a Genstat regression.
VRTRY tries the effect of adding and dropping individual terms from a REML fixed model in a Genstat regression.
VSAMPLESIZE estimates the replication to detect a fixed term or contrast in a REML analysis, using parametric bootstrap.

VSCREEN performs screening tests for fixed terms in a REML analysis.
VSOM analyses a simple REML variance components model for outliers using a variance shift outlier model.
VSPECTRALCHECK forms the spectral components from the canonical components of a multitiered design, and constrains any negative spectral components to zero.
VSPREADSHEET saves results from a REML analysis in a spreadsheet.
VSUMMARY summarizes a variate, with classifying factors, into a data matrix of variates and factors.
VSURFACE fits a 2-dimensional spline surface using REML, and estimates its extreme point.
VTABLE forms a variate and set of classifying factors from a table.
VTCOMPARISONS calculates comparison contrasts within a multi-way table of predicted means from a REML analysis.
VUVCOVARIANCE forms the unit-by-unit variance-covariance matrix for specified variance components in a REML model.
WADLEY fits models for Wadley's problem, allowing alternative links and errors.
WILCOXON performs a Wilcoxon Matched-Pairs (Signed-Rank) test.
WINDROSE plots rose diagrams of circular data like wind speeds.
WSTATISTIC calculates the Shapiro-Wilk test for Normality.
XOCATEGORIES performs analyses of categorical data from cross-over trials.
XOEFFICIENCY calculates efficiency of estimating effects in cross-over designs.
XOPOWER estimates the power of contrasts in cross-over designs.
YTRANSFORM estimates the parameter lambda of a single parameter transformation.

ABIVARIATE

Produces graphs and statistics for bivariate analysis of variance (R.F.A. Poultney).

Options

PRINT = *string tokens*	Controls printing of statistics from the bivariate analysis (error, treatment); default erro, trea
APRINT = *string tokens*	Controls output from the (univariate) ANOVAs of Y1 and Y2 (usual ANOVA print options); default aovt
TREATMENTSTRUCTURE = *formula*	Treatment terms to be fitted in the analysis of variance; this option must be set
BLOCKSTRUCTURE = *formula*	Block model defining the error terms in the analysis of variance; if unset, the design is assumed to be unstratified (i.e. to have a single error term)
TERM = *formula*	Single model term identifying the treatment term whose means are to be plotted
STRATUM = *formula*	Stratum from which to extract treatment information; default is to take the bottom stratum
FACTORIAL = *scalar*	Limit on number of factors in a treatment term; default 3
PROBABILITY = *scalar*	Significance level to use in the calculation of the radius of the confidence region and the region of non-significance; default 0.95
GRAPHICS = *string token*	Type of graphical output (lineprinter, highresolution); default high
STYLE = *string token*	controls the style of axes in a high-resolution graph (xy, none); default xy
LABELS = *factor* or *text*	Plotting symbols for the means; default is to take the letters A to Z, then a to z

Parameters

Y1 = *variates*	First variate for the bivariate analysis
Y2 = *variates*	Second variate for the bivariate analysis
TITLE = *texts*	Title for the graph

Description

ABIVARIATE produces a bivariate analysis of variance with a graphical representation of the results, as described by Dear & Mead (1983, 1984). The procedure was developed from a Genstat 4 macro, further information about which is given by Poultney & Riley (1986), and is intended primarily for data from intercropping experiments. The variates to be analysed (specified by parameters Y1 and Y2) are measurements, usually yields, taken on the two crops. The final parameter, TITLE, defines a title for the graph.

The procedure will work for any of the designs that can be analysed by ANOVA, except that there must be no pseudo-factors. Option TREATMENTSTRUCTURE defines the treatment formulae for the analysis, and the block formula is defined by the BLOCKSTRUCTURE option. BLOCKSTRUCTURE can be omitted if there is a single error stratum (i.e. the analysis is of a completely randomized design). The FACTORIAL option controls the number of factors in each treatment term, as in the ANOVA directive.

First of all, ABIVARIATE calculates a univariate analysis of variance for each of the variates Y1 and Y2, with output controlled by the APRINT option. The settings are the same as those in the ANOVA directive; by default APRINT=aovtable.

Output from the bivariate analysis of variance, which follows, is controlled by the PRINT option. The setting error generates the error summary statistics from the bivariate analysis:

Error Sum of Products, Variances after Adjustment for Covariance, Correlation Coefficient between `Y1` and `Y2`, Radius of Standard Errors, Radius of Confidence Regions, and Radius of Non-Significance Regions. The setting `treatment` produces the following statistics for each treatment term estimated within the specified error stratum: Treatment Sum of Products, Wilks' Lambda, Bivariate F-Statistic.

The stratum from which the means (and other information) are to be taken is defined by `STRATUM` option; if this is omitted, the lowest stratum is used. The significance level to use in the calculation of confidence regions is defined by the `PROBABILITY` option; by default this is 0.95.

The `TERM` option specifies a treatment term whose means are to be represented graphically. The means are plotted on axes transformed to allow for the variability in, and the correlation between, each crop variate. The plotting symbols can be defined as a factor or text using the option `LABELS`. Alternatively they will be taken to be the first *n* values of the series A to Z, a to z where *n* is the number of means to be plotted. The graph can be either line printer or high resolution, the default being high resolution. The external axes of a high-resolution graph can be suppressed by setting `STYLE=none`.

Problems arise in situations where the table of means to be plotted is incomplete; this can occur when a whole factor level is restricted out, or where the treatment structure is nested within a control. The length of the vector `LABELS` is calculated as the number of cells in the table, including missing values. If `LABELS` is declared, it must have length equal to the dimension of the table otherwise a fault will occur. Similarly, the calculation of the radius statistics is based on the assumption that the table of means is complete and has equal replication. These values, if printed, would be incorrect for a table with missing cells and so are suppressed. They can be calculated by hand as shown by Dear & Mead (1983).

Options: `PRINT`, `APRINT`, `TREATMENTSTRUCTURE`, `BLOCKSTRUCTURE`, `TERM`, `STRATUM`, `FACTORIAL`, `PROBABILITY`, `GRAPHICS`, `STYLE`, `LABELS`.
Parameters: `Y1`, `Y2`, `TITLE`.

Method

(1) calculate the SSP matrix for all terms in the formula
(2) transform the variables such that the new set are uncorrelated and have unit error variance
(3) calculate new axes based on the maximum and minimum points of the transformed variables
(4) draw the graph of the transformed means with the axes rotated such that they are at the same angle to the vertical

Action with `RESTRICT`

Variates `Y1` and `Y2` can be restricted, however this restriction must be identical for the two variates. Some problems may occur when whole levels of factors are restricted out leaving empty cells in the table of means to be plotted (see above).

References

Dear, K.B.G. & Mead, R. (1983). The use of bivariate analysis techniques for the presentation, analysis and interpretation of data. *Statistics in Intercropping Technical Report No. 1.* Department of Applied Statistics, University of Reading, U.K.

Dear, K.B.G. & Mead, R. (1984). Testing assumptions and other topics in bivariate analysis. *Statistics in Intercropping Technical Report No. 2.* Department of Applied Statistics, University of Reading, U.K.

Poultney, R.F.A. & Riley, J. (1986). A Genstat Macro for the Bivariate Analysis of Intercropping Data. *Genstat Newsletter*, **17**, 27-46

See also
Directive: ANOVA.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ABLUPS

Calculates BLUPs for block terms in an ANOVA analysis (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (blups); default blup
PTERMS = *formula*	Specifies the block terms whose BLUPs are to be printed; default is to print them all
PSE = *string tokens*	Types of standard errors to be printed with the BLUPs (differences, alldifferences, blups, allblups); default diff, blup
SAVE = *identifier*	Save structure for the ANOVA analysis; default is to take the most recent ANOVA analysis

Parameters

TERMS = *formula*	Block terms whose BLUPs etc are to be saved
BLUPS = *table* or *pointer* to *tables*	Saves the BLUPs
SEBLUPS = *table* or *pointer* to *tables*	Standard errors for the BLUPs of each term
SEDMEANS = *symmetric matrix* or *pointer* to *symmetric matrices*	Standard errors of differences between the BLUPs of each term

Description

This procedure can be used to calculate best linear unbiased predictors (BLUPs) for block terms in an analysis of variance that has been performed by the ANOVA directive. These differ from the ordinary ANOVA residuals in that they are *predictors* rather than estimates of the random effects. They usually have the property of *shrinkage*, i.e. they are biased towards zero. As a result they are more likely to represent future observations of the same terms.

By default, the BLUPs are from most recent ANOVA analysis. However, you can use an earlier analysis, by using the SAVE option of ABLUPS to specify its save structure (saved using the SAVE parameter of the earlier ANOVA command).

The BLUPs are usually printed. However, this can be suppressed by setting option PRINT=*. The PTERMS option can be used to specify the block terms whose BLUPs are to be printed. The default is to print the BLUPs for all the block terms.

The PSE option specifies which standard errors are printed, with the following settings.

differences	prints a summary of the standard errors of differences between pairs of BLUPs,
alldifferences	prints all the standard errors of differences between pairs of BLUPs,
blups	prints a summary of the standard errors of the BLUPs, and
allblups	prints all the standard errors of the BLUPs.

By default PSE=differences,blups.

The parameters of ABLUPS can save the BLUPs and standard errors. The TERMS parameter specifies the block terms whose BLUPs or standard errors are to be saved. The BLUPS parameter saves tables of BLUPs, the SEMEANS parameter saves tables containing their standard errors, and the SEDMEANS parameter saves symmetric matrices containing standard errors of differences between pairs of BLUPs. If you have a single term, you can supply a table or symmetric matrix for each of these parameters, as appropriate. However, if you have several terms, you must supply a pointer which will then be set up to contain as many tables or symmetric matrices as there are TERMS. A fault is given if the pointer has been defined already with a different number of elements to the number of TERMS.

Options: PRINT, PTERMS, PSE, SAVE.
Parameters: TERMS, BLUPS, SEBLUPS, SEDBLUPS.

Method
The BLUPs are calculated by a ridge regression where the y-variate is the variate of combined residuals from the ANOVA, and the explanatory terms are the block terms. The ridge variate contains the reciprocals of the gamma parameters, which are the variance components of the block terms divided by the variance of the bottom stratum. See, for example, equation 5.4 and the subsequent discussion in Robinson (1991). The variance-covariance matrix for the BLUPs is calculated as

$$\sigma^2 \times (\mathbf{Z}' \mathbf{Z} + \mathbf{G}^{-1} - \mathbf{Z}' \mathbf{X} (\mathbf{X}' \mathbf{X})^{-1} \mathbf{X}' \mathbf{Z})^{-1}$$

where **G** is a diagonal matrix containing the gamma parameters, **X** is the design matrix for the treatment effects and any covariates, **Z** is the design matrix for the block effects, and σ^2 is the variance of the bottom stratum. The design matrices and the inverse of $\mathbf{X}' \mathbf{X}$ are obtained using Genstat regression; see the TERMS and RKEEP directives for details.

Action with RESTRICT
If the y-variate originally analysed by ANOVA was restricted, the calculations will use only the units not excluded by the restriction.

Reference
Robinson, G.K. (1991). That BLUP is a good thing: the estimation of random effects. *Statistical Science*, **6**, 15-32.

See also
Directives: ANOVA, REML.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ABOXCOX

Estimates the power λ in a Box-Cox transformation, that maximizes the partial log-likelihood in ANOVA (W. van den Berg).

Options

PRINT = *string tokens*	Controls printed output (aovtable, lambda, monitoring); default aovt, lamb
TREATMENTSTRUCTURE = *formula*	Defines the treatment model; if this is not set, the default is taken from any existing setting defined by the TREATMENTSTRUCTURE directive
BLOCKSTRUCTURE = *formula*	Defines any block model; if this is not set, the default is taken from any existing setting defined by the BLOCKSTRUCTURE directive
COVARIATE = *variates*	Specifies any covariates; if this is not set, the default is taken from any existing setting defined by the COVARIATE directive
FACTORIAL = *scalar*	Limit in the number of factors in the terms generated from the TREATMENTSTRUCTURE formula; default 3
CONTRASTS = *scalar*	Limit on the order of a contrast of a treatment term; default 4
DEVIATIONS = *scalar*	Limit on the number of factors in a treatment term for the deviations from its fitted contrasts to be retained in the model; default 9
PLOT = *string token*	Whether to plot the partial log-likelihood (partialloglikelihood); default part
CIPROBABILITY = *scalar*	Probability level for the confidence interval for lambda; default 0.95, i.e. a 95% confidence interval
TRIALVALUES = *variate*	Values of λ for which the partial log-likelihood is to be calculated; default !(-4, -3.75 ... 4)
TRANSFORM = *string token*	How to transform the y-variate (estimate, trialvalue); default tria
STEPLENGTH = *scalar*	Steplength for estimating λ; default 0.01
MAXCYCLE = *scalar*	Maximum number of iterations; default 100
TOLERANCE = *scalar*	Tolerance for convergence; default 0.00001
ASAVE = *identifier*	Saves the ANOVA save structure from the analysis of variance

Parameters

Y = *variates*	Response variate
NEWY = *variates*	Saves the transformed response variate
LAMBDA = *scalars*	Saves the estimated value of λ
LOWER = *scalar*s	Saves the lower confidence limit for λ
UPPER = *scalar*s	Saves the upper confidence limit for λ

Description

ABOXCOX uses profile likelihood to estimate the parameter λ for a Box-Cox transformation (Box & Cox 1964) in a an ANOVA analysis. The transformation is defined as

$y^{\lambda} = (y^{\lambda} - 1) / \lambda$ $\quad \lambda \neq 0$

$y^{\lambda} = \log_e(y)$ $\quad \lambda = 0$

The TRIALVALUES option supplies trial values of λ (in a variate) at which the partial log-likelihood is evaluated. If the maximum value is within the range of the supplied values,

ABOXCOX then finds the value of λ that maximizes the partial log-likelihood, using a Newton-Raphson algorithm. It also estimates confidence limits for λ. The probability for the interval is specified by the option CIPROBABILITY; default 0.95 (i.e. 95%). Note: if the confidence region includes the value one, there is no evidence (at the specified probability level) to support taking a transformation.

The response variate is supplied by the Y parameter, and must contain only positive values. The transformed variate can be saved by the NEWY parameter. The TRANSFORM option controls whether the transformation uses the estimated value of λ or the best of the trial values (default). Using the trial value will usually provide results that are easier to interpret. For example, if the estimated value is close to zero, it may be clearer to use a logarithmic transformation than the power transformation. The estimated value of λ can be saved by the LAMBDA parameter, and its confidence limits can be saved by the LOWER and UPPER parameters.

The treatment model can be specified using the TREATMENTSTRUCTURE option, the block structure (if any) can be specified by the BLOCKSTRUCTURE option, and the COVARIATE option can be used to list any covariates. If any of these options is unset, the default is taken from any existing setting defined by the directives TREATMENTSTRUCTURE, BLOCKSTRUCTURE or COVARIATE, respectively. The FACTORIAL option can be used to set a limit on the number of factors in the terms generated from the TREATMENTSTRUCTURE option.

Contrasts can be specified by using the functions POL, REG, COMPARISON, POLND or REGND in the TREATMENTSTRUCTURE formula, as in ANOVA. The CONTRASTS option places a limit on the order of contrasts that are fitted. The DEVIATIONS option sets a limit on the number of factors in the terms whose deviations from the fitted contrasts are to be retained in the model. See ANOVA for more details.

Printed output is controlled by the PRINT option, with settings:

aovtable	prints the analysis-of-variance table of the transformed variate;
lambda	prints the estimated value of λ, and its confidence limits; and
monitoring	reports the progress of the estimation.

The default is to print the analysis-of-variance table and the estimate of λ with its confidence limits.

The ASAVE option can be used to save the ANOVA save structure from the analysis of the transformed variate. This can then be used to produce further output, by the usual commands ADISPLAY, APLOT and so on.

By default, a plot of the partial log-likelihood is produced. This can be suppressed by setting option PLOT=*.

The STEPLENGTH option specifies the steplength for the estimation process (default 0.00001), the MAXCYCLE option specifies the maximum number of iterations (default 100), and the TOLERANCE option specifies the tolerance for convergence (default 0.00001).

Options: PRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE, FACTORIAL, CONTRASTS, DEVIATIONS, PLOT, CIPROBABILITY, TRIALVALUES, STEPLENGTH, MAXCYCLE, TOLERANCE, ASAVE.
Parameters: Y, NEWY, LAMBDA, LOWER, UPPER.

Method

The partial log-likelihood for λ can be found on pages 178-180 of Pawitan (2001). The confidence limits are estimated by cubic interpolation, using the INTERPOLATE directive. This is feasible only if at least two values have been evaluated on either side of the maximum. The TRIALVALUES option can be used to include additional values if this fails.

Action with `RESTRICT`
The y-variate may be restricted.

References
Box, G.E.P. & Cox, D.R. (1964). An analysis of transformations. *Journal of the Royal Statistical Society Series B*, **26**, 211–252.
Pawitan, Y. (2001). *In All Likelihood: Statistical Modelling And Inference Using Likelihood.* Oxford: Clarendon Press.

See also
Directive: `ANOVA`.
Procedure: `YTRANSFORM`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ACANONICAL

Determines the orthogonal decomposition of the sample space for a design, using an analysis of the canonical relationships between the projectors derived from two or more model formulae (C.J. Brien).

Options

PRINT = *string tokens*	What to print (decomposition, df, ecriteria, efficiencies); default deco
GRANDMEAN = *string token*	Add the term for the grand or overall mean to each formula (include, omit); default omit
CRITERIA = *string tokens*	The efficiency criteria to be saved and/or printed (aefficiency, mefficiency, sefficiency, eefficiency, xefficiency, order, dforth); default aeff, eeff, orde
FACTORIAL = *scalar*	Limit on the number of factors and variates in each model term default * i.e. no limit
TOLERANCE = *variate*	Tolerances for zero in various contexts; default 10^{-8} for all of these

Parameters

FORMULAE = *pointers*	Each pointer contains two or more model formulae whose joint decomposition is required
ORTHOGONALMETHOD = *string tokens*	Specifies the method to use for each model formula when orthogonalizing a projection matrix to those for terms that occur earlier in the formula (differencing, eigenmethods, hybrid); default hybr
PROJECTIONSETS = *pointers*	Saves the projection pointers formed from the formulae
COMBINEDPROJECTIONSET = *pointers*	Saves the projector pointers that produce the orthogonal decomposition
EFFICIENCYFACTORS = *pointers*	Saves the canonical efficiency factors
ECRITERIA = *pointers*	Saves the unadjusted efficiency criteria
ADJECRITERIA = *pointers*	Saves the adjusted efficiency criteria
ADJDF = *pointers*	Saves the adjusted degrees of freedom
SAVE = *pointers*	Saves information about the analysis for use by ACDISPLAY and ACKEEP

Description

ACANONICAL forms the decomposition of the sample space of a design, to examine its "anatomy" (Brien 2016a, b), and summarizes this in a decomposition table (Brien & Bailey 2009, Bailey & Brien 2016). This reflects the properties of the design, by showing the confounding between sources from different model formulae. The decomposition table is similar to the skeleton anova table that the ANOVA directive produces based on the BLOCKSTRUCTURE and TREATMENTSTRUCTURE formulae for designs that have first order balance. ANOVA performs a dummy analysis of a randomly-generated variate on which a series of sweeps are performed. On the other hand, ACANONICAL produces the decomposition table by perfoming an eigenanalysis of the canonical relationships between projection matrices corresponding to the sources derived from the terms in the formulae. It is more general than the ANOVA directive in that it can produce a decomposition table for arbitrarily non-orthogonal designs and is not restricted to two formulae. However, for designs with 500 or more observations, the analysis may take in excess

of 5 minutes.

The FORMULAE parameter specifies the model formulae for which the decomposition table is to be produced.

The ORTHOGONALMETHOD parameter controls which method to use for orthogonalizing a projection matrix to those for terms that occur before it in a formula. Different methods can be used for different formulae.

The parameters PROJECTIONSETS, COMBINEDPROJECTIONSET, EFFICIENCYFACTORS, ECRITERIA, ADJECRITERIA, ADJDF save information from the analysis. Each of these forms a pointer, whose number of elements is one less than the number of formulae. The first element contains the result of using the projectors from the second formula to decompose those from the first. The second takes that result, and decomposes it according to the projectors from the third formula. This process of refining the current decomposition using the projectors from the next formula is continued until there are no unused formulae.

The PROJECTIONSETS parameter saves a triply-suffixed system of pointers to projector pointers from the pairs of decompositions. Each projection pointer has a 'matrix' element containing the projection matrix for a source, and a 'df' element containing the degrees of freedom of the projection matrix. Suppose that $\{P_i\}$ is a set of projection pointers for a decomposition up to the *i*th formula, and that this decomposition is to be further refined using the set $\{P_j\}$ of projection pointers corresponding to the *j*th formula. The projection matrix in each case will be the projector onto the subspace of a $\{P_i\}$ projector, pertaining to the subspace of a $\{P_j\}$ projector.

The COMBINEDPROJECTIONSET parameter saves a pointer, with a single suffix, containing the set of projection pointers whose 'matrix' component contain the non-zero projection matrices, that produce the orthogonal decomposition summarized in the decomposition table; see Brien & Bailey (2009, 2010) and Bailey & Brien (2016) for structure-balanced examples. The 'labels' of the projection pointers reflect the sources involved in the subspaces that are projected onto by the corresponding projector.

The EFFICIENCYFACTORS parameter saves a pointer, with a pair of suffices, that contains the set of variates containing the canonical efficiency factors for each combination of a matrix from $\{P_j\}$, and a matrix from $\{P_i\}$, with non-zero efficiency factors. The efficiency factors are adjusted for all matrices preceding it in forming the decomposition.

The ECRITERIA parameter saves a set of matrices, each of which contains one of the unadjusted efficiency criteria, nominated by the option CRITERIA, for one of the decompositions.

The ADJECRITERIA parameter saves a set of matrices, each of which contains one of the adjusted efficiency criteria, nominated by the option CRITERIA, for one of the decompositions.

The ADJDF parameter saves a set of matrices, each of which contains the degrees of freedom of an adjusted projector from $\{P_j\}$, where the matrix from $\{P_j\}$ has been adjusted for all those preceding it in $\{P_j\}$.

The SAVE parameter saves all the information from the analysis, in a pointer with elements status, efficiencies, effcriteria, adjeffcriteria, adjdf and combinedset.

The PRINT option controls printing, with settings:

decomposition	table summarizing the decomposition,
df	degrees of freedom,
ecriteria	efficiency criteria (as requested by the CRITERIA option), and
efficiencies	efficiency factors.

The GRANDMEAN option controls the inclusion or omission of a term for the grand or overall mean for each formula.

The CRITERIA option specifies the efficiency criteria to save or print, with the following settings:

`aefficiency`	the harmonic mean of the canonical efficiency factors;
`mefficiency`	the mean of the canonical efficiency factors,
`sefficiency`	the variance of the canonical efficiency factors,
`eefficiency`	the minimum of the canonical efficiency factors,
`xefficiency`	the maximum of the canonical efficiency factors,
`order`	the number of unique canonical efficiency factors, and
`dforth`	the number of degrees of freedom that are orthogonal.

The `FACTORIAL` option can be used to limit on the number of factors and variates in each term.

The `TOLERANCE` option specifies the values that are small enough to be considered zero. Its setting is a variate with two values. The first is used in determining if elements of structures, usually matrices, are sufficiently close to zero. The second determines if eigenvalues, or quantities derived from them, are sufficiently close to zero.

Options: `PRINT`, `GRANDMEAN`, `CRITERIA`, `FACTORIAL`, `TOLERANCE`.
Parameters: `FORMULAE`, `ORTHOGONALMETHOD`, `PROJECTIONSET`, `COMBINEDPROJECTIONSET`, `EFFICIENCYFACTORS`, `ECRITERIA`, `ADJECRITERIA`, `ADJDF`, `SAVE`.

Method

First of all, the set of projection pointers is obtained for each formula supplied by the `FORMULAE` parameter; there is one projection pointer for each term in a formula. Then an analysis of the canonical relationships is performed between the sets of projection matrices for the first two formulae. If there is a third formula, the relationships between its projectors and the already established decomposition are formed, and so on until all the formulae have been processed. The core of the analysis is the determination of eigenvalues of the product of pairs of projectors using the results of James & Wilkinson (1971).

However, if the order of balance between two projection matrices is 10 or more, the James &Wilkinson (1971) methods fails to produce an idempotent matrix. Equation 5.3 of Payne & Tobias (1992) is then used to obtain the projection matrices for their joint decomposition; this requires the inversion of a product involving the two projections matrices.

References

Bailey, R.A. & Brien, C.J. (2016) Randomization-based models for multitiered experiments. I. A chain of randomizations. *The Annals of Statistics*, **44**, 1131-1164.

Brien, C. J. (2016a) Multiphase experiments in practice, with an emphasis on nonorthogonal designs. I. A look back. submitted to *The Australian & New Zealand Journal of Statistics*.

Brien, C.J. (2016b) Multiphase experiments in practice, with an emphasis on nonorthogonal designs. II. Developments. submitted to *The Australian & New Zealand Journal of Statistics*.

Brien, C.J. & R.A. Bailey (2009). Decomposition tables for multitiered experiments. I. A chain of randomizations. *The Annals of Statistics*, **36**, 4184 - 4213.

Brien, C.J. & R.A. Bailey (2010). Decomposition tables for multitiered experiments. II. Two-one randomizations. *The Annals of Statistics*, **38**, 3164 - 3190.

James, A.T. & Wilkinson, G.N. (1971) Factorization of the residual operator and canonical decomposition of nonorthogonal factors in the analysis of variance. *Biometrika*, **58**, 279-294.

Payne, R.W. & R.D. Tobias (1992). General balance, combination of information and the analysis of covariance. *Scandinavian Journal of Statistics*, **19**, 3-23.

See also

Procedures: `ACDISPLAY`, `ACKEEP`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ACDISPLAY

Provides further output from an analysis by ACANONICAL (C.J. Brien).

Option

PRINT = *string tokens*	What to print (decomposition, df, ecriteria, efficiencies); default deco

Parameter

SAVE = *pointer*	Information saved from ACANONICAL; if this is not set, the information is saved from the most recent ACANONICAL analysis

Description

ACDISPLAY allows you to display further output from the decomposition produced by the ACANONICAL procedure, without having to repeat the calculations.

The output is specified by the PRINT option, with settings:

decomposition	table summarizing the decomposition,
df	degrees of freedom,
ecriteria	efficiency criteria (as requested by the CRITERIA option), and
efficiencies	efficiency factors.

By default, the output is from the most recent ACANONICAL analysis. The SAVE parameter allows you to print information from an earlier analysis, by setting it to a pointer saved earlier using the SAVE parameter of ACANONICAL.

Option: PRINT.
Parameters: SAVE.

See also

Procedures: ACANONICAL, ACKEEP.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ACHECK

Checks assumptions for an ANOVA analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (tests, confirmation); default conf
ASSUMPTION = *string tokens*	Which assumptions to test (homogeneity, normality, stability); default homo, norm, stab
PROBABILITY = *scalar*	Critical value for the test probabilities to decide whether to generate warning messages; default=0.025
SAVE = *ANOVA save structure*	Specifies the analysis to be checked; by default this will be the most recent ANOVA

No parameters

Description

Procedure ACHECK checks some of the assumptions for an analysis of variance that has been performed by the ANOVA directive. By default, the most recent ANOVA analysis is checked. However, you can check an earlier analysis, by using the SAVE option of ACHECK to specify its save structure (saved using the SAVE parameter of the earlier ANOVA command).

The assumptions to check are controlled by the ASSUMPTIONS option, with the following settings.

homogeneity	performs Levene tests to check whether the residual variance seems to be affected by any of the terms in the analysis. With stratified designs it will make similar checks for the residual variation in the higher strata (e.g. for the whole-plot variation in a split-plot design).
normality	performs a Shapiro-Wilk test to check for evidence that the residuals do not come from a Normal distribution.
stability	performs two Levene tests to check whether the residual variance differs according to the size of the response. The data are divided into three groups (small, intermediate and large) according to the sizes of their fitted values. The tests compare the variance of the residuals in the first (small) group with those in the third (large) group, and the variance of the second (intermediate) group with the variance of other two groups combined.

By default, they are all tested.

ACHECK produces warning messages if any of the tests generates a test probability less than or equal to the value specified by the PROBABILITY option. The default value is 0.025 (i.e. 2.5%), which is the same as the value used for the similar messages that may occur with the summary of analysis in regression. It is important to realise that the estimated residuals (from either regression or analysis of variance) will be correlated. The Levene and Shapiro-Wilk tests assume that the residuals are independent Normally-distributed observations. Their test probabilities may therefore be too low – and generate too many significant results. So the use of a smaller critical probability value provides some protection against spurious messages.

Other output is controlled by the PRINT option, with settings:

tests	prints the detailed test results, and
confirmation	prints a confirmatory message if there are no problems.

By default PRINT=confirmation.

Options: PRINT, ASSUMPTIONS, PROBABILITY, SAVE.
Parameters: none.

Method
Details about Levene tests can be found in Snedecor & Cochran (1989); also see O'Neill & Mathews (2002) for further information about the issues that arise in their use in balanced analysis of variance.

The Shapiro-Wilk test is performed by the WSTATISTIC procedure, which uses the methods of Royston (1993, 1995).

Action with RESTRICT
If the y-variate in the ANOVA was restricted, only the units not excluded by the restriction will be included in the checks.

References
O'Neill, M.E. & Mathews, K.L. (2002) Levene tests of homogeneity of variance for general block and treatment designs. *Biometrics*, **58**, 216-224.

Royston, P. (1993). A toolkit for testing for non-normality in complete and censored samples. *The Statistician*, **42**, 37-43.

Royston, P. (1995). A remark on Algorithm AS 181: the W-test for Normality. *Applied Statistics*, **44**, 547-551.

Snedecor, G.W. & Cochran, W.G. (1989). *Statistical Methods (eighth edition)*. Iowa State University Press, Ames.

See also
Directive: ANOVA.
Procedures: APLOT, VCHECK, WSTATISTIC.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ACKEEP

Saves information from an analysis by ACANONICAL (C.J. Brien).

Options

COMBINEDPROJECTIONSET = *pointer*	Saves the projector pointers that produce the orthogonal decomposition
EFFICIENCYFACTORS = *pointer*	Saves the canonical efficiency factors
ECRITERIA = *pointer*	Saves the unadjusted efficiency criteria
ADJECRITERIA = *pointer*	Saves the adjusted efficiency criteria
ADJDF = *pointer*	Saves the adjusted degrees of freedom
SAVE = *pointer*	Information saved from ACANONICAL; if this is not set, the information is saved from the most recent ACANONICAL analysis

No parameters

Description

ACKEEP allows you to save information from an analysis by the ACANONICAL procedure.

ACANONICAL determines the decomposition of the sample space for a design, using an analysis of the canonical relationships between the projectors derived from two or more model formulae. It has parameters that allow you to save the information at the time of the analysis. ACKEEP provides a way to save information afterwards. It has options with the same names as the COMBINEDPROJECTIONSET, EFFICIENCYFACTORS, ECRITERIA, ADJECRITERIA and ADJDF parameters of ACANONICAL, which save the information in exactly the same way; see ACANONICAL for details.

By default, the information is saved from the most recent ACANONICAL analysis. The SAVE option of ACKEEP allows you to save information from an earlier analysis, by setting it to a pointer saved earlier using the SAVE parameter of ACANONICAL.

Options: COMBINEDPROJECTIONSET, EFFICIENCYFACTORS, ECRITERIA, ADJECRITERIA, ADJDF, SAVE.
Parameters: none.

See also

Procedures: ACANONICAL, ACDISPLAY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ACONFIDENCE

Calculates simultaneous confidence intervals for ANOVA means (D.M. Smith).

Options

PRINT = *string token*	Controls printed output (intervals); default inte
METHOD = *string token*	Type of interval (individual, smm, product, Bonferroni, Scheffe); default smm
FACTORIAL = *scalar*	Limit on the number of factors in each term; default 3
PROBABILITY = *scalar*	The required significance level; default 0.05
SAVE = *ANOVA save structure*	Save structure to provide the tables of means and associated information; default uses the save structure from the most recent ANOVA

Parameters

TERMS = *formula*	Treatment terms whose means are to be required
MEANS = *pointer* or *table*	Saves the means
LOWER = *pointer* or *table*	Saves the lower limits
UPPER = *pointer* or *table*	Saves the upper limits

Description

ACONFIDENCE calculates sets of simultaneous confidence intervals i.e. intervals whose formation takes account of the number of intervals formed, and the fact that the intervals are (slightly) correlated because of the use of a common variance (see Hsu 1996 and Bechhofer, Santner & Goldsman 1995). The methodology implemented in the procedure closely follows that described in Section 1.3 of Hsu (1996).

The type of interval to be formed is specified by the METHOD option, with settings individual, smm (studentized maximum modulus), product (inequality), Bonferroni and Scheffe. The individual setting calculates the intervals as if they were independent, each with the input probability. The smm setting calculates the intervals as correlated, each with a probability adjusted for the multiplicity of intervals. The two settings product and Bonferroni calculate the intervals as independent, but with a probability adjusted for the multiplicity of intervals. These two settings produce very similar intervals although the Bonferroni intervals are always slightly larger. The final setting Scheffe calculates the intervals using pivoted F statistics; see Hsu (1996, Section 1.3.7). The default setting is smm because it produces exact simultaneous confidence intervals.

The TERMS parameter specifies a model formula to define the treatment terms whose means and confidence intervals are required. The means (and the necessary associated information) are usually taken from the most recent analysis of variance (performed by ANOVA), but you can set the SAVE option to a save structure from another ANOVA if you want to examine means from an earlier analysis. As in ANOVA, the FACTORIAL option sets a limit on the number of factors in each term (default 3). Note: intervals cannot be formed for means whose effects are estimated in different strata.

The MEANS parameter can save the means. If the TERMS parameter specifies a single term, MEANS should be set to a table. If TERMS specifies several terms, you must supply a pointer which will then be set up to contain as many tables as there are terms. Similarly the LOWER parameter can save the lower bounds of the confidence intervals, and the UPPER parameter can save the upper bounds.

You can set option PRINT=* to suppress printing of the intervals; by default PRINT=intervals.

Options: PRINT, METHOD, FACTORIAL, PROBABILITY, SAVE.

Parameters: TERMS, MEANS, LOWER, UPPER.

Method

The methodology implemented is based on that described and reviewed in Hsu (1996), and Bechhofer, Santner & Goldsman (1995). For specific details of the tests these books should be referred to.

References

Bechhofer, R.E., Santner, T.J. & Goldsman, D.M. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. Wiley, New York.

Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also

Directive: ANOVA.

Procedures: AMCOMPARISON, CONFIDENCE.

Genstat Reference Manual 1 Summary section on: Analysis of variance.

ADETECTION

Calculates the minimum size of effect or contrast detectable in an analysis of variance (R.W. Payne).

Options

PRINT = *string token*	Prints the minimum size of response that can be detected (detected); default dete
TERM = *formula*	Treatment term to be assessed in the analysis
TREATMENTSTRUCTURE = *formula*	Treatment structure of the design; determined automatically from an ANOVA save structure if TREATMENTSTRUCTURE is unset or if SAVE is set
BLOCKSTRUCTURE = *formula*	Block structure of the design; determined automatically from an ANOVA save structure if BLOCKSTRUCTURE is unset or if SAVE is set
FACTORIAL = *scalar*	Limit on the number of factors in treatment terms; default 3
PROBABILITY = *scalar*	Significance level at which the response is required to be detected (assuming a one-sided test); default 0.05
TMETHOD = *string token*	Type of test to be made (onesided, twosided, equivalence, noninferiority); default ones
XCONTRASTS = *variate*	X-variate defining a contrast to be detected
CONTRASTTYPE = *string token*	Type of contrast (regression, comparison); default rege
TOLERANCE = *scalar*	Tolerance for the iterations to calculate the detectable response
SAVE = *ANOVA save structure*	Save structure to provide the information about the design

Parameters

POWER = *scalars* or *variates*	Specifies the power i.e. probability with which the response should be detected
RMS = *scalars*	Anticipated residual mean square corresponding to TERM; can be omitted if a SAVE structure is available
DETECTED = *scalars* or *variates*	Minimum size of difference or contrast between the effects of TERM that is to be detected

Description

ADETECTION finds the minimum size of effect or contrast that is detectable with a specified power (or probability) in an analysis of variance. The treatment term to test is specified using the TERM option of ADETECTION, and the power with which you want to detect it is given by the POWER parameter. You can save the size of response using the DETECTED parameter. This is printed by default, but you can set option PRINT=* to stop this.

As an alternative to detecting a difference between treatment effects, you can ask to detect a contrast. However, here the treatment term must be a main effect (that is, TERM must involve just one factor). The XCONTRASTS option then species a variate containing the coefficients defining the contrast, and the CONTRASTTYPE option indicates whether this is a regression contrast (as specified by the REG function) or a comparison (as specified by COMPARISON).

The PROBABILITY option specifies the significance level that you will be using in the analysis to detect the treatment difference or contrast; the default is 0.05, i.e. 5%. By default, ADETECTION assumes that a one-sided t-test is to be used, but you can set option TMETHOD=twosided to take a two-sided t-test instead.

Other settings of TMETHOD enable you to test for equivalence or for non-inferiority. With equivalence (TMETHOD=equivalence), DETECTED defines a threshold below which the treatments can be assumed to be equivalent. If the treatments have effects e_1 and e_2, the null hypothesis that the treatments are not equivalent is that either

$(e_1 - e_2) \leq -$DETECTED

or

$(e_1 - e_2) \geq$ DETECTED

with the alternative hypothesis that they are equivalent, i.e.

$-$DETECTED $< (e_1 - e_2) <$ DETECTED

(For further details see the *Method* information for procedure ASAMPLESIZE.) With non-inferiority (TMETHOD=noninferiority), DETECTED again specifies the threshold for the effect of one treatment to be superior to another. So, for example, to demonstrate non-inferiority of treatment 1 compared to treatment 2, the null hypothesis becomes

$(e_1 - e_2) \geq -$DETECTED

which represents a simple one-sided t-test.

ADETECTION needs to know the design, and the size of residual mean square anticipated for the stratum where the treatment term is estimated. This is provided most easily by supplying the analysis of a design with similar units and the same block and treatment structures as those that are to be used in the new design. To do this, you should analyse the earlier set of data with the ANOVA directive in the usual way. First define the strata (or error terms) for the design using the BLOCKSTRUCTURE directive, and the treatment model to be fitted using the TREATMENTSTRUCTURE directive. Then analyse the y-variate using the ANOVA directive. Provided you do not give any other ANOVA commands in the interim, ADETECTION will pick up the information automatically from the save information held within Genstat about the most recent ANOVA analysis. Alternatively, you can save the information explicitly in an ANOVA save structure, using the SAVE parameter of ANOVA, and then use this same save structure as the setting of the SAVE option of ADETECTION.

If you do not have a suitable earlier set of data, you should set up the design factors to contain the values required to define the units of the design. Then use the BLOCKSTRUCTURE and TREATMENTSTRUCTURE options of ADETECTION to define the strata and the treatment model, and the RMS option to specify the anticipated residual mean square for the stratum where TERM is estimated. There is also the compromise possibility that you can take the information about the design, the strata and treatment model from an ANOVA save structure (generated for example by the analysis of an artificial data set), but use the RMS parameter to specify a different residual mean square from the one in the analysis in the save structure. The treatment terms to be included are controlled by the FACTORIAL option; this sets a limit (by default 3) on the number of factors in a treatment term: terms containing more than that number are deleted.

The procedure involves an iterative search to find the response that gives the specified power. The TOLERANCE option sets the convergence criterion (on the probability scale); the default is 10^{-7}.

Options: PRINT, TERM, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, FACTORIAL, PROBABILITY, TMETHOD, XCONTRASTS, CONTRASTTYPE, TOLERANCE, SAVE.
Parameters: POWER, RMS, DETECTED.

Method

The standard error of difference between two treatment effects is

$\sqrt{(s^2 \times 2 / (r \times e))}$

where s^2 is the stratum variance of the stratum where the treatment term is estimated, e is the efficiency factor, and r is the replication of each effect. For a regression contrast the standard error is

$\sqrt{(s^2 \times 2 / (r \times sdiv \times e))}$

where *sdiv* is the sum of squares of the XCONTRASTS variate, and for a comparison contrast the standard error is

$\sqrt{(s^2 \times sdiv / (r \times e))}$

ADETECTION assumes that the treatment effects have equal replication. Unequal replication can be studied by defining a comparison between the effects. For example, to allow for a control level with two replicates, you could assume that the first two levels are for the control, and then study comparisons between their mean and the other levels.

See also

Directive: ANOVA.

Procedures: APOWER, ASAMPLESIZE.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

ADPOLYNOMIAL

Plots single-factor polynomial contrasts fitted by ANOVA (R.W. Payne).

Option

SAVE = *ANOVA save structure*	Save structure (from ANOVA) to provide details of the analysis from which the polynomials are to be plotted; default uses the save structure from the most recent ANOVA

Parameters

XFACTOR = *factors*	Factor over which the polynomial contrasts have been formed
GROUPS = *factors* or *pointers*	Factor(s) for which different polynomial coefficients should be plotted in the same graph
TRELLISGROUPS = *factors* or *pointers*	Factor or factors for which different polynomial coefficients should be plotted in a trellis plot
TITLE = *texts*	Title for the graph; default defines a title automatically
YTITLE = *texts*	Title for the y-axis; default ' '
XTITLE = *texts*	Title for the x-axis; default is to use the identifier of the XFACTOR
PENS = *variates*	Defines the pen to use to plot the points and/or line for each group defined by the GROUPS factors

Description

ADPOLYNOMIAL plots polynomials fitted in analyses by the ANOVA directive. (These are included in the treatment model for ANOVA by the functions POL or POLND.) It also plots the corresponding means so that you can see how well the polynomials fit. By default, the polynomials are plotted from the most recent analysis performed by ANOVA, but the SAVE option can be used to supply the save structure from an earlier analysis to use instead.

The XFACTOR parameter specifies the factor over whose effects the polynomial contrasts have been fitted. If the analysis contains interactions between the XFACTOR and other factors, you can plot the polynomials for all the combinations of levels of these other factors by setting the GROUPS and TRELLISGROUPS parameters. If only GROUPS is specified, all the polynomials are plotted in a single graph. Alternatively, you can set the TRELLISGROUPS parameter to one or more of the factors to produce a trellis plot; there is then a graph for each of the combination of levels of the trellis factors (and each of these graphs plots the polynomials for every level of the group factors, at the relevant levels of the trellis factors). You should set GROUPS or TRELLISGROUPS to the factor if there is only one factor, or to a pointer containing all the factors if there are several.

The TITLE, YTITLE and XTITLE parameters can supply titles for the graph, the y-axis and the x-axis, respectively. The symbols, colours and line styles that are used in a high-resolution plot are usually set up by ADPOLYNOMIAL automatically. If you want to control these yourself, you should use the PEN directive to define a pen with your preferred symbol, colour and line style, for each of the groups defined by combinations of the GROUPS factors. The pen numbers should then be supplied to ADPOLYNOMIAL, in a variate with a value for each group, using the PENS parameter.

Option: SAVE.
Parameters: XFACTOR, GROUPS, TRELLISGROUPS, TITLE, YTITLE, XTITLE, PENS.

Method
The coefficients of the polynomials are obtained using the APOLYNOMIAL procedure.

See also
Directives: ANOVA, TREATMENTSTRUCTURE.
Procedure: APOLYNOMIAL.
Functions: POL, POLND.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ADSPREADSHEET

Puts the data and plan of an experimental design into a spreadsheet (R.W. Payne).

Options

DATA = *factors* or *variates*	Data variables (e.g. design factors and covariates) to put into the data spreadsheet; default takes the factors defined by previous BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives
NEWDATA = *variates*	New variates (e.g. measurements to be taken during the experiment) to create and put into the data spreadsheet; default * i.e. none
Y = *variate* or *factor*	Specifies the *y*-coordinates of the plots for the plan spreadsheet
X = *variate* or *factor*	Specifies the *x*-coordinates of the plots for the plan spreadsheet
CONSTANTFACTORS = *string tokens*	Whether to put factors whose levels are constant in the *y* or *x* direction in a separate row or column of the Plan spreadsheet (y, x); default * i.e. neither
SEPARATOR = *text*	Separator for factor values in the plan spreadsheet; default '; '
OMITGAPS = *string token*	Whether to omit gaps when the plots in the plan are equally spaced (yes, no); default no
FOREGROUND = *scalar*, *variate* or *text*	Foreground colours to use for the plots in the experiment; default 'Black'
BACKGROUND = *scalar*, *variate* or *text*	Background colours to use for the plots in the experiment; default 'BlanchedAlmond'
CFACTORS = *factors*	Factors to determine the colour to use for each plot; default uses the first block factor or no colouring otherwise
GAPFOREGROUND = *text* or *scalar*	Foreground colour for gaps and surrounding plots; default 'Black'
GAPBACKGROUND = *text* or *scalar*	Background colour for gaps and surrounding plots; default 'LightGreen'
YFOREGROUND = *text* or *scalar*	Foreground colour for factors constant in y-direction; default 'Black'
YBACKGROUND = *text* or *scalar*	Background colour for factors constant in y-direction; default 'PaleTurquoise'
XFOREGROUND = *text* or *scalar*	Foreground colour for factors constant in x-direction; default 'Black'
XBACKGROUND = *text* or *scalar*	Background colour for factors constant in x-direction; default 'LightCyan'
SPREADSHEET = *string tokens*	Which spreadsheets to form (data, plan); default data
OUTFILENAME = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create

Parameters

FACTOR = *factors*	Factors to include in the plan spreadsheet; if unset, includes the factors defined by a previous

	TREATMENTSTRUCTURE directive
LABELS = *texts*	Labels to be used for each factor if its own levels or labels are inappropriate

Description

ADSPREADSHEET puts information about an experimental design into a spreadsheet. By default the spreadsheet is opened within Genstat itself, but you can save it to an external file by supplying its name using the OUTFILENAME option. The file can be a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx). If the name is specified without a suffix, '.gwb' is added (so that a Genstat workbook is saved).

The SPREADSHEET option specifies which sheets to form, with settings:

data	contains data variables i.e. design factors, covariates and measurements, and
plan	constructs a plan of the design.

By default, SPREADSHEET=data. If both sheets are formed, they are put together, as pages of a Genstat workbook.

The contents of the data spreadsheet are specified by the DATA and NEWDATA options. The DATA option lists existing data variables (i.e. design factors and covariates) to put into the data spreadsheet. If this is unset, the default is to take the factors defined by previous BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives; ADSPREADSHEET gives a failure diagnostic if the DATA option is unset and there has been no previous BLOCKSTRUCTURE or TREATMENTSTRUCTURE. The NEWDATA option allows you to include new spreadsheet columns to provide blank cells for new variates like measurements that are to be taken during the experiment. For security all the existing variables are protected so that they are read-only.

The locations of the plots in the plan spreadsheet are specified by variates or factors supplied by the X and Y parameters; these define the row and column of the plots in the sheet, respectively (with row coordinates increasing from top to bottom, and column coordinates increasing from left to right in the usual way). The plots need not be equally spaced. However, ADSPREADSHEET looks to see whether the coordinates in either direction are taken from a regular grid, possibly with some gaps: for example coordinates (1, 2, 4, 6) are on a grid with spacing 1 and gaps at 3 and 5. If so, ADSPREADSHEET will include rows or columns for all the coordinates, including the gaps (i.e, 1, 2, 3, 4, 5 and 6 for the example), unless you set option OMITGAPS=yes. The *x*-coordinates are shown in a units column of the spreadsheet, and the *y*-coordinates are given in a row at the bottom of the plan. If either Y or X is not specified, ADSPREADSHEET will generate values automatically according to the factors in the design – factors from a previous BLOCKSTRUCTURE directive, if available, otherwise from a previous TREATMENTSTRUCTURE directive.

The factors to include in the plan can be specified using the FACTOR parameter. If this is omitted, ADSPREADSHEET takes the factors from a previous TREATMENTSTRUCTURE directive (and fails if there has been none). The values of each factor are represented by its labels, if available, or otherwise its levels. The LABELS parameter allows alternative labels to be specified for each factor, if the existing levels or labels are too unsuitable. The values of the factors in each plot are listed in the equivalent cell of the spreadsheet. By default, they are separated from each other by a semi-colon and a space, but you can supply alternative separating characters using the SEPARATOR option. You can set option CONSTANTFACTORS to x to list the values of factors whose values are constant in the *x* direction separately, in a column on the left-hand side of the sheet. Similarly, the setting y causes factors whose values are constant in the *y* direction to be listed in a row at the top of the sheet.

The colouring of the cells in a Genstat can be controlled using the FOREGROUND, BACKGROUND, CFACTORS, GAPFOREGROUND, GAPBACKGROUND, YFOREGROUND, YBACKGROUND, XFOREGROUND and XBACKGROUND options. The colours can be specified as numbers defining

RGB values, or texts containing names of the standard Genstat colours; see the PEN diective for details. The FOREGROUND and BACKGROUND options control the colours of the text and background, respectively, of the spreadsheet cells that correspond to plots in the experiment. You can give the plots different colours by supplying several values (in texts or variates). ADSPREADSHEET then uses a different colour for each combination of levels of the factor or factors specified by the CFACTORS option. If several colours are defined, but CFACTORS is not set, the first factor in the block factor (in BLOCKSTRUCTURE) is used. If there are no block factors, the first defined colour is used for all the plots. The GAPFOREGROUND and GAPBACKGROUND options define the colour to use for the cells representing gaps in the experiment or surrounding it. The YFOREGROUND and YBACKGROUND options specify the colour for the text and background in the cells containing the names and levels of the factors constant in the y-direction. The XFOREGROUND and XBACKGROUND options similarly specify the colour for the text and background for the factors constant in the x-direction.

Options: DATA, NEWDATA, Y, X, CONSTANTFACTORS, SEPARATOR, OMITGAPS, FOREGROUND, BACKGROUND, CFACTORS, GAPFOREGROUND, GAPBACKGROUND, YFOREGROUND, YBACKGROUND, XFOREGROUND, XBACKGROUND, SPREADSHEET, OUTFILENAME.
Parameters: FACTOR, LABELS.

Action with RESTRICT
If X or Y or any of the factors in the plan is restricted, only the unrestricted plots will be included in the plan spreadsheet.

See also
Directive: SPLOAD.
Procedures: ASPREADSHEET, AUSPREADSHEET, DDESIGN, PDESIGN, FSPREADSHEET, VSPREADSHEET.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AEFFICIENCY

Calculates efficiency factors for experimental designs (R.W. Payne).

Options

FACTORIAL = *scalar*	Limit on the number of factors in each treatment term generated from TERMS; default 3
METHOD = *string token*	Whether to eliminate or ignore earlier model terms from the TERMS formula (eliminate, ignore); default elim
FORCED = *formula*	Terms to be eliminated before fitting TERMS; default * i.e. none

Parameters

TERMS = *formula*	Model terms
DF = *pointer* or *scalar*	Saves the degrees of freedom of the terms
EFFICIENCY = *pointer* or *variate*	Saves the efficiency factors of the terms
DFALIASED = *pointer* or *scalar*	Saves the number of aliased degrees of freedom of the terms

Description

The *efficiency factors* of a model term represent the proportion of the information about various contrasts amongst its effects that remains available for estimating the contrasts, after fitting the earlier terms in the analysis. If the term is balanced, the efficiency factors will all be equal. If not, their range gives an indication of the degree of imbalance.

The model terms of interest are specified by the TERMS parameter. You can also use the FORCED option to specify a set of model terms that must be eliminated before those in TERMS are fitted. By default, the efficiency factors are calculated under the assumption that the model terms in TERMS are to be fitted sequentially. So, each term is estimated eliminating the earlier terms in TERMS. Alternatively, you can set option METHOD=ignore to calculate the efficiency factors for the terms eliminating only their marginal terms and the terms in the FORCED formula. (Marginal terms are terms whose factors are a subset of those in the term: e.g. the main effects A and B are marginal terms of the interaction A.B.)

The EFFICIENCY parameter saves the efficiency factors. If the TERMS parameter specifies a single term, EFFICIENCY must be undeclared or set to a variate. If TERMS specifies several terms, you must supply a pointer which will then be set up to contain as many variates as there are terms. Similarly the DF parameter can save the numbers of degrees of freedom of each term, and the DFALIASED parameter can save the numbers of degrees of freedom of each term that are aliased either with terms in the FORCED formula or with terms that come before it in the TERMS formula.

Options: FACTORIAL, METHOD, FORCED.
Parameters: TERMS, DF, EFFICIENCY, DFALIASED.

Method

The efficiency factors are the eigenvalues of the matrix **TST**, where **T** is the projection matrix for the model term, and **S** is the projection matrix into the space orthogonal to the previous terms. The corresponding contrasts are the eigenvectors of the matrix. See Payne & Tobias (1992), Section 4.

AEFFICIENCY uses the FPROJECTIONMATRIX procedure to form projection matrices for the model terms. Marginal terms are eliminated using Equation (2.7) of Payne & Tobias (1992), amd the efficiency factors are calculated by an eigenvalue decomposition as in Equation (4.9).

Action with RESTRICT

AEFFICIENCY takes account of any restrictions on the y-variate.

Reference

Payne, R.W. & Tobias, R.D. (1992). General balance, combination of information and the analysis of covariance. *Scandinavian Journal of Statistics*, **19**, 3-23.

See also

Directive: ANOVA.
Procedure: ASWEEP.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AFALPHA

Generates alpha designs (R.W. Payne).

Option

PRINT = *string token*	Whether to print the design (design); default * i.e. no printing

Parameters

GENERATOR = *matrices*	generating array (of size number-of-plots-per-block by number-of-reps)
LEVELS = *scalars* or *variates*	Defines the levels of each treatment factor; if this is omitted, the levels of the TREATMENT factor are used, if available, otherwise LEVELS is determined from the generating array on the assumption that the blocks are to be of equal size
SEED = *scalar*	Seed to be used to randomize the design, if required
TREATMENTS = *factors*	Specifies the treatment factor for each design
REPLICATES = *factors*	Specifies the replicate factor
BLOCKS = *factors*	Specifies the block factor
UNITS = *factors*	Specifies the factor to index the units within each block

Description

Alpha designs are a very flexible class of resolvable incomplete block designs. A resolvable design is one in which each block contains only a selection of the treatments, but the blocks can be grouped together into subsets in which each treatment is replicated once. The groupings of blocks thus form replicates, and the block structure of the design is

```
Replicates / Blocks / Units
```

Such designs are particularly useful when there are many treatments to examine and the variability of the units is such that the block size needs to be kept small. Alpha designs were thus devised originally for the analysis of plant breeding trials (Patterson & Williams 1976), where many varieties may need to be evaluated in a single trial, and have the advantage that they can provide effective designs for any number of treatments.

The construction of an alpha design requires a $k \times r$ array of integers between 0 and s−1, where r is the number of replicates, and s is the number of blocks per replicate. If the number of treatments, v, is a multiple of the number of blocks per replicate, k will be the number of units in each block, and v will be given by $s \times k$. Otherwise, the design will have some blocks of size k and some of size $k-1$, and v will lie between $s \times (k-1)$ and s x k. Clearly, the properties of the design that is formed will be very dependent on the choice of array. Patterson, Williams and Hunter (1978) present 11 basic arrays to generate designs with up to 100 treatments and 2, 3 or 4 replicates when k is greater than 3 and s is greater than or equal to k; these arrays are reproduced in John (1987). Williams (1975) presents arrays for any sensible values of s and k with up to 100 treatments and 2 to 4 replicates.

Procedure AFALPHA generates the treatment, replicate, block and unit factors for an alpha design. The design can be printed by setting option PRINT=design, and the factors can be saved using the parameters TREATMENTS, REPLICATES, BLOCKS and UNITS. The generating array for the design must be specified as a $k \times r$ matrix using the GENERATOR parameter, and the number of levels of the treatment factor can be defined by the LEVELS parameter. If LEVELS is omitted, AFALPHA will see whether the TREATMENTS parameter has been set to a factor whose levels have already been defined; if not, AFALPHA will set LEVELS to the scalar value $v = s \times k$. By default the design is unrandomized, but randomization can be requested by setting the SEED parameter.

Option: PRINT.
Parameters: GENERATOR, LEVELS, SEED, TREATMENTS, REPLICATES, BLOCKS, UNITS.

Method
Each column of the generating array is used to form $s-1$ further columns by successively adding 1 modulo s. Next, s is added to row 2 of every column, $2s$ to row 3, and so on. Each resulting column then gives one of the blocks of the design, and the replicates are formed by the sets of columns that were all generated from the same initial column. If the design needs to have blocks of unequal sizes, procedure SUBSET is used to omit the necessary plots to form the smaller blocks.

References
Patterson, H.D. & Williams E.R. (1976). A new class of resolvable incomplete block designs. *Biometrika*, **63**, 83-92.
Patterson, H.D., Williams E.R. & Hunter, E.A. (1978). Block designs for variety trials. *Journal of Agricultural Science, Cambridge*, **90**, 395-400.
Williams, E.R. (1975). *A new class of resolvable block designs*. Ph.D. Thesis, University of Edinburgh.

See also
Procedure: AGALPHA.
Genstat Reference Manual 1 Summary sections on: Design of experiments, REML analysis of linear mixed models.

AFAUGMENTED

Forms an augmented design (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (design); default * i.e. none
TREATMENTSTRUCTURE = *formula*	Treatment terms, other than GENOTYPES, to be included in the analysis
BLOCKSTRUCTURE = *formula*	Defines the block structure of the basic design
COVARIATE = *variates*	Specifies any covariates to be included in the analysis
LEVTEST = *variate*	Levels to represent the test genotypes in the augmented GENOTYPES factor
LEVCONTROL = *scalar* or *variate*	Levels to represent the control genotype(s) if these are not already in the GENOTYPES factor
GENOTYPES = *factor*	Genotype factor
CONTROLS = *factor*	Factor identifying the controls
TESTVSCONTROL = *factor*	Factor representing the comparison between test and control genotypes
SUBPLOTS = *factor*	Factor to represent the subplots to be created for the test genotypes in the basic design
NSUBPLOTS = *scalar*	Number of subplots to create within each plot of the basic design
SUBCONTROLS = *scalar* or *variate*	Subplots to be used for control genotypes, if not already pre-allocated in the GENOTYPES and SUBPLOTS factors; default selects subplots for the controls at random within each whole-plot
NREPTEST = *scalar* or *variate*	Number of times to replicate the test genotypes; default 1
SEED = *scalar*	Seed for the random numbers used to randomize the allocation of the genotypes (a negative value implies no randomization); default 0

No parameters

Description

An augmented design is a design for assessing large numbers of treatments, usually test genotypes in a variety trial. The trial also contains controls; these are replicated while the tests are usually unreplicated.

The design is constructed from a basic design, which can be any standard design, for example, a randomized complete block design or a Latin square. In the simplest situation, a control genotype is allocated to each plot of the basic design. The design is then expanded, or *augmented*, so that each plot of the basic design is split into subplots. (So the plots of the basic design become the whole-plots of the augmented design.) The control genotype is allocated to one of the subplots in each plot, and test genotypes are allocated to the other subplots.

So you first need to generate the basic design, using a procedure like AGHIERARCHICAL or AGLATIN. You can then use AFAUGMENTED to augment it.

In the simplest situation, the basic design has blocking factors identifying its plots, and a treatment factor defined to indicate the control genotype allocated to each plot. For example, Lin & Poushinsky (1983) used a 4 × 4 Latin square as their basic design, with 4 different control genotypes. In Genstat this can be constructed using AGLATIN

```
POINTER [VALUES=Genotypes] tfact
AGLATIN [PRINT=*; ANALYSE=no] NROWS=4; NSQUARES=1; SEED=584578;\
```

```
                TREATMENTFACTORS=tfact; ROWS=Rows; COLUMNS=Columns
```

They then split each plot into 9 subplots, allocating the control to subplot 5 in each plot, and randomly allocated 128 test genotypes to the other subplots across the design. The Genstat command to do this is

```
VARIATE [VALUES=5...132] Tests
AFAUGMENTED [PRINT=design; BLOCKSTRUCTURE=Rows*Columns;\
            LEVTEST=Tests; GENOTYPES=Genotypes;\
            NSUBPLOTS=9; SUBCONTROLS=5]
```

The `BLOCKSTRUCTURE` option specifies the blocking structure of the basic design (here rows crossed with columns), and thus the blocking factors that need to be expanded. The `GENOTYPES` option specifies the genotypes factor which, on input, indicates the control genotype on each plot. The `NSUBPLOTS` option specifies the number of subplots to define within each plot, and the `SUBCONTROL` option specifies the subplot(s) to contain the control(s). The `LEVTEST` option specifies which levels of the augmented `GENOTYPES` factor are to represent the test genotypes. Setting option `PRINT=design` prints the design, using procedure `PDESIGN`; by default it is not printed.

Note that, if there are insufficient test genotypes, some plots may contain `NSUBPLOTS` minus one subplots. An error is given if there are too few genotypes for any of the plots to contain `NSUBPLOTS` subplots.

The `SEED` option specifies a seed for the random numbers that are used to make the allocations. The default value of zero continues an existing sequence of random numbers if any have already been used in the current Genstat job, or obtains a random seed using the system clock if none have been used already. You can also set `SEED=-1` if you want to suppress any randomization.

If the design has other treatments (as well as `GENOTYPES`), these can be specified using the `TREATMENTSTRUCTURE` option. This takes a model formula as its setting (so you would define the treatment terms that are to be included in the analysis). However, but it is sufficient just to list the factors if you prefer. These will then be expanded similarly to the blocking factors. Likewise, if you have covariates whose values are defined on the plots of the basic design, these can be specified using the `COVARIATE` option.

You can use the `CONTROLS` option to save a factor with a level for each control, and another level for all the test genotypes. You can also use the `TESTVSCONTROL` option to save a factor with one level for the control genotypes, and another level for the test genotypes. (These will be identical if there is only one control genotype.)

If you want to specify several controls in each whole-plot of the augmented design, you can define the basic design to have subplots already, namely those with the controls. For example, the program below has a balanced-incomplete-block design for three treatments as the basic design. The first block has controls 1 and 3, the second has 2 and 3, and the third has 1 and 2. So we start with two subplots. The `AFAUGMENTED` command expands the design to have eight subplots, adding 18 test genotypes. . The `SUBCONTROLS` option is now set to a variate to put the controls onto subplots 3 and 6, randomizing the allocation within each plot.

```
FACTOR      [LEVELS=3; VALUES=1,1,2,2,3,3] Blocks
FACTOR      [LEVELS=3; VALUES=1,3,2,3,1,2] Genotypes
VARIATE     [VALUES=101...118] Tests
VARIATE     [VALUES=3,6] Csubs
AFAUGMENTED [PRINT=design; BLOCKSTRUCTURE=Blocks;\
            LEVTEST=Tests; GENOTYPES=Genotypes;\
            NSUBPLOTS=8; SUBCONTROL=Csubs]
```

You can predefine the `SUBPLOTS` factor if you want to allocate the controls to the subplots explicitly, yourself. For example,

```
FACTOR      [LEVELS=32; VALUES=2,6...30] plots
FACTOR      [LEVELS=2; VALUES=(1,2)4] genotypes
```

```
AFAUGMENTED [SUBPLOTS=plots; LEVTEST=!(3...26);\
            GENOTYPES=genotypes; CONTROLS=controls]
```

puts control 1 in block 1 explicitly onto subplot 2, and control 2 in block 1 explicitly onto subplot 6, etc. The NSUBPLOTS option of AFAUGMENTED then need not be set, but will default to the number of levels defined for SUBPLOTS. Of course, if you do predefine the SUBPLOTS factor, you no longer need to have the same number of controls in each plot.

You can even define a null basic design. The "augmented" design will then simply consist of some control and test genotypes allocated to the (sub)plots within the field (with the SUBPLOTS and SUBCONTROL options determining the allocation of the controls as before). For example:

```
FACTOR      [LEVELS=32; VALUES=2,6...30] plots
FACTOR      [LEVELS=2; VALUES=(1,2)4] genotypes
VARIATE     [VALUES=3...26] tests
AFAUGMENTED [SUBPLOTS=plots; LEVTEST=tests;\
            GENOTYPES=genotypes; CONTROLS=controls]
```

By default, the test genotypes are unreplicated. You can set the NREPTEST option to a scalar to replicate every test genotype the same number of times, or to a variate to have different numbers of replicates (as, for example. in a partially-replicated design).

Options: PRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE, LEVTEST, LEVCONTROL, GENOTYPES, CONTROLS, TESTVSCONTROL, SUBPLOTS, NSUBPLOTS, SUBCONTROL, NREPTEST, SEED.
Parameters: none.

Action with RESTRICT
The procedure does not allow for restrictions, and will cancel any that have been applied.

Reference
Lin, C.S. & Poushinsky, G. (1983). A modified augmented design for an early stage of plant selection involving a large number of test lines without replication. *Biometrics*, **39**, 553-561.

See also
Procedure: CDNAUGMENTEDDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AFCARRYOVER

Forms factors to represent carry-over effects in cross-over trials (R.W. Payne).

Option

NONELEVEL = *scalar* or *text*	Level or label to use for the units with no carry-over

Parameters

TREATMENTS = *factors*	Factors identifying the (direct) effects of the treatments
SUBJECTS = *factors*	Factors identifying the subjects
PERIODS = *factors*	Factors identifying the periods
CARRYOVERFACTOR = *factors*	Factors to represent the carry-over effect of the treatments in the period immediately after the period in which they were applied
NOCARRYOVER = *factors*	Factors to represent the comparison between none and any carry-over effect of the treatments

Description

Cross-over trials are designed to study the effects of various treatments on a set of plots (in a field experiment) or subjects (in a medical trial). The special feature of these experiments is that the same plots or subjects are treated during several successive time periods, and there is interest both in the direct effect of a treatment during the period in which it is applied and its carry-over (or "residual") effect during later periods.

AFCARRYOVER can be used to construct the factors required to represent the carry-over effects. To do this it requires factors to identify the treatment, subject (or plot) and period corresponding to each unit of the data. These must be specified by the TREATMENTS, SUBJECTS and PERIODS parameters respectively. The NONELEVEL option can be used to supply a scalar to specify the level, or a single-line text to specify the label, to be used to represent the "no carry-over" treatment which occurs during the first period. If the level is not specified, an appropriate value is found automatically: zero if that is not an existing treatment level, or the minimum treatment level minus one otherwise.

Two factors can be generated. The CARRYOVERFACTOR parameter provised a factor that simply identifies the carry-over treatment on each unit, while the NOCARRYOVER parameter provides a factor representing the comparison between the units with no carry-over and those with any type of carry-over (essentially this is a comparison between the periods 2 onwards where there were carry-over effects from earlier times, and period 1 where there was none). The NOCARRYOVER factor may be required to be able to analyse the design using ANOVA (see the description of procedure AGCROSSOVERLATIN).

Option: NONELEVEL.
Parameters: TREATMENTS, SUBJECTS, PERIODS, CARRYOVERFACTOR, NOCARRYOVER.

Method

AFCARRYOVER generates the factors using the standard Genstat calculation and manipulation commands.

See also

Directive: ANOVA.
Procedure: AGCROSSOVERLATIN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AFCOVARIATES

Defines covariates from a model formula for ANOVA (R.W. Payne).

Options

COVARIATES = *pointer*	Saves the covariates
COVGROUPS = *pointer*	Saves the pointers defined to contain the covariates formed for each term in TERMS
FACTORIAL = *scalar*	Limit on number of factors in the model terms formed from TERMS; default 3

Parameters

TERMS = *formula*	Model terms from which to define covariates

Description

Analysis of covariance is performed in Genstat using the ANOVA directive. The treatment model must be specified first, using the TREATMENTSTRUCTURE directive, and the underlying structure of the design (or, equivalently, the error terms for the analysis) is specified using the BLOCKSTRUCTURE directive as in ordinary analysis of variance. The extra step for analysis of covariance is to specify the covariates for the analysis using the COVARIATE directive. The covariates must be continuous variables, and so COVARIATE requires a list of variates. Alternatively, a refinement introduced in Release 12 allows you to put some of the covariates into pointers. The covariates in each pointer will then be pooled into a single line in the analysis of variance table.

However, COVARIATE does not allow for more complicated situations. For example you might want to fit a different covariate regression coefficient within each block of a randomized-block experiment, or to use the covariate to fit the effects of terms in an unbalanced design.

The AFCOVARIATES procedure has therefore been provided as an alternative to the COVARIATE directive, to allow you to specify a model formulae to define the terms to be fitted as covariates in the analysis. The model formula is specified by the TERMS parameter, using the same conventions as for example in the Genstat regression commands. The dummy variables that are generated to represent the model terms in the formula use the same parameterization as the regression commands; see Section 3.3.2 of the *Guide to the Genstat Command Language, Part 2 Statistics* for details.

So, for example, you can fit a different regression coefficient for the variate X within each block defined by the factor Blocks, by specifying

```
AFCOVARIATES Blocks.X
```

The COVARIATES option allows you to supply a pointer to store the covariates that are calculated (otherwise they will be unnamed, and thus usable only by later ANOVA commands). The covariates are grouped into a pointer for each model term specified by TERMS. The COVGROUPS option allows you to supply a pointer to store these pointers (otherwise they too will be unnamed, and thus usable only by later ANOVA commands). Each covariate is each defined with an extra text, using the EXTRA parameter of the VARIATE directive, to indicate the parameter that it represents. Also the IPRINT option of VARIATE is set to extra, so that this extra text will be used in output instead of the identifier of the covariate itself. Similarly, the COVGROUPS pointers are given extra texts indicating the model term that each one represents.

The FACTORIAL option sets a limit on the number of factors or variates in each of the terms formed from the TERMS formula. Any term containing more than that limit is deleted.

Options: COVARIATES, COVGROUPS, FACTORIAL.
Parameter: TERMS.

Method
AFCOVARIATES defines the covariates from a design matrix constructed using the TERMS directive.

Action with RESTRICT
AFCOVARIATES takes account of any restrictions on the factors or variates in the TERMS formula.

See also
Directives: COVARIATE, ANOVA.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AFCYCLIC

Generates block and treatment factors for cyclic designs (R.W. Payne).

Option

PRINT = *string token*	Whether to print the design (design); default * i.e. no printing

Parameters

INITIALBLOCKS = *variates* or *pointers*	Defines one (variate) or more (pointer to variates) initial blocks for a treatment factor
INCREMENT = *scalars* or *pointers*	Defines the size of the successive increment (scalar) or increments (pointer to scalars) for each initial block
LEVELS = *scalars* or *variates*	Defines the levels of each treatment factor; this need not be specified if the factor has already been declared
SEED = *scalar*	Seed to be used to randomize each design, if required
TREATMENTS = *factors*	Specifies treatment factors
BLOCKS = *factors*	Specifies block factors
UNITS = *factors*	Specifies factors to index the units within each block

Description

The cyclic method is a powerful way of constructing incomplete block designs. In its simplest form, it starts with an initial block, containing some subset of the treatments. This subset is then represented by the ordinal number in the range 0...$m-1$ where m is the number of treatment levels. The second and subsequent blocks are then generated by successive addition modulo m of one to the numbers in the subset. Thus, for seven treatments (0...6) and an initial block (0,1,4), the subsequent blocks would contain treatments (1,2,5), (2,3,6), (3,4,0), (4,5,1), (5,6,2) and (6,0,3). As can be seen, if m is a prime number, m blocks are generated with each initial block. However, if m can be expressed as the product of other integers, shorter cycles can occur. For example, for m=8 and initial block (0,1,4,5), four blocks are generated altogether, the others being (1,2,5,6), (2,3,6,7) and (3,4,7,0). The procedure allows for all of this. It is also possible to have more than one initial block, and the increment need not be one.

The INITIALBLOCKS parameter specifies the initial blocks. If the design is to be generated from a single initial block, INITIALBLOCKS should be set to a variate containing the levels corresponding to the treatments concerned; if there are several, the appropriate variates should be placed into a pointer. Similarly the INCREMENT parameter, which specifies the increment to be used, should be set to a scalar if the same increment is to be used for all the initial blocks, otherwise to a pointer of scalars. The levels of the treatment factor are specified by the LEVELS parameter and the SEED parameter allows the design to be randomized. As is customary in Genstat, if LEVELS is set to a scalar the levels are assumed to be represented by the integers 1 upwards, but LEVELS can be set to a variate to specify other numbers. LEVELS can be omitted if the TREATMENTS parameter is used to supply a factor to store the treatments, provided the levels of that factor have already been defined outside the procedure. The factors for blocks and units within blocks can be saved similarly by the BLOCKS and UNITS parameters respectively. The design can also be printed, by setting option PRINT=design.

The properties of the cyclic designs that can be generated for any particular number of treatments or size of block varies according to the choice of initial block and increment. Tables showing the most efficient combinations have been presented for example by John, Wolock & David (1972), John (1981, 1987) and Lamacraft & Hall (1982).

Option: PRINT.

Parameters: INITIALBLOCKS, INCREMENT, LEVELS, SEED, TREATMENTS, BLOCKS.

Method

The procedure generates the design using the standard Genstat directives for calculation and manipulation.

References

John, J.A., Wolock, F.W. & David, H.A. (1972). *Cyclic Designs*. National Bureau of Standards, Applied Mathematics Series 62.

John, J.A. (1981). Efficient cyclic designs. *Journal of the Royal Statistical Society Series B*, **43**, 76-80.

John, J.A. (1987). Cyclic Designs. Chapman & Hall, London.

Lamacraft, R.R. & Hall, W.B. (1982). Tables of incomplete cyclic block designs: r=k. *Australian Journal of Statistics*, **24**, 350-360.

See also

Procedure: AGCYCLIC.

Genstat Reference Manual 1 Summary sections on: Design of experiments, REML analysis of linear mixed models.

AFDISCREPANCY

Calculates the discrepancy of a design (B.M. Parker).

Options

PRINT = *string tokens*	Controls whether to print the discrepancy (results); default resu
METHOD = *string token*	Specifies the method to use to calculate the discrepancy (L2, maximin, entropy); default L2
SWAP = *variate*	A variate of length two indicating which design points have swapped when updating the discrepancy criterion for the maximin or entropy criteria; default none

Parameters

DESIGN = *matrices* or *pointers*	A matrix, or a pointer of variates, specifying the design points
DISCREPANCY = *scalars*	Saves the discrepancy
DISTANCES = *matrices*	Stores the distances, to allow fast updates with the maximin or entropy criteria

Description

A space filling design is an experimental design for a number of runs, which each have a number of (usually) continuous factors. They are designed to ensure that the experiment is spread over the entire design space, so that large and potentially important regions are not ignored. AFDISCREPANCY can calculate a measure of the *discrepancy* of the design, that indicates how well it fills the space. This is used by the AGSPACEFILLINGDESIGN procedure to form a good design, that is, one with a low discrepancy.

The DESIGN parameter supplies either as a matrix with n rows and m columns, or a pointer with n variates each with m units, to specify a design with n points in a unit hypercube $[0,1]^m$.

The METHOD option specifies the criterion to use to measure the discrepancy of the design. The maximin criterion maximizes the minimum inter-point Euclidean distance. The entropy criterion minimizes $-\log|R|$, where R is a measure of correlation between points in the design. The L_p discrepancy is a measure of non-uniformity of a design. More precisely, the L_p discrepancy measures the difference between the empirical cumulative distribution function of a design and the uniform cumulative distribution function. Here, we minimize the centred L_2 discrepancy. (See Fang *et al.* 2000.)

The DISTANCES option can supply a matrix to store a measure of the distance between the points in the designs for the maximin and entropy criteria. If a variate of two numbers is specified by the SWAP option, AFDISCREPANCY will update the distance criterion only for the design points that are changed, making a far faster procedure. This is used in the ESE algorithm adapted in AGSPACEFILLINGDESIGN.

By default the discrepancy is printed, but you can set option PRINT=* to suppress this. The discrepancy can be saved, in a scalar, using the DISCREPANCY option.

Options: PRINT, METHOD, SWAP.
Parameters: DESIGN, DISCREPANCY, DISTANCES.

Method

The maximin design maximizes the minimum Euclidean distance between points as described in Johnson *et al.* (1990). The entropy design maximizes $|nR|$ where R is a Gaussian correlation matrix between design points. Thus, here we minimize a Gaussian correlation function. In a Bayesian context, minimizing the expected posterior entropy is equivalent to maximizing the

prior entropy. See Koehler & Owen (1996). *R* here, for design points *i* and *j*, is defined as
$\exp(\sum_{k=1}^{m} |(x_{ik} - x_{jk})|)^2$
The L_2 discrepancy is calculated according to the procedure of Hickernell (1988).

References

Fang, K.T., Lin, D.K., Winker, P. & Zhang, Y. (2000). Uniform design: theory and application. *Technometrics*, **42**, 237-248.

Hickernell, F. (1998). A generalized discrepancy and quadrature error bound. *Mathematics of Computation of the American Mathematical Society*, **67**, 299-322.

Johnson, M.E., Moore, L.M. & Ylvisaker, D. (1990). Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, **26**, 131-148.

Koehler, J.R. & Owen, A.B. (1996). Computer experiments. *Handbook of Statistics*, **13**, 261-308.

See also

Procedure: AGSPACEFILLINGDESIGN.

AFFYMETRIX

Estimates expression values for Affymetrix slides (D.B. Baird).

Options

PRINT = *string tokens*	What to print (estimates, background, monitoring); default para
METHOD = *string token*	Method for calculating probe expression values (mas4, mas5, rma, rma2); default rma
BMETHOD = *string token*	Method to use for background values (mean, quantile, none); default mean for METHOD settings mas4 and mas5, but none for settings rma and rma2
BWEIGHTING = *string token*	Method for weighting background grids (affymetrix, distance); default affy
TRANSFORMATION = *string token*	How to transform the data (log2, none); default log2
NMETHOD = *string token*	Method for normalization i.e. whether to use a mean, median or geometric mean for the averaged normalized distribution (means, medians, geometricmeans, none); default mean
REPLACEDATA = *string token*	Whether to replace the DATA variates with background corrected intensities (yes, no); default no
SPREADSHEET = *string token*	What to save in a spreadsheet (results); default * i.e. nothing
MAXCYCLE = *scalar*	Maximum number of iterations; default 50
TOLERANCE = *scalar*	Tolerance for convergence; default 0.0001

Parameters

DATA = *variates*	Intensities to be analysed
SLIDES = *factors*	Identify the slides (or chips)
PROBES = *factors*	Identify the probes (or genes) within each slide
ATOMS = *factors*	Identify the PM/MM pairs within each probe
PMMM = *factors*	Distinguish between PM and MM values
TYPEPROBES = *factors*	Defines the probe-type corresponding to each intensity
ROWS = *factors*	Identifies rows within each slide (required only if background corrections are to be made)
COLUMNS = *factors*	Identifies columns within each slide (required only if background corrections are to be made)
ESTIMATES = *variates*	Saves the estimated expression values for each slide and probe combination
SE = *variates*	Saves approximate standard errors for the estimates
IDSLIDES = *factors*	Saves factors to identify the slides in the ESTIMATES variates
IDPROBES = *factors*	Saves factors to identify the probes in the ESTIMATES variates

Description

AFFYMETRIX estimates expression values over the perfect match (PM) and mismatch (MM) pairs for each probe on Affymetrix slides (or chips). On Affymetrix chips, each probe has 8-20 pairs of DNA sequences with a central base changed between the perfect match and mismatch sequences. The value for the probe level of expression is taken as an average over the pairs of perfect match (PM) and mismatch (MM) spots. The intensity values are obtained by reading in a series of Affymetrix CEL files, and the chip information from a CDF file.

The METHOD option selects the method to use to summarize over the PM and MM pairs, with settings:

rma	Robust Means Analysis model – the probe level model introduced by Irizarry *et al*. (2003) which only uses PM information and transforms the values based on a kernel density estimate of the PM distribution;
rma2	Robust Means Analysis 2 – an adaptation of RMA algorithm which fits the kernel density to a truncated distribution of the PM values, with the truncation point based on an initial kernel density estimate;
mas4	Affymetrix Version 4 – the AvDiff algorithm introduced in the Affymetrix version 4 software; and
mas5	Affymetrix Version 5 – the Tukey biweight algorithm introduced in the Affymetrix version 5 software.

In the Affymetrix MAS 4 and 5 methods, the difference between the signals (PM – MM) is averaged using a robust averaging method. The MAS 4 algorithm uses the AvDiff algorithm which discards the minimum and maximum difference, and any differences greater than 3 standard deviations from the mean. The MAS 5 algorithm uses the Tukey biweight algorithm which reweights the values depending on how far they are from the median, and discards any that are more than 5 times the median absolute distance away. The MAS 5 algorithm also replaces the MM value with a value known as an Ideal Mismatch (IM), which is always less than the PM value.

The standard RMA algorithm would normally use the $\log_2$ transformed PM values with no background correction, which then have a quantile normalization applied to them. The adjusted PM values then have a Normal function transformation applied to them with the values for the transformation being calculated from a kernel density estimate applied to the adjusted PM values. Finally the transformed PM values are summarized with a median polish of the slides by atom values for each probe. The $\log_2$ transformation can be suppressed by setting option TRANSFORMATION=none.

The RMA model performs a background correction by fitting a two component model to the PM intensities:

Observed intensity = *Signal* + *Noise*

where *Signal* has an exponential distribution with parameter α (the reciprocal of the mean), the *Noise* has an Normal distribution with parameters μ (the mean) and σ (the standard deviation). α, μ and σ are then estimated and the expected value of the signal is estimated, given the observed value of the intensity.

For all algorithms, the lowest 2% of spots on each slide can be used to estimate a background correction for the intensities. The chip is divided into 16 zones in a 4 × 4 grid, and each spot has a weighted average of these 16 levels removed from it. The levels used are controlled by the BMETHOD options, with settings:

means	the means of the values below the 2% quantile are used as the background levels;
quantiles	the actual 2% quantiles are used as the background levels; and
none	if you want no background correction to be made.

The BWEIGHTING option controls how the background levels are combined before removing them from each spot:

affymetrix	the weights are 1/(*squared-distance* + 100); and
distance	the weights are 1/(min(*squared-distance*, 100),

where *Squared-distance* = (*distance from the spot to the zone centroid*)^2.

The quantile normalization of the PM/MM values on each slide is controlled by the NMETHOD

option. Its settings select the way in which the overall distribution is produced from the cumulative density functions on each slide:

`means`	takes the means;
`medians`	takes the medians; and
`geometricmeans`	takes geometric means (i.e. the mean on the log scale, back-transformed to the natural scale); and
`none`	if you do not want any quantile normalization.

The intensity values are specified by the `DATA` parameter. If these are in a single variate, the `SLIDE` parameter should supply a factor to index the slides, and the `PROBES` parameter should supply a factor to index the probes (or genes). Alternatively you can supply a pointer containing a variate for each slide. The slides factor is then not required; if it is given it should just have one entry for each slide in the order of the variates in the pointer. The `PROBES` factor is that for a single slide, and all slides must have a common layout.

The `ATOMS` parameter supplies a factor to identify the PM/MM pairs within each probe, and the `PMMM` parameter supplies a factor, with levels labelled `'PM'` and `'MM'`, to distinguish between PM and MM values. The `TYPEPROBES` parameter supplies a factor to specify the probe types. The types of probes that can occur on Affymetrix chips are: `'Expression'`, `'Genotyping'`, `'CustomSeq'`, `'Tag'`, `'Unknown'`, `'Checkerboard Negative'`, `'Checkerboard Positive'`, `'Hybridization Negative'`, `'Hybridization Positive'`, `'Text Negative'`, `'Text Positive'`, `'Central Negative'`, `'Central Positive'`, `'Gene Exp Negative'`, `'Gene Exp Positive'`, `'Cycle Fidelity Negative'`, `'Cycle Fidelity Positive'`, `'Central Cross Negative'`, `'Central Cross Positive'`, `'Cross Hyb Negative'` and `'Cross Hyb Positive'`.

The `ROWS` and `COLUMNS` parameters can supply factors to identify the rows and columns within each slide. These are required only if background corrections are to be made.

The `ESTIMATES` parameter must supply a variate to save the estimated expression value for each slide and probe combination. The `IDPROBES` and `IDSLIDES` parameters must supply factors to identify the probes and slides, respectively, in the `ESTIMATES` variate. You can also set parameter `SPREADSHEET=results` to save these in a Genstat spreadsheet. The SE parameter can supply a variate to save approximate standard errors and, if this is set, the standard errors are included in the spreadsheet.

Options: `PRINT`, `METHOD`, `BMETHOD`, `BWEIGHTING`, `TRANSFORMATION`, `NMETHOD`, `REPLACEDATA`, `SPREADSHEET`, `MAXCYCLE`, `TOLERANCE`.

Parameters: `DATA`, `SLIDES`, `PROBES`, `ATOMS`, `PMMM`, `TYPEPROBES`, `ROWS`, `COLUMNS`, `ESTIMATES`, `SE`, `IDSLIDES`, `IDPROBES`.

References

Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U. & Speed, T.P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, Number 2, 249-264.

See also

Procedures: `FDRBONFERRONI`, `FDRMIXTURE`, `MAANOVA`, `MABGCORRECT`, `MAEBAYES`, `MAREGRESSION`, `MARMA`, `MAROBUSTMEANS`, `MAVDIFFERENCE`, `MAVOLCANO`, `QNORMALIZE`.

Genstat Reference Manual 1 Summary section on: Microarray data.

AFIELDRESIDUALS

Display residuals in field layout (R.W. Payne & A.D.Todd).

Options

PRINT = *string tokens*	Controls output (contour, shade, table); default cont
GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
METHOD = *string token*	Type of residuals to take from the save structure when the RESIDUALS parameter is not specified (combined, finalstratum, standardizedfinal); default comb
MARGIN = *string token*	Whether to include margins in printed tables (yes, no); default no
YORIENTATION = *string token*	Y-axis orientation of the plot (reverse, normal); default norm
PENCONTOUR = *scalar*	Pen number to be used for the contours; default 1
PENFILL = *scalar* or *variate*	Pen number(s) defining how to fill the areas between contours; default 3
PENSHADE = *scalar* or *variate*	Pen(s) to use for the shade plot; default 3

Parameters

Y = *variates* or *factors*	Specifies the y-coordinates of the plots
X = *variates* or *factors*	Specifies the x-coordinates of the plots
RESIDUALS = *variates*	Residuals to be plotted; default is to take the residuals from the save structure specified by the SAVE option, or from the most recent ANOVA if that is unspecified
SAVE = *ANOVA, REML or regression save structures*	Save structure of the ANOVA, REML or regression analysis from which to take the residuals if the RESIDUALS parameter is not specified; default is to take the most recent ANOVA analysis
FIELDWIDTH = *scalars*	Field width for printing the residuals; default 12
DECIMALS = *scalars*	Number of decimal places to use when printing the residuals
TITLE = *texts*	Titles for the plots

Description

In a field experiment it can be useful to study the spatial pattern of the residuals, for example to see if there are any systematic trends in fertility.

The locations of the plots are defined by the Y and X parameters, specifying variates or factors containing their y- and x-coordinates respectively. The residuals can be supplied, in a variate, by the RESIDUALS parameter. If this is not set, the default is to take the residuals from the most recent ANOVA analysis. You can take the residuals from some other analysis, by specifying its save structure using the SAVE parameter. This can be from another ANOVA analysis, a REML analysis or a regression analysis (see MODEL).

The METHOD option determines the type of residuals that are taken. The default setting combined gives residuals combining the residuals from all the strata or error terms in the analysis. These corresponds to the CBRESIDUALS option of the AKEEP directive, or the use of the RESIDUALS option in VKEEP with option RMETHOD=all. Regression allows only a single error term, so combined is treated as the same as the next setting, finalstratum.

The setting finalstratum uses simple residuals from the final stratum or error term. These

correspond to the RESIDUALS option of AKEEP with option RMETHOD=simple, or the RESIDUALS option of VKEEP with option RMETHOD=final, or the RESIDUALS parameter of RKEEP with option RMETHOD=simple.

The last setting, standardizedfinal, uses standardized residuals from the final stratum or error term. These correspond to the RESIDUALS option of AKEEP with option RMETHOD=standardized, or the RESIDUALS parameter of RKEEP with option RMETHOD=deviance. They are calculated using standard errors from procedure VFRESIDUALS for REML analyses.

Usually, the plots will all have different coordinates. However, if there are several plots with the same coordinates, mean residuals are calculated for each location. Thus for example, if you wanted only to look at the block and whole-plot residuals in a split-plot design, you could request combined residuals and then set identical coordinates for the (sub-) plots within each whole plot.

AFIELDRESIDUALS provides three forms of representation, selected using the PRINT option as follows:

table	prints the residuals in a table whose structure corresponds to the field layout,
contour	generates a contour plot if the plots are on a regular grid or a line graph if they are arranged in a single line, and
shade	can produce a shade plot for plots that are on a regular grid.

The GRAPHICS option determines the type of graphics that is used, with settings highresolution (the default) and lineprinter. No graph can be produced if the plots are in an irregular 2-dimensional arrangement. High-resolution contour plots require more than 3 rows and columns, and line-printer contour plots require more than 4 rows and columns. The way in which the lines are drawn in high-resolution contour plots is defined by the properties of the pen specified by the PENCONTOUR option, while the pen specified by the PENFILL parameter defines how to shade the areas between the contours. Their defaults are 1 and 3 respectively. Similarly, the pen or pens specified by the PENSHADE option control the colouring of the shade plot; the default is to use pen 3. For more information see the DCONTOUR and DSHADE directives.

The MARGIN option, with settings no (default) and yes, determines whether or not marginal summaries are included with the printed tables. The FIELDWIDTH and DECIMALS parameters can be used to specify the formats of the printed tables (as in the PRINT directive). The TITLE parameter can supply a title for the plots. If this is unset, a default title is formed.

The YORIENTATION option controls the orientation of the y-coordinates in the plots and tables. By default this is normal, so that they run upwards from the bottom of the page (as in a map).

Options: PRINT, GRAPHICS, METHOD, MARGIN, YORIENTATION, PENCONTOUR, PENFILL, PENSHADE.
Parameters: Y, X, RESIDUALS, SAVE, FIELDWIDTH, DECIMALS, TITLE.

Method

AFIELDRESIDUALS obtains the residuals using the AKEEP, VKEEP or RKEEP directives, and then uses the standard Genstat facilities for manipulation and plotting.

Action with RESTRICT

If any of X, Y or RESIDUALS is restricted, only the unrestricted plots are displayed.

See also
Directive: ANOVA.
Procedures: AGRAPH, APLOT, VDFIELDRESIDUALS.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AFLABELS

Forms a variate of unit labels for a design (R.W. Payne).

Options

UNITLABELS = *variate*	Stores the labels
MAXDIGIT = *scalar*	Number of available digits; default 8

Parameters

FACTOR = *factors*	Factors indexing the units of the design; if this is unset, the factors from the most recent BLOCKSTRUCTURE command are used
NEWLEVELS = *variates*	Allows new levels to be specified for each FACTOR; if this is unset, uses the levels already defined for the factor

Description

AFLABELS forms a variate, specified using the UNITLABELS option, containing a unique code for each unit of a design. By default, it is assumed that the codes can be up to eight digits long, but this can be modified using the MAXDIGIT option.

The units are assumed to be indexed by a set of factors which can be specified by the FACTOR parameter; if this is not set, AFLABELS takes those from the most recent BLOCKSTRUCTURE command (if any). By default, the codes are formed from the levels of the factors but, if these are unsuitable, alternative levels can be supplied using the NEWLEVELS parameter. In particular, AFLABELS requires the levels (or new levels) all to be positive integers.

Options: UNITLABELS, MAXDIGIT.
Parameters: FACTOR, NEWLEVELS.

Method

The codes are formed by ordinary arithmetic so that the initial digits are the levels of the first indexing factor, then the second, and so on. If there is too much information to fit within the MAXDIGIT limit, AFLABELS tries to decrease the sizes of the codes by successively combining the final pairs of factors.

See also

Procedure: AFUNITS.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AFMEANS

Forms tables of means classified by ANOVA treatment factors (R.W. Payne).

Options

PRINT = *string tokens*	What to print (means, sed, sedsummary, ese, lsd, lsdsummary); default mean, sed
MEANS = *table*	Saves means; default *
SED = *symmetric matrix*	Saves matrices of standard errors of differences between means; default *
ESE = *table*	Saves effective standard errors; default *
LSD = *symmetric matrix*	Saves least significant differences between means; default *
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
DFMEANS = *symmetric matrices*	Saves degrees of freedom for comparisons between every pair of entries in the table of means
EQFACTORS = *factors*	Factors whose levels are to be assumed to be equal within the comparisons between means, when calculating effective standard errors
SAVE = *ANOVA save structure*	Save structure to provide the table of means; default uses the save structure from the most recent ANOVA

Parameter

CLASSIFY = *vectors*	Factors to classify table of means (from those in the TREATMENTSTRUCTURE in the ANOVA analysis)

Description

AFMEANS calculates and prints tables of predicted means classified by treatment factors from an ANOVA analysis. It uses the same method as ANOVA itself, but with the extension that the term defined by the full list of factors need not have been included in the analysis. So, for example, you can obtain an A × B table of means, even if the model contained only the A and B main effects. Alternatively, in a more realistic scenario, you may have significant A.B and B.C interactions, but no A.B.C interaction. You might then still want to present an A × B × C table means, even though you might not want to include an A.B.C interaction.

The factors classifying the table of means are specified by the CLASSIFY parameter. By default the means are formed for the most recent ANOVA, but you can use the SAVE option to supply the save structure from an earlier analysis.

Printed output is controlled by settings of the PRINT option:

means	means,
ese	effective standard errors of the means,
sed	standard errors for differences between the means,
sedsummary	summary of the standard errors for differences between the means,
dfmeans	degrees of freedom for the standard errors of differences between means,
lsd	least significant differences between the means, and
lsdsummary	summary of the least significant differences between the means.

The default is to print means and a summary of the standard errors of differences. Note: if all the differences between means have the same standard error of difference, a summary is printed for the settings sed and lsd, instead of the full symmetric matrix of values. The LSDLEVEL option

specifies the significance level (%) to use in the calculation of least significant differences (default 5%). The EQFACTORS option allows you to specify factors within the tables of means whose levels are assumed to be equal for the two means, when calculating effective standard errors.

The MEANS, SED, ESE, LSD and DFMEANS options allow the results to be saved in appropriate Genstat data structures.

Options: PRINT, MEANS, SED, ESE, LSD, LSDLEVEL, DFMEANS, EQFACTORS, SAVE.
Parameter: CLASSIFY.

See also
Directive: ANOVA.
Procedure: AUPREDICT.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AFNONLINEAR

Forms D-optimal designs to estimate the parameters of a nonlinear or generalized linear model (W. van den Berg).

Options

PRINT = *string token*	Controls printed output (results, monitoring); default resu, moni
PLOT = *string token*	Controls whether to plot the design (design); default desi
YARGUMENT = *identifier*	Data structure that stores the results of the function when it is calculated by expressions supplied by the FUNCTION option; must be set
XARGUMENT = *identifier*	Data structure representing the x-variate in the expressions supplied by the FUNCTION option; must be set
FUNCTION = *expression structures*	Specifies the function whose parameters are to be estimated; must be set
FNDERIVATIVES = *expression structures*	Specifies expressions to calculate derivative of the function with respect to each parameter; must be set
ITERATIVEWEIGHTS = *identifier*	Data structure that stores the iterative weights in the expressions supplied by the FNITERATIVEWEIGHTS option
FNITERATIVEWEIGHTS = *expression structures*	Specifies expressions to calculate the iterative weights when estimating the parameters of a generalized linear model
XSUPPORT = *variate*	Supplies the support points for the initial design, and saves those of the final design; if no initial values are supplied, an initial design is formed at random
XWEIGHTS = *variate*	Supplies the weights for the support points for the initial design, and saves those of the final design; if no initial values are supplied, equal weights are used initially
GRID = *variate*	Specifies the grid points where the design will be evaluated
A0 = *scalar*	Initial update weight; default 0.1
SEED = *scalar*	Seed for the random numbers used to select the initial design when not supplied by XSUPPORT and XWEIGHTS
NCYCLE = *scalar*	Number of iterations to make between at each value of A0, before halving it for the next batch of iterations; default 100
MAXCYCLE = *scalar*	Maximum number of iterations; default 2500
TOLERANCES = *variate*	Variate with two values specifying the convergence criterion and the tolerance for zero weights; default !(1.E-6, 1.E-5)

Parameters

PARAMETER = *scalars*	Parameters of the nonlinear or generalized linear model (with values giving an indication of their likely estimated values)
DERIVATIVE = *identifiers*	Data structures that store the results of the calculation of

the derivative for each parameter, in the expressions specified by the FNDERIVATIVES option

Description

AFNONLINEAR constructs a design for estimating the parameters of a nonlinear or generalized linear model involving a single continuous variable *x*. The aim is to find the best values of *x* (i.e. the best *support points*) at which to observe the model, and a weight for each one. The design should then contain replicate observations at each of the support points, with the numbers of replicates in the same proportions as their weights. Suppose, for example, we have support points 1, 2 and 4, with weights 0.25, 0.25 and 0.5. A suitable design might then consist of observations at x-values 1, 2, 4 and 4 (i.e. 4 should have twice the replication of either 1 or 2). The designs that are produced are known as *continuous* designs, as the weights are not constrained to give an exact integer partitioning of the available points for any specific design size *N*. Instead you need to round *N* multiplied by each weight to the nearest feasible integer.

The model is specified in one, or more, expression structures by the FUNCTION option. The YARGUMENT gives the identifier of the data structure that receives the result of the function in the expressions, and the XARGUMENT gives the identifier of the data structure that provides the *x*-values. For example, we could define the negative exponential model

$$y = e^{(-b \times x)} + c$$

by

```
EXPRESSION Func; VALUE=!e( Y = EXP(-1*B*X) + C)
AFNONLINEAR [FUNCTION=Func; YARGUMENT=Y; XARGUMENT=X; ...
```

Notice that the data structures X and Y do not need to be declared. AFNONLINEAR simply needs to know which they are within the expression, so that it can replace them by the sets of *x*- and *y*-values that it really needs (using the REFORMULATE directive).

The parameters of the model (here B and C) must be specified by the PARAMETER parameter. These must be scalars, with values that give an indication of their likely estimated values. AFNONLINEAR also needs to be able to calculate the derivative of the function with respect to each parameter. You must specify expressions to do this using the FNDERIVATIVES option, and indicate the data structures that will receive the results of the calculations using the DERIVATIVE parameter. So, for the negative exponential above, we need

```
EXPRESSION  Gfunc[1,2]; VALUE=!e( GradB = -1*X*EXP(-1*B*X) ),\
                                !e( GradC = 1 )
AFNONLINEAR [FUNCTION=Function; YARGUMENT=Y; XARGUMENT=X;\
             FNDERIVATIVE=Gfunc[]; XSUPPORT=X; XWEIGHTS=W;\
             GRID=Grid] PARAMETER=B,C; DERIVATIVE=GradB,GradC
```

The GRID option defines the *x*-values at which the design is evaluated. These should cover the range of feasible *x*-values.

The XSUPPORT option saves the support points of the design, in a variate. If the variate has values already defined on entry to AFNONLINEAR, these are used to provide the support points for the initial design where AFNONLINEAR begins its search. Otherwise AFNONLINEAR chooses an initial design at random by selecting *m* points at random from the grid points, where *m* is twice the number of parameters in the model. The SEED option specifies a seed for the random numbers that are used to make the selection. The default value of zero continues an existing sequence of random numbers if any have already been used in the current Genstat job, or obtains a random seed using the system clock if none have been used already.

The XWEIGHTS option saves the weights of the support points, in a variate, and can supply weights for an initial design. Otherwise AFNONLINEAR starts with equal weights.

To form designs for generalized linear models, you also need to supply expressions to calculate the iterative weights at various *x*-values. The FNITERATIVEWEIGHTS option supplies the expressions, and the ITERATIVEWEIGHTS option indicate the data structure that will receive

the results of the calculations.

By default AFNONLINEAR produces a plot showing the function and prediction variance at the selected grid points, but you can suppress this by setting option PLOT=*.

AFNONLINEAR uses the algorithm of Federov (1972). This involves a sequence of iterations in which a new support point may be added, or the weight of an existing point may be increased. The A0 option specifies the weights to be given to a new point, or to be added to an existing point. (The weights of the other support points are then decreased, proportionally, so that the weights still add up to one.) The NCYCLE option controls how many iterations are made with each value of AO (default 100); so, at the end of each set of NCYCLE iterations, AO is divided by two in order for the weights to converge to a stable solution.

The TOLERANCES option can be set to a variate of length two, to specify the convergence criterion and the tolerance for zero weights (defaults 10^{-6} and 10^{-5}, respectively). The algorithm stops when the number of support points equals the number of parameters, and the prediction variance minus the number of parameters is less than the first TOLERANCES value. Weights less than the second TOLERANCES value are set to zero at each iteration (so that the corresponding points leave the design).

Options: PRINT, PLOT, YARGUMENT, XARGUMENT, FUNCTION, FNDERIVATIVES, ITERATIVEWEIGHTS, FNITERATIVEWEIGHTS, XSUPPORT, XWEIGHTS, GRID, A0, SEED, NCYCLE, MAXCYCLE, TOLERANCES.
Parameters: PARAMETER, DERIVATIVE.

Method

For a D-optimal design the number of support points is equal to the number of parameters, and the determinant of the information matrix is at its maximum. Instead of maximizing the information matrix, AFNONLINEAR uses the General Equivalence Theorem. By minimizing the maximum prediction variance of the response a G-optimal design is obtained, with the maximum prediction variance of the response equal to the number of parameters and occurring at the support points. According to the General Equivalence Theorem the design will also be D-optimal.

Fedorov (1972) proposes starting with a value for A0 of e.g. 0.1, and dividing A0 by two after each iteration so that the determinant of the information matrix is lower than the determinant of the information matrix of the preceding iteration. AFNONLINEAR allows you to set the initial value of A0 (option A0 default 0.1), and the number of iterations to make before dividing it by two (option NCYCLE, default 100).

Sometimes the weight of a support point may be divided over two neighbouring points of the grid, and the algorithm may fail to converge. In that case you could replace the two support points by a new point with a weight equal to the total of their two weights, and located at their average. Alternatively, you could change the SEED option to run AFNONLINEAR again from a different starting design.

Reference

Fedorov, V.V. (1972). *Theory of Optimal Experiments*. Academic Press, New York & London.

See also

Directive: AFRESPONSESURFACE.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AFORMS

Prints data forms for an experimental design (R.W. Payne).

Options

BLOCKSTRUCTURE = *formula*	Defines the block factors to be used to label the units of the design; default takes those specified in an earlier BLOCKSTRUCTURE directive
TREATMENTSTRUCTURE = *formula*	Defines the treatment factors to be used, if any, to label the forms
NLINES = *scalar*	Number of lines to be allowed for each measurement; default 1

Parameters

LABEL = *texts*	Labels for the measurements to be recorded on the forms
FIELDWIDTH = *scalar*	Fieldwidth to be allowed for each label

Description

AFORMS prints data forms which can be used to record data from an experimental design. Several measurements can be recorded, in separate columns across the page, and space is provided for a row of values for each unit of the design. The block factors to label the units can be supplied by setting the BLOCKSTRUCTURE option to the block formula of the design. If this is not set AFORMS will use the formula, if any, defined previously by the BLOCKSTRUCTURE directive.

The units can also be labelled with the treatments that have been used in the design, by setting the TREATMENTSTRUCTURE option to the appropriate treatment formula. However, to guard against bias, experimenters will often prefer not to know which treatments were applied to each unit when recording the results, so if this is omitted no treatment information is included.

The LABEL parameter supplies the column label to identify each column of measurements, and the FIELDWIDTH parameter can specify the width of the column. By default, a single line is provided for row of measurements but this can be increased using the NLINES option.

Options: BLOCKSTRUCTURE, TREATMENTSTRUCTURE, NLINES.
Parameters: LABEL, FIELDWIDTH.

Method

AFORMS uses the standard Genstat directives for printing and manipulation.

Action with RESTRICT

AFORMS needs to use RESTRICT in order to organise the labelling of the forms, and so any existing restrictions will be cancelled.

See also

Procedure: ADSPREADSHEET.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AFPREP

Searches for an efficient partially-replicated design (R.W. Payne).

Options

PRINT = *strings*	Controls printed output (design, efficiency, factors, monitoring); default * i.e. none
LEVELS = *scalar* or *variate*	Levels of the treatment factor; if unset, takes the levels declared for the factor specified by the TREATMENTS option
NREPEATS = *variate*	Number of times each treatment occurs in the design
NBLOCKS = *scalar*	Number of blocks
TREATMENTS = *factor*	Treatment factor
BLOCKS = *factor*	Block factor
UNITS = *factor*	Unit-within-block factor
EFFICIENCY = *variate*	Saves the efficiency factors of the treatment term within blocks
NSTARTS = *scalar*	Specifies the number of random starting configurations to take in the search for the best design; default 5
NTRIES = *scalar*	Number of designs to try from each starting configuration; default 20
SEED = *scalar*	Seed for the random numbers used to randomize the design; default 0
TRYSEED = *scalar*	Seed for the random numbers used to select the random starting configurations; default 0
SPREADSHEET = *string*	Whether to put the design factors into a spreadsheet (design); default *

No parameters

Description

Partially replicated designs can be used when a design is too small to provide more than one unit for every treatment. So some treatments occur on only one unit in the design, others occur on two units, and there may be others (usually control treatments) that occur on several. The designs can thus provide an effective way of screening large number of treatments.

The NREPEATS option specifies the number of repeats (i.e. replicates) of each treatment in the design, in a variate with the same number of values as the number of treatments. Also, the sum of its values defines the number of units in the design.

The LEVELS option can be used to define the treatment levels, as in the FACTOR directive, and the TREATMENTS option can save a factor containing the generated values. LEVELS can be omitted if the TREATMENTS factor has already been declared with the right levels.

The NBLOCKS option can be used to specify the number of blocks in the design, and the BLOCKS option can save a factor containing the generated values. NBLOCKS can be omitted if the BLOCKS factor has already been declared with the right number of levels. Note that, if the number of units in the design is not an exact multiple of the number blocks, some blocks will contain one fewer unit than others.

The UNITS option can supply a factor to save the values generated for the unit-within-block factor (which identifies the units within each block).

The SEED option allows you to set the seed to be used to randomize the design. The default setting of zero continues the sequence of random numbers from those used to select the random starting configurations. The NSTARTS, NTRIES and TRYSEED options control the way in which AFPREP searches for the best design, as described in the Method Section.

Printed output is controlled by the PRINT option, with settings:

design	to print the design,
efficiency	to print the harmonic mean and the range of values of the treatment efficiency factors,
factors	to print the factor values, and
monitoring	to provide monitoring information during the search.

During monitoring the current best design is marked by an asterisk (*).

You can set option SPREADSHEET=design to put the design factors into a Genstat spreadsheet.

Options: PRINT, LEVELS, NREPEATS, NBLOCKS, TREATMENTS, BLOCKS, UNITS, EFFICIENCY, NSTARTS, NTRIES, SEED, TRYSEED, SPREADSHEET.
Parameters: none.

Method

AFPREP uses the AEFFICIENCY procedure to calculate the within-block efficiency factors of the treatments. If there are fewer than 201 treatments, all their efficiency factors are calculated. This is not feasible, however, when there are more treatment. So the unreplicated treatments are then ignored. AFPREP chooses the best candidate design by firstly taking the design with the largest minimum efficiency factor (i.e. it tries to avoid having a low efficiency for any treatment contrast). Then, if there are several designs with the same minimum efficiency factor, it takes the design with the largest harmonic mean efficiency factor; this aims to minimize the (ordinary, arithmetic) average standard error of difference between pairs of treatments. The efficiency factors of the best design can be saved using the EFFICIENCY option.

The candidate designs are generated in a way that avoids any treatment occurring more times than are necessary in the same block. This is done by generating block and unit values with the block factor as the fastest moving factor (i.e. block values that are repeated sequences of 1, 2...), and generating the treatment factor with all the repeated levels together. So the designs that AFPREP considers will differ according to the way in which the different levels are ordered within the treatment factor. It can run through these orderings systematically but, unless there are very few treatment levels, there will be too many orderings to examine them all. So it pursues a mixed strategy, running through a systematic sequence of orderings from several random starting arrangements. The NSTARTS option specifies the number of random starts to make, and the NTRIES option specifies the number of designs to examine in each sequence. The seed for the random numbers used to select the random starts is specified by the TRYSEED option. The default value of zero continues the existing sequence of random numbers if any have been used already in this run of Genstat. Otherwise, it initializes the seed automatically.

See also

Procedure: CDNPREP.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AFRCRESOLVABLE

Forms doubly resolvable row-column designs, with output (D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output (design, plotnumbers, factors, efficiency; default desi, effi
DESIGNPLOT = *string token*	What factors to display in the design plot (treatment, plotandtreatment); default * i.e. no plot
FIRSTPLOT = *string token*	Defines the starting location for allocating plots to the row-by-column grid (lowleft, lowright, upleft, upright); default uple
PLOTORDER = *string token*	Defines the order in which the blocks are filled (colserpentine, colbycol, rowserpentine, rowbyrow); default rowb
TIME = *scalar*	Time in seconds to spend searching for an optimal design; default 60
SEED = *scalar*	Seed for the randomization; default 0
MAXITERATIONS = *scalar*	The number of random designs to search for an optimal design; default 10000
SPREADSHEET = *string token*	What to save in a spreadsheet (data, plan); default *

Parameters

NROWS = *scalars*	Number of rows in the layout of each design
NCOLUMNS = *scalars*	Number of columns in the layout of each design
LEVELS = *scalar*, *variate* or *text*	Defines the number of levels or labels of the TREATMENT factor for each design
TREATMENTS = *factors*	Saves the treatment allocation in each design
ROWREPLICATES = *factors*	Saves the row replicates in each design
COLREPLICATES = *factors*	Saves the column replicates in each design
ROWS = *factors*	Saves the row locations of the plots in each design
COLUMNS = *factors*	Saves the column locations of the plots in each design
PLOTNUMBER = *factors*	Saves the plot numbers
TITLE = *texts*	The title for the design plot; default an automatic description of the design
OUTFILE = *texts*	Gives a file name (with extension .gsh, .gwb, or .xlsx) to save the factors in each design
EXIT = *scalars*	Saves the exit code from the design search program (0 for success, greater than 0 for failure)

Description

AFRCRESOLVABLE creates approximately optimal row-column designs. They are formed into replicates in both the row and column directions so that they are doubly resolvable, i.e. resolvable in both row and column directions. The layout of plots must be a complete rectangular array, and the treatments must be equally replicated. This requires that the number of rows multiplied by the number of columns in the array must be equal to the number of treatments multiplied by the number of replicates. The row replicates consist of units in adjacent rows, and the column replicates consist of units in adjacent columns. This design can be thought of as a generalization of a Latin square, with each treatment occurring once in each row and column replicate.

An example design with four replicates of five genotypes in a five-row by four-column array is shown below. As the number of treatments is the same as the number of rows, the column

replicates are the same as the columns, so each treatment occurs once in each column. The row replicates are shaded in different colours and consist of five plots from adjacent rows. This is the optimal design, as the treatments in the five rows form a balanced incomplete block design (within the rows, each treatment occurs three times with every other treatment).

1	**5**	**4**	**3**
2	**1**	**5**	**4**
3	**2**	**1**	**5**
4	**3**	**2**	**1**
5	**4**	**3**	**2**

The number of rows and columns in the row-column array must be specified by the NROWS and NCOLUMNS parameters respectively. The number of treatments is specified by the LEVELS parameter, either as a scalar (defining the number explicitly), or as a variate giving the levels for the treatments, or by a text defining a name for each treatment. The maximum number of rows, columns or treatments is 4000, with at most 8000 plots in the array, and the maximum number of replicates is 20. There must be at least two rows, columns and treatments, and at least three replicates, in the design. The number of columns must be less than or equal to the number of treatments. The number of rows can be greater than the number of treatments. It must then be a multiple of the number of treatments, and multiple replicates are stacked in the columns.

The PLOTNUMBER parameter saves the plot numbers, in a factor. The FIRSTPLOT option specifies where the numbering should start, as follows:

lowleft	left-hand plot at the bottom of the design, i.e. in the final row and first column;
lowright	right-hand plot at the bottom of the design, i.e. in the final row and final column;
upleft	left-hand plot at the top of the design, i.e. in the first row and first column (default);
upright	right-hand plot at the top of the design, i.e. in the first row and final column.

The PLOTORDER option defines the order in which the row or column replicates are formed and the order the plots are numbered from the first plot:

colserpentine	column-by-column in a serpentine way, e.g. top-to-bottom, and then bottom to top;
colbycol	column-by-column taking the same direction for every column;
rowserpentine	in a serpentine way, e.g. left-to-right, and then right-to-left;
rowbyrow	row-by-row taking the same direction for every row (default).

The TIME option (default 60 seconds) specifies the maximum time in seconds to spend searching for an optimal design. For large designs, TIME should be increased. For example, 1000 seconds is recommended for more than 100 treatments, and 4000 seconds for more than 200 treatments.

The SEED parameter specifies the starting seed for the randomization process; the default of zero initializes the seed automatically.

The MAXITERATIONS option (default 10000) sets the maximum number of random starting designs to use in the search. The search stops when either the TIME or the MAXITERATIONS limit is reached.

The factors for the resulting design can be saved by the TREATMENTS, ROWREPLICATES, COLREPLICATES, ROWS and COLUMNS parameters.

The EXIT parameter can save a scalar which is set to 0 if the design search has found a valid design, 1 if the design limits have been exceeded, 2 if a design is not possible, 3 if no design has been found, and 9 if not enough memory could be allocated for the design search.

Printed output is controlled by the PRINT option, with settings:

design	to print treatments in the design in a row-column layout;
plotnumbers	to print the plot numbers in a row-column layout;
factors	to print the design factors for the best design; and
efficiency	to print the combined design efficiency (using row and column replicates, rows and columns as fixed terms) for the design.

The design can be displayed in a spreadsheet using the SPREADSHEET option: the data setting creates a sheet containing the design factors (plot numbers, rows, columns, row replicates, column replicates and treatments); plan setting will display the plot numbers and treatments in a rectangular array in a sheet. The spreadsheet can be saved to a file if the OUTFILE parameter specifies a Genstat or Excel spreadsheet filename (.gsh, .gwb or .xlsx).

The design layout can be plotted in graph using the DESIGNPLOT option: the treatment setting just displays the treatments in each cell in the array; plotandtreatment displays both the plot numbers and treatments. The row and column replicates are demarked in the graph by blue and red lines respectively. The title for this graph can be set with the TITLE parameter. If this not set, a descriptive title will be created from the design parameters.

Options: PRINT, DESIGNPLOT, FIRSTPLOT, PLOTORDER, TIME, SEED, MAXITERATIONS, SPREADSHEET.

Parameters: NROWS, NCOLUMNS, LEVELS, TREATMENTS, ROWREPLICATES, COLREPLICATES, ROWS, COLUMNS, PLOTNUMBER, TITLE, OUTFILE, EXIT.

Method

The procedure calls the directive AGRCRESOLVABLE to form the design which is chosen to optimise the row-column efficiency using the MS criterion (Shah 1960). The procedure FPLOTNUMBER is used to form the plot numbers and the procedure AEFFICIENCY to calculate the design's efficiency. The ADSPREADSHEET procedure is used to save or display the design in a spreadsheet.

Reference

Shah, K.R. (1960). Optimality criteria for incomplete block designs. *Annals of Mathematical Statistics*, **22**, 235-247.

See also

Directive: AGRCRESOLVABLE.

Procedures: ADSPREADSHEET, AEFFICIENCY , DDESIGN, FPLOTNUMBER.

Genstat Reference Manual 1 Summary section on: Design of experiments.

AFUNITS

Forms a factor to index the units of the final stratum of a design (R.W. Payne & W. van den Berg).

Option

BLOCKSTRUCTURE = *formula*	Defines the block factors for the design; the default is to take those specified by the BLOCKSTRUCTURE directive

Parameter

UNITS = *factor*	Factor to be formed

Description

When analysing experimental data in Genstat, it is usually unnecessary to specify the final stratum of the design. For example ANOVA, as explained in the *Guide to the Genstat Command Language*, Part 2, Section 4.2.1, will set up an internal factor called *Units* to define (along with the other block factors) the final stratum. However, it is then impossible, for example, to put the residuals into a table classified by the block factors, or to tabulate the treatment levels according to the block structure. Thus AFUNITS takes a set of block factors (specified in either a pointer or a model formula by the BLOCKSTRUCTURE option) and sets up the necessary extra factor, which is then returned by the UNITS parameter.

Option: BLOCKSTRUCTURE.
Parameter: UNITS.

Method

The FCLASSIFICATION and FORMULA directives are used, if necessary, to form a list of factors from the block formulae and then the factor values are set up using the standard Genstat facilities for calculations and manipulation.

Action with RESTRICT

None of the block factors must be restricted.

See also

Procedure: AFLABELS.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AGALPHA

Forms alpha designs by standard generators for up to 100 treatments (M.F. Franklin & R.W. Payne).

Option

PRINT = *string token*	Controls whether or not to print a plan or the generator of of the design (design, generator); if unset in an interactive run AGALPHA will ask whether the design and generator are to be printed, in a batch run the default is not to print anything

Parameters

LEVELS = *scalars*	Number of treatments
NREPLICATES = *scalars*	Number of replicates
NBLOCKS = *scalars*	Number of blocks per replicate
SEED = *scalars*	Seed for randomization; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor
REPLICATES = *factors*	Identifier for the replicate factor
BLOCKS = *factors*	Identifier for the factor to index the blocks within replicates
UNITS = *factors*	Identifier for the factor to index the units (or plots) within each block
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGALPHA)

Description

Alpha designs are a very flexible class of resolvable incomplete block designs. A resolvable design is one in which each block contains only a selection of the treatments, but the blocks can be grouped together into subsets in which each treatment is replicated once. The groupings of blocks thus form replicates, and the block structure of the design is

```
Replicates / Blocks / Units
```

Such designs are particularly useful when there are many treatments to examine and the variability of the units is such that the block size needs to be kept small. Alpha designs were thus devised originally for the analysis of plant breeding trials (Patterson & Williams 1976), where many varieties may need to be evaluated in a single trial, and have the advantage that they can provide effective designs for any number of treatments.

The formation of an alpha design requires a generating array, as explained in the description of procedure AFALPHA, and the effectiveness of the design that is produced will be very dependent on the choice of array. Procedure AGALPHA selects an appropriate array from those presented by Patterson, Williams & Hunter (1978) and Williams (1975), and then calls AFALPHA to generate the design.

AGALPHA is easiest to use interactively. It then asks questions to determine the necessary information to select the generating array: for example, the number of treatments, the number of blocks per replicate and so on. The parameters allow you to anticipate questions, or to define all the necessary information if you want to use AGALPHA in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The number of treatments can be defined using the LEVELS parameter. Similarly, the NREPLICATES and NBLOCKS parameters define the number of replicates and the number of

blocks per replicate. If the number of blocks per replicate is greater than or equal to the number of units (or plots) per block, generators are available for either two, three or four replicates; otherwise there can only be two. The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is –1, to suppress randomization. If you do not set SEED when running interactively AGALPHA will ask for a seed, and again a negative value suppresses any randomization. The remaining parameters, TREATMENTS, REPLICATES, BLOCKS and UNITS, allow you to specify identifiers for the treatment, replicate, block-within-replicate and unit-within-block factors. If these are not specified in a batch run, AGALPHA will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGALPHA will ask you to provide identifiers, and these will remain available after AGALPHA has finished running.

AGALPHA has a PRINT option which can be set to design to print the plan of the design, and generator to print the generator of the design. By default, if you are running Genstat in batch, neither are printed. If you do not set PRINT when running interactively, AGALPHA will ask whether or not you wish to print the design or generator.

Option: PRINT.

Parameters: LEVELS, NREPLICATES, NBLOCKS, SEED, TREATMENTS, REPLICATES, BLOCKS, UNITS, STATEMENT.

Method

The QUESTION procedure is used to obtain the necessary details of the design. Procedure AFALPHA is then called to generate the design.

References

Patterson, H.D. & Williams E.R. (1976). A new class of resolvable incomplete block designs. *Biometrika*, **63**, 83-92.

Patterson, H.D., Williams E.R. & Hunter, E.A. (1978). Block designs for variety trials. *Journal of Agricultural Science, Cambridge*, **90**, 395-400.

Williams, E.R. (1975). *A new class of resolvable block designs*. Ph.D. Thesis, University of Edinburgh.

See also

Procedure: AFALPHA.

Genstat Reference Manual 1 Summary sections on: Design of experiments, REML analysis of linear mixed models.

AGBIB

Generates balanced incomplete block designs (R.W. Payne).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design and whether to print a catalogue of the designs in the subfile (design, catalogue); if unset in an interactive run AGBIB will ask whether the design is to be printed, in a batch run the default is not to print anything
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run

Parameters

LEVELS = *scalars*	Number of treatments
NBLOCKS = *scalars*	Number of blocks
NUNITS = *scalars*	Number of units per block
SEED = *scalars*	Seed for randomization; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor
BLOCKS = *factors*	Identifier for the factor to index the blocks
UNITS = *factors*	Identifier for the factor to index the units within each block
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGBIB)

Description

Incomplete block designs occur when the units in an experiment need to be divided into blocks that are not large enough to contain a unit for every treatment. In a balanced incomplete block design the contents of the blocks are arranged so that every pair of treatments occurs in an equal number of blocks. All comparisons between treatments are thus made with equal accuracy, so the design is balanced and, in particular, can be analysed by ANOVA.

AGBIB provides a range of balanced incomplete block designs, and is easiest to use interactively. It then asks questions to determine the necessary information to form the design. The options and parameters allow you to anticipate questions, or to define all the necessary information if you want to use AGBIB in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

First of all, AGBIB asks you to select the design. It lists those available, specifying the number of treatments, the number of blocks, the size of each block and the number of blocks containing each pair of treatments. Alternatively, if you set the LEVELS parameter to the required number of treatments, the NBLOCKS parameter to the number of blocks and the NUNITS parameter to the number of units per block, AGBIB will select the design (if available) automatically.

The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is –1, to suppress randomization. If you do not set SEED when running interactively AGBIB will ask for a seed, and again a negative value suppresses any randomization.

Parameters TREATMENTS, BLOCKS and UNITS, allow you to specify identifiers for the

treatment, the block and unit-within-block factors. If these are not specified in a batch run, AGBIB will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGBIB will ask you to provide identifiers, and these will remain available after AGBIB has finished running.

The PRINT option controls printed output, with setting design to print a plan of the design, and catalogue to print a list of the available designs. By default, if you are running Genstat in batch, nothing is printed. If you do not set PRINT when running interactively, AGBIB will ask whether or not you wish to print the design, after it has been generated. Similarly the ANALYSE option governs whether or not AGBIB produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGBIB assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run.

Options: PRINT, ANALYSE.
Parameters: LEVELS, NBLOCKS, NUNITS, SEED, TREATMENTS, BLOCKS, UNITS, STATEMENT.

Method

AGBIB generates designs with blocks of size two by using the standard Genstat calculation and manipulation facilities to form all the combinations of pairs of treatments. Other designs are generated from Hadamard matrices, as described by Hedayat & Wallis (1978). The QUESTION procedure is used to obtain the necessary details of the design. The matrices are then recovered from a backing-store file and the standard Genstat manipulation directives are used to generate the design.

Reference

Hedayat, A. & Wallis, W.D. (1978). Hadamard matrices and their applications. *Annals of Statistics*, **6**, 1184-1238.

See also

Procedures: AGCYCLIC, FHADAMARDMATRIX.
Genstat Reference Manual 1 Summary sections on: Design of experiments, REML analysis of linear mixed models.

AGBOXBEHNKEN

Generates Box-Behnken designs (R.W. Payne).

Options

`PRINT` = *string token*	Controls printed output (`design`); if unset in an interactive run `AGBOXBEHNKEN` will ask whether the design is to be printed, in a batch run the default is not to print anything
`NCENTRALPOINTS` = *scalar*	Defines the number of central points to include; default 4
`LEVELS` = *variate*	Defines the outer levels to be used; default `!(-1,1)`
`NCOMBINATIONS` = *scalar*	Number of factors to vary in combination at once; default 2
`SEED` = *scalar*	Seed to be used to randomize each design; a negative value implies no randomization
`STATEMENT` = *text*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from `AGBOXBEHNKEN`)

Parameter

`TREATMENTFACTOR` = *factors*	Treatment factors

Description

Box-Behnken designs are often used to study response surfaces. The design is usually formed to allow a quadratic response surface to be fitted. The factors are studied at three equally-spaced levels, below denoted by −1, 0 and 1. The construction uses a balanced incomplete block design to select successive sets of factors to be applied at all factorial combinations of −1 and +1, while other factors are held at 0. For example, with three factors `A`, `B` and `C`, the relevant balanced incomplete block design would have three blocks (`A`,`B`), (`A`,`C`) and (`B`,`C`). So the design would first have a section with `A` and `B` varying but `C` constant

```
 A    B    C
-1   -1    0
-1   +1    0
+1   -1    0
+1   +1    0
```

then a section where `B` is held constant but `A` and `C` take all combinations of −1 and +1

```
 A    B    C
-1    0   -1
-1    0   +1
+1    0   -1
+1    0   +1
```

and finally a section with `A` constant

```
 A    B    C
 0   -1   -1
 0   -1   +1
 0   +1   -1
 0   +1   +1
```

In addition, there can be some "central points", where all the factors take the central value

```
 A    B    C
 0    0    0
 0    0    0
 0    0    0
 0    0    0
```

The treatment factors are listed using the TREATMENTFACTOR parameter. If this is omitted in an interactive run, you will be asked how many factors you want and their names. The number of central points is specified by the NCENTRALPOINTS option; by default this is taken to be four. The LEVELS option can supply a variate to specify the outer treatment levels; the defaults are 1 and −1 (so the central point is at zero). The NCOMBINATIONS option defines the number of factors whose combinations of (outer) levels are to be varied at once. For the default of two, the relevant balanced incomplete block design is formed within AGBOXBEHNKEN. Other values can be supplied, but the corresponding balanced incomplete block design must be one of those obtainable from procedure AGBIB. You can find out the possibilities by putting

```
AGBIB [PRINT=catalogue]
```

The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is −1, to suppress randomization. If you do not set SEED when running interactively AGBOXBEHNKEN will ask for a seed, and again a negative value suppresses any randomization. The PRINT option can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGBOXBEHNKEN will ask whether or not you wish to print the design.

The STATEMENT option allows you to save a Genstat text structure containing a command to recreate the design. This is particularly useful if AGBOXBEHNKEN is being used interactively, and the information to define the design has been provided in response to questions from the procedure.

Options: PRINT, NCENTRALPOINTS, LEVELS, NCOMBINATIONS, SEED, STATEMENT.
Parameter: TREATMENTFACTOR.

Method

The QUESTION procedure is used to obtain the necessary details of the design and this is then generated by the standard Genstat manipulation directives and procedure AGBIB.

See also

Directive: AFRESPONSESURFACE.
Procedures: AFNONLINEAR, AGCENTRALCOMPOSITE, AGMAINEFFECT, RQUADRATIC.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Regression analysis.

AGCENTRALCOMPOSITE

Generates central composite designs (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (design); if unset in an interactive run AGCENTRALCOMPOSITE will ask whether the design is to be printed, in a batch run the default is not to print anything
NCENTRALPOINTS = *scalar*	Defines the number of central points to include; default 4
NSTARPOINTS = *scalar*	Defines the number of star points to include; default 1
LFACTORIAL = *variate*	Defines the treatment levels in the factorial part of the design; default !(-1,1)
LSTAR = *variate*	Defines the treatment levels for the star points; default is to use the levels defined by LFACTORIAL
FRACTION = *scalar*	Denominator for fractional factorial; default 1 specifies a complete design
SEED = *scalar*	Seed to be used to randomize each design; a negative value implies no randomization
STATEMENT = *text*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGCENTRALCOMPOSITE)

Parameter

TREATMENTFACTOR = *factors*	Treatment factors

Description

Central composite designs are used for estimating quadratic response surfaces, that is, the model to be fitted to the results is a quadratic function of the various factors. The design is made up of three sets of points.

a) a factorial design: usually this contains all combinations of the factors at a pair of levels (l_1,l_2), but for five or more factors it is feasible to use a fractional factorial (and still be able to estimate all the parameters of the response surface)
b) star points: this contains a pair of points for each factor where the other factors take the value (l_1+l_2)/2 and the factor has the values s_1 and s_2
c) centre points: here all the factors have the value (l_1+l_2)/2

The treatment factors are listed using the TREATMENTFACTOR parameter. If this is omitted in an interactive run, you will be asked how many factors you want and their names. The number of central points is specified by the NCENTRALPOINTS option; by default this is taken to be four. The LFACTORIAL option can supply a variate to specify the levels to be used in (a); the defaults are 1 and −1 (so the central point is at zero). Similarly, LSTAR specifies the levels for (b), which are taken, by default, to be the same as in (a). The star levels must, however, be equally spaced around the centre point. Option NSTARPOINTS defines how may replicates to have of each star point. The FRACTION option supplies the denominator of a fractional design, if required for (a); the default of one indicates that a complete factorial design is to be used.

The SEED option allows you to specify a seed to be used to randomize the design. In batch the default seed is −1, to suppress randomization. If you do not set SEED when running interactively AGCENTRALCOMPOSITE will ask for a seed, and again a negative value suppresses any randomization. The PRINT option can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGCENTRALCOMPOSITE will ask whether or not you wish to print the

design.

The STATEMENT option allows you to save a Genstat text structure containing a command to recreate the design. This is particularly useful if AGCENTRALCOMPOSITE is being used interactively, and the information to define the design has been provided in response to questions from the procedure.

Options: PRINT, NCENTRALPOINTS, NSTARPOINTS, LFACTORIAL, LSTAR, FRACTION, SEED, STATEMENT.
Parameter: TREATMENTFACTOR.

Method

The QUESTION procedure is used to obtain the necessary details of the design and this is then generated by the Genstat manipulation directives and procedure AGFRACTION.

See also

Directive: AFRESPONSESURFACE.
Procedures: AFNONLINEAR, AGBOXBEHNKEN, AGMAINEFFECT, RQUADRATIC.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Regression analysis.

AGCROSSOVERLATIN

Generates Latin squares balanced for carry-over effects (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (design); if unset in an interactive run ACROSSOVERGLATIN will ask whether the design is to be printed, in a batch run the default is not to print anything
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (yes, no); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run

Parameters

LEVELS = *scalars* or *variates*	Number of treatments (scalar) or levels for the treatments
SEED = *scalars*	Seed to be used to randomize the design; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for a factor to represent the direct effects of the treatments
SUBJECTS = *factors*	Identifier for a factor to represent the subjects
PERIODS = *factors*	Identifier for a factor to represent the periods
CARRYOVERFACTOR = *factors*	Identifier for a factor to represent the carry-over (or "residual") effect of the treatments in the period immediately after the period in which they were applied
NOCARRYOVER = *factors*	Identifier for a factor to represent the comparison between none and any carry-over effect of the treatments
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGCROSSOVERLATIN)

Description

Cross-over trials are designed to study the effects of various treatments on a set of plots (in a field experiment) or subjects (in a medical trial). The special feature of these experiments is that the same plots or subjects are treated during several successive time periods, and there is interest both in the direct effect of a treatment during the period in which it is applied and its carry-over (or "residual") effect during later periods. AGCROSSOVERLATIN can generate designs for a single treatment factor for the most usual situation, where the carry-over effect is assumed to last over only one subsequent period. The design balances the direct and carry-over effects by ensuring that each treatment follows each other treatment an equal number of times. For an even number of treatments t the design consists of a single $t \times t$ Latin square, while for an odd number t it is formed from a pair of Latin squares.

The design can be analysed by ANOVA by setting

```
BLOCKSTRUCTURE Subjects * Periods
TREATMENTSTRUCTURE Nocarryover / Carryoverfactor + Treatments
```

The factor Carryoverfactor represents the carry-over effects of the treatments, and factor Nocarryover assesses whether there were any carry-over effects at all (essentially this is a comparison between the periods 2 onwards where there were carry-over effects from earlier times, and period 1 where there was none). So the treatment formula expands to specify terms

Nocarryover	none versus any carry-over effect

`Nocarryover.Carryoverfactor`	differences in carry-over effect amongst the treatments (assuming that there was an earlier treatment)
`Treatments`	direct effects of treatments, eliminating any carry-over effect

The direct and carry-over effects are not orthogonal, so it may be of interest also to specify

```
TREATMENTSTRUCTURE Treatments + Nocarryover / Carryoverfactor
```

in order to estimate the carry-over effects eliminating the direct effects.

`AGCROSSOVERLATIN` is easiest to use interactively. All the information required to generate the design is then obtained by (clearly explained) questions. You need set the parameters only if you wish to anticipate some of the questions, or if you wish to use `AGCROSSOVERLATIN` in batch. If, however, you wish to recreate the same design later, the `STATEMENT` parameter allows you to save a Genstat text structure containing a command specifying the same information.

The number of treatments can be defined using the `LEVELS` parameter. The `SEED` parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is –1, to suppress randomization. If you do not set `SEED` when running interactively `AGCROSSOVERLATIN` will ask for a seed, and again a negative value suppresses any randomization.

Parameters `TREATMENTS`, `CARRYOVERFACTOR` and `NOCARRYOVER` allow you to specify identifiers for factors to represent the direct effects of the treatments, the carry-over effects in the subsequent period, and the comparison between none and any carry-over effect. Similar the parameters `SUBJECTS` and `PERIODS` can specify identifiers for factors to represent the subjects (or plots) and time periods respectively. If these parameters are not specified in a batch run, `AGCROSSOVERLATIN` will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, `AGCROSSOVERLATIN` will ask you to provide identifiers, and these will remain available after `AGCROSSOVERLATIN` has finished running.

The `PRINT` option can be set to `design` to print the design. By default, if you are running Genstat in batch, the nothing is printed. If you do not set `PRINT` when running interactively, `AGCROSSOVERLATIN` will ask what you want to print. Similarly the `ANALYSE` option governs whether or not `AGCROSSOVERLATIN` produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again `AGCROSSOVERLATIN` assumes that this is not required if `ANALYSE` is unset in a batch run, and asks whether it is required if `ANALYSE` is unset in an interactive run.

Options: `PRINT`, `ANALYSE`.

Parameters: `LEVELS`, `SEED`, `TREATMENTS`, `SUBJECTS`, `PERIODS`, `CARRYOVERFACTOR`, `NOCARRYOVER`, `STATEMENT`.

Method

`AGCROSSOVERLATIN` generates the design using the standard Genstat calculation and manipulation commands.

See also

Procedures: `AFCARRYOVER`, `AGLATIN`, `AGSEMILATIN`, `AGQLATIN`, `GALOIS`, `XOEFFICIENCY`, `XOPOWER`.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGCYCLIC

Generates cyclic designs from standard generators (M.F. Franklin & R.W. Payne).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGCYCLIC will ask whether the design is to be printed, in a batch run the default is not to print the design
METHOD = *string token*	Type of design – ordinary cyclic, cyclic change-over or cyclic superimposed (cyclic, changeover, superimposed); if unset in an interactive run AGCYCLIC will ask about the type of design, in a batch the default is assumed to be cyclic

Parameters

LEVELS = *scalars*	Number of treatments
NBLOCKS = *scalars*	Number of blocks
NUNITS = *scalars*	Number of units per block, or number of periods in a cyclic change-over design
SEED = *scalars*	Seed for randomization; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor
SUPERIMPOSED = *factors*	Identifier for the second treatment factor in a cyclic superimposed design
BLOCKS = *factors*	Identifier for the factor to index the blocks
UNITS = *factors*	Identifier for the factor to index the units within each block, or the periods of a cyclic change-over design
INITIALBLOCKS = *variates* or *pointers*	To save one (variate) or more (pointer to variates) initial blocks
STATEMENT = *texts*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGCYCLIC)

Description

Cyclic designs provide an effective way of assessing treatments using a block design where the blocks are each too small to hold all the treatments. In its simplest form, the cyclic method of generation starts with an initial block containing some subset of the treatments. This subset is represented by integers in the range 0...$m-1$ where m is the number of treatment levels. The second and subsequent blocks are then generated by successive addition modulo m of one to the numbers in the subset. Some designs have more than one initial block, and the increment need not be one. Further details of the method are given in the description of procedure AFCYCLIC.

The efficiency of the design depends very much on the choice of initial blocks. Procedure AGCYCLIC selects appropriate initial blocks from a repertoire obtained mainly from the program DSIGNX (Franklin & Mann 1986), and including designs from Davis & Hall (1969), Hall & Williams (1973) and John, Wolock & David (1972). It then calls AFCYCLIC to generate the design.

AGCYCLIC is easiest to use interactively. It then asks questions to determine the necessary information to form the design. In particular, it will tell you which block sizes are available for your chosen number of treatments. The options and parameters allow you to anticipate questions, or to define all the necessary information if you want to use AGCYCLIC in batch. If, however, you

wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The first question, which can be anticipated by setting the METHOD option, determines the type of cyclic design. In addition to the standard cyclic designs, AGCYCLIC can also generate the cyclic change-over designs of Davis & Hall (1969) and the cyclic superimposed designs of Hall & Williams (1973). The change-over designs are used for trials in which subjects are given different treatments in different time periods; these thus have a crossed block structure subjects*periods. The extension in the cyclic superimposed design is that there are two treatment factors (each with the same number of levels); the design is intended to estimate their main effects but not their interaction.

The PRINT option controls whether AGCYCLIC prints a plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGCYCLIC will ask whether or not you wish to print the design, after it has been generated.

The number of treatments can be defined using the LEVELS parameter. Similarly, the NBLOCKS and NUNITS parameters define the number of blocks and the number of units per block (or the number of periods in a cyclic change-over design). The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is −1, to suppress randomization. If you do not set SEED when running interactively AGCYCLIC will ask for a seed, and again a negative value suppresses any randomization.

Parameters TREATMENTS, SUPERIMPOSED, BLOCKS and UNITS, allow you to specify identifiers for the treatment, the superimposed treatment (for a cyclic superimposed design), the block and unit-within-block factors. If these are not specified in a batch run, AGCYCLIC will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGCYCLIC will ask you to provide identifiers, and these will remain available after AGCYCLIC has finished running.

Finally, the INITIALBLOCKS parameter allows you to save the initial blocks, in a variate if there is only one, or in a pointer (to a list of variates) if there are several.

Options: PRINT METHOD.

Parameters: LEVELS, NBLOCKS, NUNITS, SEED, TREATMENTS, SUPERIMPOSED, BLOCKS, UNITS, INITIALBLOCKS, STATEMENT.

Method

The QUESTION procedure is used to obtain the necessary details of the design. The initial blocks are then recovered from a backing-store file and procedure AFCYCLIC is called to generate the design.

References

Davis, A.W. & Hall, W.B. (1969). Cyclic change-over designs. *Biometrika*, **56**, 283-293.

Franklin, M.F. & Mann, A.D. (1986). *DSIGNX a Program for the Construction of Randomized Experimental Plans*. Scottish Agricultural Statistics Service, Edinburgh (revised edition).

Hall, W.B. & Williams, E.R. (1973). Cyclic superimposed designs. *Biometrika*, **60**, 47-53.

John, J.A., Wolock, F.W. & David, H.A. (1972). *Cyclic Designs*. National Bureau of Standards, Applied Mathematics Series 62.

See also

Procedures: AFCYCLIC, AGBIB.

Genstat Reference Manual 1 Summary sections on: Design of experiments, REML analysis of linear mixed models.

AGDESIGN

Generates generally balanced designs (R.W. Payne).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design and whether to print a catalogue of the designs in the subfile (design, catalogue); if unset in an interactive run AGDESIGN will ask whether the design is to be printed, in a batch run the default is not to print anything
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run
FILENAME = *text*	Name of the backing store file containing the design information; default uses the standard design file
SUBFILE = *identifier*	Subfile of the backing store file to be used

Parameters

DESIGN = *variates*	Contains codes to indicate the choice of design
TREATMENTFACTORS = *pointers*	Specifies identifiers for the treatment factors
BLOCKFACTORS = *pointers*	Specifies identifiers for the block factors
PSEUDOFACTORS = *pointers*	Specifies identifiers for any pseudo-factors
REPLICATEFACTOR = *factors*	Specifies the identifier of the factor to represent the replicates (if any) in each design
UNITLABELS = *variates*	Specifies the identifier of a variate to store a unique numerical label for each plot in the design
SEED = *scalars*	Seed to be used to randomize each design; a negative value implies no randomization
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGDESIGN)

Description

AGDESIGN generates the factors and, if necessary, pseudo-factors required to define a generally balanced design. It also sets the block and treatment formulae (using the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives) to allow the design to be analysed by ANOVA. It can be accessed most conveniently through interactive Genstat *design system*, using the procedure DESIGN.

AGDESIGN relies upon a backing-store subfile that contains a repertoire of available designs, together with the information required to form them. FILENAME has a default file containing four subfiles.

FACTORIAL – factorial designs (with blocking): these have several treatment factors and a single blocking factor (giving strata for blocks and plots within blocks); the blocks are too small to contain a complete replicate of the treatment combinations and so various interaction are confounded with blocks.

LATTICE – lattice designs: designs for a single treatment factor with number of levels that is the square of some integer *k*; the design has replicates, each containing *k* blocks of *k* plots, and different treatment contrasts can be confounded with blocks in each replicate.

LATTSQ – lattice squares: these are similar to lattices except that the blocking structure with the replicates has rows crossed with columns; again different treatment contrasts can be

confounded with the rows and columns in each replicate.

LATIN – Latin squares: designs are available for 3 to 14 treatments; several different orthogonal squares are available for most of these so, for example, Graeco Latin squares can be formed by using a different square for each of the two treatment factors.

If the default FILENAME is being used, the usual abbreviation rules are used to match SUBFILE with the names of the subfiles in the default file.

The backing-store file can be created by a procedure called FDESIGNFILE. This requires a data file, details of whose format can be obtained by setting option PRINT=filestructure when running FDESIGNFILE. You can thus provide additional files of designs which can be accessed by setting the FILENAME and SUBFILE options as appropriate.

AGDESIGN has two other options. The PRINT option can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGDESIGN will ask whether or not you wish to print the design. The other setting catalogue lists the designs in the subfile. Similarly the ANALYSE option governs whether or not AGDESIGN produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGDESIGN assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run.

The information required to select the design and give identifiers to its factors can be defined using the parameters of AGDESIGN. In an interactive run, AGDESIGN will ask questions to obtain any necessary information that is not supplied in this way; when running in batch, if any of the required information has not been specified, AGDESIGN will terminate with a warning message.

It is thus easiest to use AGDESIGN interactively. Then only the SUBFILE option need be set (assuming that you are happy to use the standard default design file), and the other information will be obtained by (clearly explained) questions. You need set the parameters only if you wish to anticipate some of the questions, or if you wish to use AGDESIGN in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The DESIGN parameter can supply a variate whose first value selects the "type" of design: for example, in the LATTICE subfile, this would select between a 3×3 lattice, a 4×4 lattice, and so on. Some of these designs are available in several different "versions": for example, in lattice designs there are several ways of defining which treatment contrasts are to be confounded with blocks. If there is more than one version, the second and subsequent values of the DESIGN variate indicate which version, or versions, are required. These need not be distinct so, for example, you can replicate a basic design several times. If the variate has a single value, AGDESIGN will select the first version.

The TREATMENTFACTORS parameter can specify a pointer to supply identifiers for the treatment factors in the design. For example, if there are two factors you could define their identifiers to be A and B by forming the pointer Tf (say) with the statement

```
POINTER [VALUES=A,B] Tf
```

and then setting TREATMENTFACTORS=Tf. Alternatively, and more succinctly, you could put TREATMENTFACTORS=!p(A,B), where !p(A,B) is an unnamed pointer containing the required two identifiers. Similarly the BLOCKFACTORS parameter can specify a pointer to define the identifiers for the block factors in the basic design. If you have requested several versions, or several replicates, of the basic design AGDESIGN will also need a factor to represent the replicates. The identifier of this factor can be supplied using the REPLICATEFACTOR parameter. Partially balanced designs, such as lattices, will require pseudo-factors in the treatment formula to enable the design to be analysed by ANOVA. Identifiers can be supplied for these using the PSEUDOFACTORS parameter.

The UNITLABELS parameter can specify a variate to store a unique number to label each of the plots in the design. In the first replicate (or version) in the generated design, the variate

contains the numbers one up to the number of plots per replicate. The second replicate (if any) contains these numbers plus the smallest power of ten greater than the number of plots per replicate, the third replicate contains the numbers plus twice this power of ten, and so on.

The SEED parameter allows you to specify a seed to randomize the design. In a batch run, this has a default of –1, to suppress randomization. If SEED is unset in an interactive run, you will be asked to provide a seed (and again a negative value will leave the design unrandomized).

Options: PRINT, ANALYSE, FILENAME, SUBFILE.
Parameters: DESIGN, TREATMENTFACTORS, BLOCKFACTORS, PSEUDOFACTORS, REPLICATEFACTOR, UNITLABELS, SEED, STATEMENT.

Method
The QUESTION procedure is used to obtain the details of the required design. The design is then generated using GENERATE and the other standard Genstat directives for calculation and manipulation.

See also
Procedures: AGFACTORIAL, AGFRACTION, AGHIERARCHICAL, FDESIGNFILE.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGFACTORIAL

Generates minimum aberration block or fractional factorial designs (P.J. Laycock, P.J. Rowley & R.W. Payne).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGFACTORIAL will ask whether the design is to be printed, in a batch run the default is not to print the design
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (yes, no); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run
FACTORIAL = *scalar*	Limit on number of factors in treatments terms in the analysis of variance; default 3

Parameters

LEVELS = *scalars*, *variates* or *texts*	Levels for the treatment factors in each design
NTREATMENTFACTORS = *scalars*	Number of treatment factors
NUNITS = *scalars*	Number of units per block
NFRACTIONBLOCK = *scalars*	Defines the number of the block to use to define a fractional factorial, or can be set to zero to take a block at random; if unset in an interactive run AGFACTORIAL will ask whether to form a fractional factorial design, in a batch run the default is to form the full (block) design
NSUBUNITS = *scalars*	Number of units in each sub-block
SEED = *scalars*	Seed to be used to randomize each design; a negative value implies no randomization
TREATMENTFACTORS = *pointers*	Specifies identifiers for the treatment factors
BLOCKS = *factors*	Identifier for the block factor
SUBBLOCKS = *factors*	Identifier for the sub-block factor
PSEUDOFACTORS = *pointers*	Specifies identifiers for pseudo-factors
UNITLABELS = *variates*	Specifies the identifier of a variate to store a unique numerical label for each unit in the design
NDESIGN = *scalars*	Saves or defines the design number
NSUBDESIGN = *scalars*	Saves or defines the sub-design number
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGFACTORIAL)

Description

AGFACTORIAL generates efficient block or fractional factorial designs using the minimum aberration algorithm of Laycock & Rowley (1995), implemented in the AFMINABERRATION directive. It also sets the block and treatment formulae (using the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives), and generates any pseudo-factors needed to analyse the design using the ANOVA directive.

To explain minimum aberration for a block design, we start by defining the resolution of a design as the largest integer r such that no interaction term with r factors is confounded with blocks. The aberration of the design is the number of interaction terms with r+1 factors that are

confounded. A minimum aberration design is defined as a design with the smallest aberration out of the designs with the highest available resolution. So, essentially this minimizes the number of interactions with the minimum number of factors that are confounded. The definition for a fractional factorial design is essentially the same. The fractional factorial is constructed by taking only one block from the block design, and the terms that were confounded with blocks in the block design become aliased in the fractional factorial.

AGFACTORIAL can be used either in batch or interactively. In an interactive run, it obtains the information necessary to select and define the design by asking questions. You need set the parameters only if you wish to anticipate some of the questions, or if you wish to use AGFACTORIAL in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The LEVELS parameter defines the number of levels of the treatment factors, either as a scalar or by providing a text or variate with the required number of levels, to use for the LEVELS option of the FACTOR directive. This must be a prime number (e.g. 2, 3, 5, 7, 11) or a power of a prime number (e.g. 4, 8, 9). The number of treatment factors is specified by the NTREATMENTFACTOR parameter. The number of the units in each block (or, equivalently, the number of units in a fractional factorial) is specified by the NUNITS parameter; this must be a power of the number of levels. The NFRACTIONBLOCK parameter allows you to form a fractional factorial, either by setting it to the number of the block to take, or by setting it to zero to take a block at random; if you set NFRACTIONBLOCK to a scalar containing a missing value, AGFACTORIAL forms a block design. You can define blocks for a fractional factorial (or, equivalently, sub-blocks for a block design) by defining their size using NSUBUNITS parameter; this too must be a power of the number of levels.

The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is –1, to suppress randomization. If you do not set SEED when running interactively AGFACTORIAL will ask for a seed, and again a negative value suppresses any randomization.

The TREATMENTFACTORS parameter can specify a pointer to supply identifiers for the treatment factors in the design. For example, if there are two factors you could define their identifiers to be A and B by forming the pointer Tf (say) with the statement

```
POINTER [VALUES=A,B] Tf
```

and then setting TREATMENTFACTORS=Tf. Alternatively, and more succinctly, you could put TREATMENTFACTORS=!p(A,B), where !p(A,B) is an unnamed pointer containing the required two identifiers. The BLOCKS and SUBBLOCKS parameters allow you to specify identifiers for the block and sub-block factors. Designs where the treatment factors have more than two levels may require pseudo-factors to be defined in order for them to be analysed by ANOVA. The PSEUDOFACTORS parameter can specify a pointer to supply their identifiers. If the treatment, block or sub-block factors and any necessary pseudo-factors are not specified in a batch run, AGFACTORIAL will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGFACTORIAL will ask you to provide identifiers, and these will remain available after AGFACTORIAL has finished running.

The UNITLABELS parameter can specify a variate to store a unique number to label each of the units in the design. In the first block, the variate contains the numbers one up to the number of units per block. The second block contains these numbers plus the smallest power of ten greater than the number of units per block, the third block contains the numbers plus twice this power of ten, and so on.

The PRINT option can be set to design to print the plan of the design, and summary to print a summary of the design properties. By default, if you are running Genstat in batch, these are not printed. If you do not set PRINT when running interactively, AGFACTORIAL will ask whether or not you wish to print them. Similarly the ANALYSE option governs whether or not AGFACTORIAL

produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGFACTORIAL assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run. The FACTORIAL option sets a limit on the number of factors in the treatment terms in the analysis of variance; by default, this is three.

The NDESIGN parameter can save a unique *design number* for the design, and the NSUBDESIGN can save a unique number for the sub-design of the design (as defined by Laycock & Rowley 1995). You can input these with NDESIGN and NSUBDESIGN later, along with the same settings for LEVELS, NTREATMENTFACTORS, NUNITS and NSUBUNITS, to generate the design factors again without repeating the design search.

Options: PRINT, ANALYSE, FACTORIAL.

Parameters: LEVELS, NTREATMENTFACTORS, NUNITS, NFRACTIONBLOCK, NSUBUNITS, SEED, TREATMENTFACTORS, BLOCKS, SUBBLOCKS, PSEUDOFACTORS, UNITLABELS, NDESIGN, NSUBDESIGN, STATEMENT.

Method

The QUESTION procedure is used to obtain the details of the required design. The design is selected using the Laycock & Rowley (1995) search algorithm for minimum aberration designs, as implemented in the AFMINABERRATION directive. The block and treatment factors are then generated using the standard Genstat directives for calculation and manipulation.

References

Laycock, P.J. & Rowley, P.J. (1995). A method for generating and labelling all regular fractions or blocks for q^{n-m} designs. *Journal of the Royal Statistical Society, Series B*, **57**, 191-204.

See also

Directive: AFMINABERRATION.

Procedures: AGDESIGN, AGFRACTION, AGHIERARCHICAL.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGFRACTION

Generates fractional factorial designs (M.F. Franklin & R.W. Payne).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGFRACTION will ask whether the design is to be printed, in a batch run the default is not to print the design
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run
FACTORIAL = *scalar*	Limit on number of factors in treatments terms in the analysis of variance; default 2
FILENAME = *text*	Name of the backing store file containing the design information; default uses the standard fractional design file

Parameters

LEVELS = *scalars*	Number of levels of the treatment factors in each design
FRACTION = *scalars*	Denominator of required fraction
NTREATMENTFACTORS = *scalars*	Number of treatment factors
NUNITS = *scalars*	Number of units per block
SEED = *scalars*	Seed to be used to randomize each design; a negative value implies no randomization
TREATMENTFACTORS = *pointers*	Specifies identifiers for the treatment factors
BLOCKS = *factors*	Identifier for the block factor
UNITS = *factors*	Identifier for the factor to index the units (or plots) within each block
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGFRACTION)

Description

AGFRACTION generates fractional factorial designs from stored keys & other information. It also sets the block and treatment formulae (using the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives) to allow the design to be analysed by ANOVA.

The procedure relies upon a backing-store file that contains a repertoire of available designs, together with the information required to form them. There is a standard file, used by default, but the FILENAME option allows you to specify another if you wish to form your own alternative file.

AGFRACTION has two other options. The PRINT option can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGFRACTION will ask whether or not you wish to print the design. Similarly the ANALYSE option governs whether or not AGFRACTION produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGFRACTION assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run. The FACTORIAL option sets a limit on the number of factors in the treatment terms in the analysis of variance; by default, this is two.

The information required to select the design and give identifiers to its factors can be defined using the parameters of AGFRACTION. In an interactive run, AGFRACTION will ask questions to obtain any necessary information that is not supplied in this way; when running in batch, if any of the required information has not been specified, AGFRACTION will terminate with a warning message.

It is thus easiest to use AGFRACTION interactively. Then all the information necessary to select and define the required design will be obtained by (clearly explained) questions. You need set the parameters only if you wish to anticipate some of the questions, or if you wish to use AGFRACTION in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The number of levels of the treatment factors can be defined using the LEVELS parameter. The FRACTION parameter defines the denominator of the required fraction, and the NTREATMENTFACTOR parameter specifies how many treatment factors the design is to contain. Thus, for example,

```
AGFRACTION [PRINT=design] LEVELS=2; FRACTION=4; NTREATMENTF=6
```

would print the plan of a quarter replicate of a 2^6 design.

For some of the designs it is possible also to allow a blocking factor (and you will be given details of what is feasible if you are running AGFRACTION interactively). The NUNITS parameter can then be used to define the number of units per block.

The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is –1, to suppress randomization. If you do not set SEED when running interactively AGFRACTION will ask for a seed, and again a negative value suppresses any randomization.

The TREATMENTFACTORS parameter can specify a pointer to supply identifiers for the treatment factors in the design. For example, if there are two factors you could define their identifiers to be A and B by forming the pointer Tf (say) with the statement

```
POINTER [VALUES=A,B] Tf
```

and then setting TREATMENTFACTORS=Tf. Alternatively, and more succinctly, you could put TREATMENTFACTORS=!p(A,B), where !p(A,B) is an unnamed pointer containing the required two identifiers. The remaining parameters, BLOCKS and UNITS, allow you to specify identifiers for the block and unit-within-block factors. If the treatment, block or unit factors are not specified in a batch run, AGFRACTION will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGFRACTION will ask you to provide identifiers, and these will remain available after AGFRACTION has finished running.

Options: PRINT, ANALYSE, FACTORIAL, FILENAME.

Parameters: LEVELS, FRACTION, NTREATMENTFACTORS, NUNITS, SEED, TREATMENTFACTORS, BLOCKS, UNITS, STATEMENT.

Method

The QUESTION procedure is used to obtain the details of the required design. The design is then generated using GENERATE and the other standard Genstat directives for calculation and manipulation.

See also

Procedures: AGDESIGN, AGFACTORIAL, AGHIERARCHICAL.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGHIERARCHICAL

Generates orthogonal hierarchical designs (R.W. Payne).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGHIERARCHICAL will ask whether the design is to be printed, in a batch run the default is not to print the design
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run
SEED = *scalars*	Seed to be used to randomize each design; a negative value implies no randomization
STATEMENT = *text*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGHIERARCHICAL)
EXCLUDELEVELS = *scalars*	Levels of the first block factor to exclude during randomization

Parameters

BLOCKFACTORS = *factors*	Specifies the identifier for the block factor used to index the units of each stratum (or level of the hierarchy)
TREATMENTFACTORS = *factors* or *pointers*	Specifies the identifier of the treatment factor or factors applied to the units of each stratum
LEVELS = *scalars* or *pointers*	Number of levels for the treatment factors in each stratum; if required, a pointer can contain an extra scalar to specify replication

Description

AGHIERARCHICAL forms orthogonal hierarchical designs: for example randomized blocks, split-plots, split-split-plots, and so on. The units of each stratum (or level of the hierarchy) are identified by a block factor: for example Replicates, Blocks, Plots, Subplots, Subjects &c.

AGHIERARCHICAL can be used either interactively or in batch. Interactively, there is no need to set any options or parameters – the procedure will ask questions to ascertain the necessary details of the design. If, however, you wish to recreate the same design later, the STATEMENT option allows you to save a Genstat text structure containing a command specifying the same information.

The first question is to find out what type of design you want to generate: either a completely randomized design, a randomized block design. a split-plot design, a split-split-plot design or a general hierarchical design. For a general hierarchical design, you will then be asked how many block factors (and thus strata) there are in the design, but this is predefined for the other designs. For example, a completely randomized design has a single blocking factor (e.g. Units). For a randomized block design there would be two, for example Blocks and Plots, defining strata for blocks and for plots within blocks. In a split-plot design there are three (for example Blocks, Wholeplots and Subplots) giving strata for blocks, whole plots within blocks and subplots within whole plots, while in a split-split-plots design there are four.

The questions then involve each stratum in turn, and asking first for the name of the block

factor to be used to identify the units of the stratum. Next it asks how many treatment factors are applied to the units of that stratum. In a randomized block design, there are no treatment factors applied to the blocks and one, or more, applied to the plots, whereas in a split-plot design treatments are applied to both the whole plots and the subplots. It then asks for the names of the treatment factors, and how many levels they are to have. Alternatively, if there are no treatments applied to the stratum, AGHIERARCHICAL asks how many levels the corresponding block factor should have – so, for example, it would how many blocks there should be in a randomized block or a split-plot design.

The example below shows the questions and answers (displayed in bold font) to generate a randomized complete block design. The design has three blocks and six units (plots) within each block. There are two treatment factors: Type with two levels, and Amount with three levels. (Note: in Genstat *for Windows*, the questions would be the same but they appear in pop-up menus.)

```
> What type of design do you want?
c         completely randomized design
r         randomized block design
s         split-plot design
ss        split-split-plot design
o         other
Code (c,r,s,ss,o; Default:  r) > r

What would you like to call the block factor?
Identifier (Default:Blocks) >

How many replicates are there of Blocks?
Number > 3

What would you like to call the unit-within-block factor?
Identifier (Default:Units) >

How many treatment factors are there?
Number > 2

What would you like to call treatment factor 1?
Identifier > Type

How many levels does treatment factor Type have?
Number > 2

What would you like to call treatment factor 2?
Identifier > Amount

How many levels does treatment factor Amount have?
Number > 3

Seed for randomization (-1 for none)?
Number (Default: -1) >

Do you want to print the design?
n         no
y         yes
Code (n,y; Default n) > n

Do you want to check the design by ANOVA?
n         no
y         yes
Code (n,y; Default n) > n
```

The parameters of AGHIERARCHICAL provide an alternative way of providing the details of the design. BLOCKFACTORS lists the block factors for the strata, TREATMENTFACTORS defines

factors for the treatments applied to the units of the strata and LEVELS defines the levels of treatments and replication of block factors. For example

```
AGHIERARCHICAL [PRINT=design; ANALYSE=yes] \
               Blocks,Plots; *,A; 3,5
```

defines a randomized block design with three blocks, and a single treatment factor A (applied to the plots) with five levels. If there are several factors in a stratum, the identifiers should be placed into a pointer. For example,

```
AGHIERARCHICAL Blocks,Plots; *,!p(A,B); 3,2
```

for a randomized block design with two treatment factors, A and B, both with two levels. Similarly, if the factors in a stratum have different numbers of levels, the LEVELS parameter may contain pointers.

```
AGHIERARCHICAL Blocks,Plots; *,!p(A,B); 3,!p(2,3)
```

defines A to have two levels and B to have three. The pointer can contain an extra element to indicate that there is to be replication (as well as treatments) in a stratum.

```
AGHIERARCHICAL Blocks,Plots; *,!p(A,B); 3,!p(2,3,4)
```

indicates that there are to be four replicates of the A and B combinations on the plots of each block.

In an interactive run, AGHIERARCHICAL will ask about the treatment factors and the levels if these are not set. In a batch run all three parameters must be set.

The SEED option allows you to specify a seed to randomize the design. In a batch run, this has a default of −1, to suppress randomization. If SEED is unset in an interactive run, you will be asked to provide a seed (and again a negative value will leave the design unrandomized). You can use the EXCLUDELEVELS parameter to specify levels of the first block factor that you do not wish to randomize. (This can be useful in "demonstration experiments", when the treatments may need to be kept in a systematic order in some parts of the trial, but it is not a good idea in more normal situations.)

AGHIERARCHICAL has two other options. The PRINT option can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGHIERARCHICAL will ask whether or not you wish to print the design. Similarly the ANALYSE option governs whether or not AGHIERARCHICAL produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGHIERARCHICAL assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run.

Options: PRINT, ANALYSE, SEED, STATEMENT, EXCLUDELEVELS.
Parameters: BLOCKFACTORS, TREATMENTFACTORS, LEVELS.

Method

The QUESTION procedure is used to obtain the details of the required design. The design is then generated using GENERATE and the other standard Genstat directives for calculation and manipulation.

See also

Procedures: AGDESIGN, AGFACTORIAL, AGFRACTION.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGINDUSTRIAL

Helps to select and generate effective designs for use in industrial experiments (R.W. Payne).

Option

STATEMENT = *text*	Saves a command to recreate the design

No parameters

Description

AGINDUSTRIAL is a procedure which can be used interactively to form designs that are popular in industrial experiments. The process involves answering questions, posed by Genstat, first to select the particular type of design, then to give details such as names of factors, numbers of treatments, and so on. A range of subsidiary procedures may be called, depending on the type of design selected. If you wish to avoid some of the question-and-answer process, the subsidiary procedures can also be called directly. They all have options and parameters which provide an alternative way of supplying the information otherwise obtained by the various questions and, provided you supply all the required information, they can also be used in batch. The STATEMENT option of AGINDUSTRIAL allows you to save a Genstat text structure containing a command to use the relevant subsidiary procedure, and setting all the options and parameters required to recreate the design.

There are 6 types of design.

Factorial designs from a repertoire (with blocking) – these have several treatment factors and a single blocking factor (giving strata for blocks and plots within blocks). The blocks are too small to contain a complete replicate of the treatment combinations and so various interaction are confounded with blocks. (See procedure AGDESIGN.)

Fractional factorial designs from a repertoire (with blocking) – again there are several treatment factors but the design does not contain every treatment combination and so some interactions are aliased; there can also be a blocking factor and some interactions will then be confounded with blocks. (See procedure AGFRACTION.)

Balanced-incomplete-block designs – designs where the experimental units are grouped into blocks such that every pair of treatments occurs in an equal number of blocks. All comparisons between treatments are thus made with equal accuracy, so the design is balanced and, in particular, can be analysed by ANOVA. Further details are given in the description of procedure AGBIB.

Central composite designs – used to study multi-dimensional response surfaces; see procedure AGCENTRALCOMPOSITE.

Box-Behnken designs – used to study multi-dimensional response surfaces; see procedure AGBOXBEHNKEN.

Plackett Burman (main effect) designs – for estimating main effects of factors with two levels, using a minimum number of experimental units (Plackett & Burman 1946). Further details are given in the description of procedure AGMAINEFFECT.

You will be asked to provide a seed to be used to randomize the design and then given the opportunity to print a plan. If the design can be analysed by ANOVA, the procedures will define appropriate block and treatment formulae and then ask if you want to see the skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Whether or not you choose to print any of this information, at the end of the whole process all the block and treatment factors necessary to define the design will be available – and they will have the identifiers that you have supplied in response to the various questions asked by the procedures.

Option: STATEMENT. Parameters: none.

Method

The QUESTION procedure is used to find out what design is required. AGINDUSTRIAL then calls either AGDESIGN (for a factorial design), AGFRACTION (for a fractional factorial design), AGBIB (for a balanced-incomplete-block design), AGCENTRALCOMPOSITE (for a central composite design), AGBOXBEHNKEN (for a Box-Behnken design) or AGMAINEFFECT (for a Plackett Burman main effect design). The designs are generated using GENERATE and the other standard Genstat directives for calculation and manipulation. Some of the information needed to specify the designs is stored in backing-store files, and much of this was adapted from the standard designs of the program DSIGNX (Franklin & Mann 1986).

References

Franklin, M.F. & Mann, A.D. (1986). *DSIGNX a Program for the Construction of Randomized Experimental Plans*. Scottish Agricultural Statistics Service, Edinburgh (revised edition).

Plackett, R.L. & Burman, J.P. (1946). The design of optimum factorial experiments. *Biometrika*, **33**, 305-325 & 328-332.

See also

Genstat Reference Manual 1 Summary section on: Design of experiments.

AGLATIN

Generates mutually orthogonal Latin squares (I. Wakeling & R.W. Payne).

Options

PRINT = *string token*	Controls printed output (design, squares, list); if unset in an interactive run AGLATIN will ask whether the design is to be printed, in a batch run the default is not to print anything
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run

Parameters

NROWS = *scalars*	Specifies the number of rows (and columns) in each square
NSQUARES = *scalars*	Number of squares to form (i.e. number of treatment factors to generate)
SEED = *scalars*	Seed to be used to randomize each design; a negative value implies no randomization
TREATMENTFACTORS = *pointers*	Pointer to identifiers for the treatment factors
ROWS = *factors*	Identifier for the row factor
COLUMNS = *factors*	Identifier for the column factor
MAXNSQUARES = *scalars*	Returns the maximum number of squares available with the specified number of rows and columns
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGLATIN)

Description

AGLATIN generates a set of orthogonal Latin squares, or a single square. It is easiest to use interactively. All the information required to generate the squares is then obtained by (clearly explained) questions. You need set the parameters only if you wish to anticipate some of the questions, or if you wish to use AGLATIN in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The size of the squares (i.e. the number of rows and columns) can be specified by the NROWS option, and the number of squares (i.e. the number of treatment factors to be generated) can be specified by the NSQUARES option. The MAXNSQUARES parameter can be used to ascertain how many squares are available. If this is set but NSQUARES is not set, the procedure then stops. Otherwise, when AGLATIN is being used interactively, if NSQUARES is unset you will be asked how many squares you want.

The squares are represented as a row factor, a column factor and NSQUARES treatment factors all of length NROWS**2. The ROWS and COLUMNS parameters can supply identifiers for the row and column factors, so that they are accessible outside the procedure. The TREATMENTFACTORS parameter can specify a pointer to supply identifiers for the treatment factors. For example, if there are two factors you could define their identifiers to be A and B by forming the pointer Tf (say) with the statement

```
POINTER [VALUES=A,B] Tf
```

and then setting TREATMENTFACTORS=Tf. Alternatively, and more succinctly, you could put

TREATMENTFACTORS=!p(A,B), where !p(A,B) is an unnamed pointer containing the required two identifiers.

The SEED parameter allows you to specify a seed to randomize the design. In a batch run, this has a default of –1, to suppress randomization. If SEED is unset in an interactive run, you will be asked to provide a seed (and again a negative value will leave the design unrandomized).

The PRINT option controls whether AGLATIN prints the design. The setting design prints it as a square table of treatment factors tabulated by the row and column factors, squares prints each treatment factor separately (again tabulated by rows and columns), and list prints row, column and treatment factor values as a list. By default, if you are running Genstat in batch, the nothing is printed. If you do not set PRINT when running interactively, AGLATIN will ask what you want to print. Similarly the ANALYSE option governs whether or not AGLATIN produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGLATIN assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run.

Options: PRINT, ANALYSE.
Parameters: NROWS, NSQUARES, SEED, TREATMENTFACTORS, ROWS, COLUMNS, MAXNSQUARES, STATEMENT.

Method

If the order of squares required is prime or any integer power of a prime number, the approach is to call GALOIS to obtain the multiplication table for the field GF[NROWS] and then cyclically develop the columns from the multiplication table to give the squares.

If the parameter NROWS is a composite number and, when decomposed into prime powers has a smallest prime power ($f = p^n$) greater than or equal to 3, then it is possible to construct $f-1$ orthogonal squares using the MacNeish-Mann method (Raghavarao 1971, p. 34). In the event that the smallest prime power is 2, it is possible to generate a single Latin square using the same approach. Essentially this process consists of embedding Galois fields one inside another. For each distinct prime factor the multiplication and addition tables from the corresponding fields are obtained by calling the procedure GALOIS. Note that while this method is general, it does not guarantee to find the maximum possible number of mutually orthogonal squares of any given order. For example, the following list details the number of squares generated for orders up to and including 40.

Order	Number of squares generated
12	2
15	2
20	3
21	2
24	2
28	3
33	2
35	4
39	2
40	4

For the orders 10, 14, 18 and 22, the procedure uses self-orthogonal squares (Franklin 1984) to give pairs of orthogonal Latin squares. The first rows of a square one order less than required are defined inside the procedure, these are developed by replacement of one of the broken diagonals with a new symbol and the addition of another row and column to produce a bordered cyclic Latin square. The second square in the pair is simply the transpose of the first.

References

Franklin, M.F. (1984). Cyclic generation of self-orthogonal Latin squares. *Utilitas Mathematica*, **25**, 135-146.

MacNeish, H.F. (1922). Euler's squares. *Annals of Mathematics*, **23**, 221-227

Mann, H.B. (1942). The construction of orthogonal Latin squares. *Annals of Mathematical Statistics*, **13**, 418-423.

Raghavarao, D. (1971). *Constructions and Combinatorial Problems in Design of Experiments*. John Wiley, New York.

See also

Procedures: AGCROSSOVERLATIN, AGSEMILATIN, AGQLATIN, GALOIS.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGLOOP

Generates loop designs e.g. for time-course microarray experiments (R.W. Payne).

Option

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGLOOP will ask whether the design is to be printed, in a batch run the default is not to print the design

Parameters

LEVELS = *scalars*	Number of treatments
INCREMENTS = *scalars*, *variates* or *pointers*	Increment or increments to be used to form the loops
SEED = *scalars*	Seed for randomization; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor
BLOCKS = *factors*	Identifier for the block (plate) factor
UNITS = *factors*	Identifier for the factor for the units within each block (or colours in a microarray experiment)
STATEMENT = *texts*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGLOOP)

Description

Loop designs are often used in two-colour microarray experiments. Here, each slide compares a pair of treatments, one of which is stained with a red dye and the other with a green dye. Suppose that the treatments are t_1, t_2 ... t_n. Then, before randomization in the basic form of the design, the first slide would compare t_1 (using red) with t_2 (using green), the second slide would compare t_2 (red) with t_3 (green), and the nth slide would compare t_n (red) with t_1 (green). The design has the advantage that treatments are balanced with colours. This basic form is also very effective for making comparisons between treatments that are adjacent in the sequence t_1 ... t_n, as might be the main point of interest when the treatments correspond to time.

Comparisons between more widely spaced treatments are less well estimated So an alternative possibility is to choose more than one increment, and construct additional cycles through the treatments using modulo arithmetic. The design is then known as an *interwoven loop design*. None of the increments, other than 1, must be a divisor of the number of treatments as its cycle would then fail to include all the treatments. For example, with 8 treatments an increment of 3 would be satisfactory (1, 4, 7, 2, 5, 8, 3, 6, 1) but 2 would not (1, 3, 5, 7, 1). Note also, that 5 (which is 8 – 3) would be equivalent to 3 (1, 6, 3, 8, 5, 2, 7, 4, 1); the treatments appear in the reverse order, so the adjacent pairs are the same.

AGLOOP is easiest to use interactively. It then asks questions to determine the necessary information to form the design: for example, the number of treatments and the increments to use. The parameters allow you to anticipate questions, or to define all the necessary information if you want to use AGLOOP in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The number of treatments can be defined using the LEVELS parameter. Similarly, the INCREMENTS parameter can supply a scalar defining a single increment, or a variate, or a pointer containing several scalars, to define several. The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is –1, to suppress randomization. If you do not set SEED when running interactively AGLOOP will ask for a seed, and again a

negative value suppresses any randomization. Note that, the randomization is constrained to ensure that the treatments remain balanced with colour.

The remaining parameters, TREATMENTS, BLOCKS and UNITS, allow you to specify identifiers for the factors representing treatments, blocks (or plates in a microarray experiment) and units within blocks (or colours in a microarray experiment). If these are not specified in a batch run, AGLOOP will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGLOOP will ask you to provide identifiers, and these will remain available after AGLOOP has finished running.

AGLOOP has a PRINT option which can be set to design to print the plan of the design. By default, if you are running Genstat in batch, neither are printed. If you do not set PRINT when running interactively, AGLOOP will ask whether or not you wish to print the design.

Option: PRINT.
Parameters: LEVELS, INCREMENTS, SEED, TREATMENTS, UNITS, STATEMENT.

Method
The QUESTION procedure is used to obtain the necessary details of the design. The design is then using the standard Genstat directives for calculation and manipulation.

See also
Procedures: AGBIB, AGREFERENCE.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Microarray data.

AGMAINEFFECT

Generates designs to estimate main effects of two-level factors (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (design, catalogue); if unset in an interactive run AGMAINEFFECT will ask whether the design or catalogue are to be printed, in a batch run the default is not to print anything
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run
FOLDED = *string token*	Whether to include an extra "folded" replicate with the levels of each factor interchanged (no, yes); default no
SEED = *scalar*	Seed to be used to randomize each design; a negative value implies no randomization
STATEMENT = *texts*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGMAINEFFECT)

Parameter

TREATMENTFACTOR = *factors*	Treatment factors

Description

AGMAINEFFECT generates designs for estimating main effects of factors with two levels, using a minimum number of experimental units; see Plackett & Burman (1946). The numbers of treatment factors for which designs are available can be printed by setting option PRINT=catalogue. They are, however, all expressible as $4n-1$ for some integer n. The treatment factors are listed using the TREATMENTFACTOR parameter. If this is omitted in an interactive run, you will be asked how many factors you want and their names.

The basic design allows the main effects to be estimated, but has no residual degrees of freedom. This is fine if you merely want to screen the main effects to identify the largest. Otherwise you can generate a design for more factors than are needed, and then use the degrees of freedom of the unnecessary factors to provide the residual. Alternatively, if you set option FOLDED=yes, AGMAINEFFECT will include a "folded" replicate of the design: this is identical to the initial replicate except that the levels of the factors are swapped (level one instead of level two and vice versa). This particular arrangement has the advantage that no main effect is aliased with any first-order interaction.

The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is -1, to suppress randomization. If you do not set SEED when running interactively AGMAINEFFECT will ask for a seed, and again a negative value suppresses any randomization. The PRINT option can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGMAINEFFECT will ask whether or not you wish to print the design. Similarly the ANALYSE option governs whether or not AGMAINEFFECT produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGMAINEFFECT assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run. The ANOVA option ORTHOGONAL is set to yes for the analysis. (If this is not done, the larger designs can take a very long time to analyse.)

The STATEMENT option allows you to save a Genstat text structure containing a command to recreate the design. This is particularly useful if AGMAINEFFECT is being used interactively, and the information to define the design has been provided in response to questions from the procedure.

Options: PRINT, ANALYSE, FOLDED, SEED, STATEMENT.
Parameter: TREATMENTFACTOR.

Method
The designs are based on Hadamard matrices, which can be generated by procedure FHADAMARDMATRIX. The QUESTION procedure is used to obtain the necessary details of the design and this is then generated by the standard Genstat manipulation directives.

Reference
Plackett, R.L. & Burman, J.P. (1946). The design of optimum factorial experiments. *Biometrika*, **33**, 305-325 & 328-332.

See also
Directive: AFRESPONSESURFACE.
Procedures: AGBOXBEHNKEN, AGCENTRALCOMPOSITE, AGFACTORIAL, FHADAMARDMATRIX.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AGNATURALBLOCK

Forms 1- and 2-dimensional designs with blocks of natural size (P.D. Johnstone & D.B. Baird).

Options

PRINT = *string token*	Controls printed output (design, search); default desi
DESIGNTYPE = *string token*	Type of design to create (block, rowcolumn); default rowc
NSIMULATIONS = *scalar*	Number of randomizations to search to find the best design; default 1000
SEED = *scalar*	Seed for the randomization; default 0
FIRSTPLOT = *string token*	Defines the starting location for allocating plots to the row-by-column grid (lowleft, lowright, upleft, upright); default uple
FILLMETHOD = *string token*	Defines the order in which the plots are filled (colserpentine, colbycol, rowserpentine, rowbyrow); default rows

Parameters

LEVELS = *scalars* or *variates*	Defines the levels of the treatment factor for each design
NROWS = *scalars*	Number of rows in the smallest rectangle containing the layout of each design; not required if the ROWS parameter is set to a factor with values
NCOLUMNS = *scalars*	Number of columns in the smallest rectangle containing the layout of each design; not required if the COLUMNS parameter is set to a factor with values
NUNITS = *scalar*	Number of plots that will be assigned a treatment in each design; not required if the either the ROWS or COLUMNS parameter is set to a factor with values
TREATMENTS = *factors*	Saves the treatment allocation for each design
ROWS = *factors*	Defines or saves the row locations of the plots to receive treatments in each design
COLUMNS = *factors*	Defines or saves the column locations of the plots to receive treatments in each design
BLOCKS = *factors*	Defines or saves the allocation of the plots to blocks
PLAN = *matrices*	Saves the treatment layout in each design

Description

This procedure uses random generation for obtaining block and row-column designs and selects the best of those designs using a criterion similar in effect to the M,S-optimality criterion of Shah (1960). The method used does not have any restrictions on the layout of plots in the array, which can be incomplete, or on the number of treatments or replicates per treatments, which may be unequal, as have some other design generators.

The number of treatments is specified by the LEVELS parameter, either as a scalar (defining the number explicitly) or by a variate (defining a number for each level), as in the FACTOR directive.

The DESIGNTYPE option specifies whether to form a row-column or a block design. With a row-column design (DESIGNTYPE=rowcolumn), the layout of the plots can be defined in two factors supplied by the ROWS and COLUMNS parameters. Alternatively, if all plots are present, then just the number of rows and columns in the row-column array can be specified by the

NROWS and NCOLUMNS parameters. If the ROWS or COLUMNS option is set to a factors whose values have not been defined, the factor values will be set up to define a regular grid of plots. The numbers of rows and columns must then defined by the corresponding NROWS or NCOLUMNS option. The NUNITS parameter defines the number of plots in the design. If this is not specified, the number of plots to allocate treatments is taken from either the number of values in ROWS or COLUMNS, or else NROWS × NCOLUMNS, if ROWS or COLUMNS are not specified. Thus, you only need to specify NUNITS if you are not using the full grid of NROWS × NCOLUMNS values, or you wish to limit the units used in ROWS and COLUMNS.

For one-way block design (DESIGNTYPE=block) with unequal block sizes, the BLOCKS parameter must be set to a factor giving the block number for each plot. The block design can allocated to a two dimensional layout by specifying the position of each unit in two factors supplied by the ROWS and COLUMNS parameters. Alternatively, you can specifying the size of the grid with the NROWS and NCOLUMNS parameters, in which case the LAYOUT option is used to allocate units to the grid. Again, if the ROWS or COLUMNS option is set to a factors whose values have not been defined, the factor values will be set up to define a regular grid of plots. With a block design, if NUNITS is unset, the number of plots is taken from the number of values of the BLOCKS factor. The FIRSTPLOT and FILLMETHOD options control how the plots are allocated to to the row and column locations when these are not defined by ROWS and COLUMNS factors. The FIRSTPLOT option defines the starting location as follows:

lowleft	left-hand plot at the bottom of the grid;
lowright	right-hand plot at the bottom of the grid;
upleft	left-hand plot at the top of the grid (default);
upright	right-hand plot at the top of the grid.

The FILLMETHOD option defines the order in which the plots are then filled:

colserpentine	column-by-column in a serpentine way e.g. top-to-bottom, and then bottom to top;
colbycol	column-by-column taking the same direction for every column;
rowserpentine	in a serpentine way e.g. left-to-right, and then right-to-left (default);
rowbyrow	row-by-row taking the same direction for every row.

The NSIMULATIONS option defines the number of random designs to be searched. The SEED option specifies the starting seed for the randomization process; the default of zero continues an existing sequence of random numbers if any have already been used in this Genstat job, or initializes the seed automatically. The resulting optimal treatment allocation can be saved, either in a factor specified by the TREATMENTS parameter, or by setting the PLAN parameter to a matrix to save the two-dimensional layout of the treatments.

Printed output is controlled by the PRINT option, with settings:

design	to print the best design;
search	to print the current best design during the search.

Options: PRINT, DESIGN, NSIMULATIONS, SEED, FIRSTPLOT, FILLMETHOD.

Parameters: LEVELS, NROWS, NCOLUMNS, NUNITS, TREATMENTS, ROWS, COLUMNS, BLOCKS, PLAN.

Method

For a row-column design the treatments are allocated in random order down the columns, while for a block design they are allocated one replicate at a time. A number of designs (specified by the NSIMULATIONS option) are generated. The design is chosen that minimizes both the range and the average of the standard errors of differences for all pairwise comparisons between elements of the generalized least squares estimates of treatment effects.

Action with RESTRICT

If any of the factors is restricted, only the part of the design not excluded by the restriction will be generated.

References

Johnstone, P.D. (2003). Random generation and selection of one- and two-dimensional designs for experiments on blocks of natural size. *Journal of Agricultural, Biological and Environmental Statistics*, **8**, 67-74.

Shah, K.R. (1960). Optimality criteria for incomplete block designs. *Annals of Mathematical Statistics*, **22**, 235-247.

See also

Genstat Reference Manual 1 Summary section on: Design of experiments.

AGNEIGHBOUR

Generates neighbour-balanced designs (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (catalogue, design); if unset in an interactive run AGNEIGHBOUR will ask whether the design is to be printed, in a batch run the default is not to print anything
METHOD = *string token*	Type of design, *n*−1 blocks of *n* plots, or *n* blocks of *n*−1 plots (N_1BLOCKS, NBLOCKS); if unset in an interactive run AGNEIGHBOUR will ask about the type of design, in a batch the default is assumed to be n blocks of n−1 plots

Parameters

LEVELS = *scalars*	Number of treatments
SEED = *scalars*	Seed for randomization; in batch there is a default of 12345
TREATMENTS = *factors*	Identifier for the treatment factor
BLOCKS = *factors*	Identifier for the factor to index the blocks within replicates
UNITS = *factors*	Identifier for the factor to index the units within each block, or the periods of a cyclic change-over design
LEFTNEIGHBOUR = *factors*	To save the treatment on the left neighbouring unit
RIGHTNEIGHBOUR = *factors*	To save the treatment on the right neighbouring unit
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGNEIGHBOUR)

Description

In experiment designs it is often necessary to allow for the possibility that a treatment may have an effect on neighbouring plots, as well as on its own plot. For example, in variety trials, tall varieties may shade their neighbours. Likewise, in experiments on insecticides and fungicides, there may be cross infection from plots receiving control or ineffective treatments to neighbouring plots. In both of these examples the neighbour effect may depend on direction (for example of prevailing wind or of sunlight), so it is usual to distinguish between left and right neighbours. To avoid bias when comparing the effects of treatments in these situations, it is important to ensure that no treatment is unduly disadvantaged by its neighbours. This is best done by using a neighbour-balanced design. Here the allocation of treatments is such that every treatment occurs equally often with each other treatment as a right neighbour, and as a left neighbour.

The table below shows a design for five treatments in 5 blocks of size 4. Notice that in addition to the experimental plots, the design also needs a line of treated border plots on each side. These provide the neighbouring treatments for plots 1 and 4, but do not provide yields or other response variables. The border plots are not included in the generated factor values.

Plot	border	1	2	3	4	border
Block						
1	5	2	3	1	5	2
2	3	5	4	1	3	5
3	4	2	5	3	4	2
4	1	4	3	2	1	4
5	4	5	1	2	4	5

Methods of constructing and randomizing neighbour-balanced designs for *n* treatments in either

n blocks of $n-1$ plots or in $n-1$ blocks of n plots are described by Azais, Bailey & Monod (1993) together with generators for $3 \leq n \leq 16$ (other than for n=4 or 6 with $n-1$ blocks of size n, for which no designs are available). AGNEIGHBOUR uses these methods and generators, together with some further generators for blocks of $n-1$ plots formed using the method of Azais (1987).

AGNEIGHBOUR is easiest to use interactively. It then asks questions to determine the necessary information to form the design, and indicates the numbers of treatments for which designs are available. The options and parameters allow you to anticipate questions, or to define all the necessary information if you want to use AGNEIGHBOUR in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The first question, which can be anticipated by setting the METHOD option, determines the type of design: n blocks of $n-1$ plots (METHOD=nblocks) or in $n-1$ blocks of n plots (METHOD=n_1blocks). The default in batch is n_1block. The PRINT option controls printed output, with setting design to print a plan of the design, and catalogue to print a list of the available designs. By default, if you are running Genstat in batch, nothing is printed. If you do not set PRINT when running interactively, AGNEIGHBOUR will ask whether or not you wish to print the design, after it has been generated.

The number of treatments can be defined using the LEVELS parameter. This can be set to zero to avoid constructing a design, as may be required if you merely wish to print the catalogue. The SEED parameter allows you to specify a seed to be used to randomize the design. If you do not set SEED when running interactively AGNEIGHBOUR will ask for a seed. In batch there is a default of 12345. Setting a negative seed suppresses any randomization. Parameters TREATMENTS, BLOCKS and UNITS, allow you to specify identifiers to save the treatment, the block and unit-within-block factors. If these are not specified in a batch run, AGNEIGHBOUR will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGNEIGHBOUR will ask you to provide identifiers and these will remain available after AGNEIGHBOUR has finished running. There are also parameters LEFTNEIGHBOUR and RIGHTNEIGHBOUR to allow you to save the treatments on the left and right neighbouring plots.

Some of the designs are such that each ordered pair of treatments occurs the same number of times as the left and right neighbours of some other treatment, the design is then said to be neighbour-balanced at distance 2. These designs have the further advantage that they are balanced if analysed with ANOVA with

```
BLOCKSTRUCTURE     BLOCKS / UNITS
TREATMENTSTRUCTURE TREATMENTS+ LEFTNEIGHBOUR + RIGHTNEIGHBOUR
```

Options: PRINT, METHOD.

Parameters: LEVELS, SEED, TREATMENTS, BLOCKS, UNITS, LEFTNEIGHBOUR, RIGHTNEIGHBOUR, STATEMENT.

Method

The generation methods are described by Azais, Bailey & Monod (1993). The QUESTION procedure is used to obtain the necessary details of the design and this is then generated by the standard Genstat manipulation directives.

References

Azais, J.M-. (1987). Design of experiments for studying intergenotypic competition. *Journal of the Royal Statistical Society Series B*, **49**, 334-345.

Azais, J.M-., Bailey, R.A. & Monod, H. (1993). A catalogue of efficient neighbour designs with border plots. *Biometrics*, **49**, 1252-1261.

See also
Procedures: AGCROSSOVERLATIN, AGQLATIN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AGNONORTHOGONALDESIGN

Generates non-orthogonal split-plot and other hierarchical designs (B. M. Parker).

Options

PRINT = *string token*	Controls printed output (design, debug); default * i.e. nothing
METHOD = *string token*	Specifies the algorithm to use (jonesgoos, trincagilmour); default trin
CRITERION = *string token*	Optimality criterion (a, d); default a
MODELMATRIX = *matrix*	Defines the model to be estimated
NSTARTS = *scalar*	Number of random starts for the jg algorithm; default 10
NTRIES = *scalar*	Number of exchanges to try from each start; default 10000
MINIMUM = *scalar*	Minimum value for levels; default −1
MAXIMUM = *scalar*	Maximum value for levels; default 1
SEED = *scalar*	Seed for the random numbers used by the algorithms; default 0

Parameters

BLOCKFACTOR = *factors*	Specifies the identifier for the block factor used to index the units of the whole-plots, the sub-plots and, if required, the sub-sub-plots
TREATMENTFACTORS = *factors* or *pointers*	Specifies the identifier of the treatment factor or factors applied to the whole, sub-plots and sub-sub-plots
BLEVELS = *scalars*	Numbers of levels for the block factors
LEVELS = *scalars* or *pointers*	Numbers of levels for the treatment factors
VARIANCES = *scalars*	Variances for the strata

Description

Many industrial and agricultural experiments involve some factors whose levels are harder to set than others. For example, in an agriculture experiment, some factors may only be applied to whole-plots, whereas some may be applied to sub-plots. This agricultural background means these are referred to as "split-plot" experiments. Where a further sub-sub-plot factor is investigated, these are known as split-split-plot experiments. In industrial or laboratory applications, the designs are known as "multi-strata" designs. The factors that are hardest to set should be in stratum 1, the next hardest in stratum 2, and so on. The results of these experiments are typically analysed using a mixed-model analysis. As randomization is restricted, care must be taken to find designs which are efficient, in the sense that the unknown parameters in a model are estimated well. (Note, though, that this restriction on randomization often precludes other desirable qualities, such as orthogonality.)

The algorithm to use is specified by the METHOD option. The default setting, trincagilmour, uses the algorithm of Trinca & Gilmour (2014). This can be used for designs with any number of strata (although in practice, designs with more than three strata are rare). It works on each stratum in turn to find a design which is optimal for that stratum, together with any terms that appear in that stratum and higher strata. The next lowest stratum is then considered, until all strata are exhausted. The NTRIES option specifies how many random exchanges to attempt; default 10,000.

The CRITERION option specifies the optimality criterion to use in the Trinca & Gilmour algorithm to assess the quality of designs: either A_s (default) or D_s. The aim is to estimate a

function of the parameters of the model with maximum efficiency. Block effects are considered nuisance parameters, and in the case of A_s optimality in a second order model, quadratic effects are weighted as 0.25 and other effects are weighted as 1.

An important unknown factor in designing multi-strata experiments correctly is the ratio of (usually unknown) variances at each stratum level. A locally optimal design could, in theory, be found for each value of this variance ratio. Whilst the Trinca & Gilmour algorithm does not find an absolutely locally optimal design, in practice locally optimal designs can perform poorly if the variance ratio is mis-specified. The algorithm finds designs that are robust against mis-specification of the variance ratio, and which should perform well when the ratio is unknown (which is the usual situation in practice). As well as being robust, the algorithm is quick to run, even for large designs.

Alternatively, setting `METHOD=jonesgoos`, selects the algorithm of Jones & Goos (2007). This implements an exchange algorithm to calculate (locally) D-optimal designs for split-plot designs. It is a candidate-set free algorithm, which helps to make it relatively fast to run. An important parameter that must be specified, is the variance ratio between the whole-plots and the sub-plots. In general, if a D-optimal split-plot design is required, the variance ratio should be known. The Jones and Goos design should then be better than a Trinca and Gilmour design. The `NSTARTS` option specifies the number of random starts to use with the Jones & Goos algorithm; default 10

The `BLOCKFACTOR` parameter lists the block factors: first the factor to index the units of the whole plots, then a factor to index the sub-plots, and so on, as required. The `BLEVELS` parameter must be set to specify the numbers of levels of the block factors, and the `VARIANCES` parameter can be set to supply the stratum variances. The number of plots (or runs) in the experiment is specified by the `NUNITS` option.

The `TREATMENTFACTORS` parameter defines factors for the treatments applied to the units of the strata, and the `LEVELS` defines their numbers of levels. If several factors are to be applied to a particular stratum, the factors and their levels should each be put into a pointer.

The `MINIMUM` and `MAXIMUM` options specify minimum and maximum possible values, respectively, for the treatment factors; default –1 and 1.

The `MODELMATRIX` option defines a polynomial model that will be fitted to the results of the experiment. This is a matrix, with a row for each model term, and a column for each treatment factor. The entries in the rows specify the powers of the factors involved in the corresponding polynomial term.

The `SEED` option specifies the seed for the random numbers used by the algorithms. The default of 0 continues an existing sequence or, if none, obtains a seed automatically from the system clock.

The `PRINT` option can be set to `design`, to print the design. There is also a setting `debug` to provide debugging information for the algorithm.

Options: `PRINT`, `METHOD`, `CRITERION`, `MODELMATRIX`, `NSTARTS`, `NTRIES`, `MINIMUM`, `MAXIMUM`, `SEED`.

Parameters: `BLOCKFACTOR`, `TREATMENTFACTORS`, `BLEVELS`, `LEVELS`, `VARIANCES`.

References

Jones, B. & Goos, P. (2007). A candidate set free algorithm for generating D optimal split plot designs. *Applied Statistics*, **56**, 347-364.

Trinca, L.A. & Gilmour, S.G. (2014). Improved Split-Plot and Multi-Stratum Designs. *Technometrics*, DOI:10.1080/00401706.2014.915235.

See also

Procedure: AGHIERARCHICAL.

Genstat Reference Manual 1 Summary section on: Design of experiments, Analysis of variance.

AGQLATIN

Generates complete and quasi-complete Latin squares (R.W. Payne).

Options

PRINT = *string token*	Controls printing of the design (design); if unset in an interactive run AGQLATIN will ask whether the design is to be printed, in a batch run the default is not to print anything
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run

Parameters

NROWS = *scalars*	Specifies the number of rows (and columns) in the square
SEED = *scalars*	Seed to be used to randomize each design; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor
ROWS = *factors*	Identifier for the row factor
COLUMNS = *factors*	Identifier for the column factor
STATEMENT = *texts*	Saves a command to recreate each design (useful if the design information has been specified in response to questions from AGQLATIN)

Description

A complete Latin square is a Latin square in which each ordered pair of treatments appears exactly once within the rows of the square, and exactly once within the columns. For example, in the four-by-four square below, the pair (1,2) is in row 1 (and only in row 1) while the pair (2,1) is only in row 4. Likewise (1,2) is only only in column 1 and (2,1) only in column 4.

```
Columns   1   2   3   4
   Rows
      1   1   2   4   3
      2   2   3   1   4
      3   4   1   3   2
      4   3   4   2   1
```

A quasi-complete Latin has similar properties, but here each unordered pair occurs exactly twice within the rows, and exactly twice within the columns. See, for example, the five-by-five Latin square below.

```
Columns   1   2   3   4   5
   Rows
      1   1   2   5   3   4
      2   2   3   1   4   5
      3   5   1   4   2   3
      4   3   4   2   5   1
      5   4   5   3   1   2
```

Complete Latin squares can be constructed for any even number of rows, while quasi-complete squares are available for any odd number of rows. Designs based on these squares are useful for example in experiments where there is the possibility of interference between a plot and its neighbours. Complete Latin squares should be used if the interference is likely to be directional,

as for example in a field experiment to assess fungicides where spores may be carried from one plot to another by a prevailing wind. Otherwise the choice of design will depend upon wether an odd or even number of treatments is required.

AGQLATIN is easiest to use interactively. All the information required to generate the design is then obtained by (clearly explained) questions. You need set the parameters only if you wish to anticipate some of the questions, or if you wish to use AGQLATIN in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The size of the square (i.e. the number of rows and columns) can be specified by the NROWS option. The ROWS, COLUMNS and TREATMENTS parameters can supply identifiers for the row, column and treatment factors, so that they are accessible outside the procedure.

The SEED parameter allows you to specify a seed to randomize the design, by making a random permutation of the treatment labels. In a batch run, SEED has a default of –1, to suppress randomization. If SEED is unset in an interactive run, you will be asked to provide a seed (and again a negative value will leave the design unrandomized).

The PRINT option can be set to design to print the design. By default, if you are running Genstat in batch, the nothing is printed. If you do not set PRINT when running interactively, AGQLATIN will ask what you want to print. Similarly the ANALYSE option governs whether or not AGQLATIN produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGQLATIN assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run.

Options: PRINT, ANALYSE.
Parameters: NROWS, SEED, TREATMENTS, ROWS, COLUMNS, STATEMENT.

Method

AGQLATIN uses the method of Williams (1949), which is based upon terraced groups (Bailey 1984).

References

Bailey, R.A. (1984). Quasi-complete Latin squares: consatruction and randomization. *Journal of the Royal Statistical Society Series B*, **46**, 323-334.

Williams, E.J. (1949). Experimental designs balanced for the estimation of residual effects of treatments. *Australian Journal of Scientific Research Series A*, **2**, 149-168.

See also

Procedures: AGCROSSOVERLATIN, AGLATIN, AGNEIGHBOUR, AGSEMILATIN.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGRAPH

Plots tables of means from ANOVA (R.W. Payne).

Options

GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
METHOD = *string token*	What to plot (means, lines, data, barchart, splines); default mean
XFREPRESENTATION = *string token*	How to label the *x*-axis (levels, labels); default labels uses the XFACTOR labels, if available
PSE = *string token*	What to plot to represent variation (differences, lsd, means, allmeans); default diff
LSDLEVEL = *scalar*	Significance level (%) to use for least significant differences; default 5
DFSPLINE = *scalar*	Number of degrees of freedom to use when METHOD=splines
YTRANSFORM = *string tokens*	Transformed scale for additional axis marks and labels to be plotted on the right-hand side of the y-axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden i.e. none
PENYTRANSFORM = *scalar*	Pen to use to plot the transformed axis marks and labels; default * selects a pen, and defines its properties, automatically
†KEYMETHOD = *string token*	What to use for the key descriptions when GROUPS specifies more than one factor (labels, namesandlabels); default name
†PLOTTITLEMETHOD = *string token*	What to use for the titles of the plots when TRELLISGROUPS specifies more than one factor (labels, namesandlabels); default name
†PAGETITLEMETHOD = *string token*	What to use for the titles of the pages when PAGEGROUPS specifies more than one factor (labels, namesandlabels); default name
†USEAXES = *string token*	Which aspects of the current axis definitions of window 1 to use (none, limits, marks, mpositions, nsubticks,); default none
SAVE = *ANOVA* or *regression save structure*	Save structure to provide the table of means; default uses the save structure from the most recent ANOVA

Parameters

XFACTOR = *factors*	Factor providing the *x*-values for each plot
GROUPS = *factors* or *pointers*	Factor or factors identifying groups of points in each plot; by default chosen automatically
TRELLISGROUPS = *factors* or *pointers*	Factor or factors specifying the different plots of a trellis plot of a multi-way table
PAGEGROUPS = *factors* or *pointers*	Factor or factors specifying plots to be displayed on different pages
NEWXLEVELS = *variates*	Values to be used for XFACTOR instead of its existing

	levels
TITLE = *texts*	Title for the graph; default defines a title automatically
YTITLE = *texts*	Title for the y-axis; default is to use the identifier of the y-variate, or to have no title if this is unnamed
XTITLE = *texts*	Title for the x-axis; default is to use the identifier of the XFACTOR
PENS = *variates*	Defines the pen to use to plot the points and/or line for each group defined by the GROUPS factors

Description

AGRAPH plots tables of means from an ANOVA analysis. In its simplest form, the behaviour of AGRAPH depends on the model. If the treatment model contains only main effects, it plots the means for the first factor in the model. Otherwise it looks for the first treatment term involving two factors; it then plots the means with one of these factors as the x-axis, and the second as a grouping factor with levels identified by different plotting colours and symbols. By default, the means are from the most recent ANOVA. However, you can plot means from an earlier analysis, by using the SAVE option of AGRAPH to specify its save structure (saved using the SAVE parameter of the ANOVA command that performed the analysis).

Usually, each mean is represented by a point. However, with high-resolution plots, the METHOD option can be set to lines to draw lines between the points, or data to draw just the lines and then also plot the original data values, or barchart to plot the means as a barchart, or splines to plot the points together with a smooth spline to show the trend over each group of points. The DFSPLINE specifies the degrees of freedom for the splines; if this is not set, 2 d.f. are used when there are up to 10 points, 3 if there are 11 to 20, and 4 for 21 or more. The GRAPHICS option controls whether a high-resolution or a line-printer graph is plotted; by default GRAPHICS=high.

The PSE option specifies the type of error bar to be plotted with the means, with settings:

differences	average standard error of difference;
lsd	average least significant difference;
means	average effective standard error for the means;
allmeans	plots plus and minus the effective standard error around every mean.

The LSDLEVEL option sets the significance level (%) to use for the least significant differences (default 5). The allmeans setting is often unsuitable for plots other than barcharts when there are GROUPS, as the plus/minus e.s.e. bars may overlap each other.

You can define the table of means to plot explicitly, by specifying its classifying factors using the XFACTOR, GROUPS, TRELLISGROUPS and PAGEGROUPS parameters. The XFACTOR parameter defines the factor against whose levels the means are plotted. With a multi-way table, there will be a plot of means against the XFACTOR levels for every combination of levels of the factors specified by the GROUPS, TRELLISGROUPS and PAGEGROUPS parameters. The GROUPS parameter specifies factors whose levels are to be included in a single window of the graph. So, for example, if you specify

```
AGRAPH [METHOD=line] XFACTOR=A; GROUPS=B
```

AGRAPH will produce a plot of the means in a single window with factor A on the x-axis, and a line for each level of the factor B. You can set GROUPS to a pointer to specify several factors to define groups. For example

```
POINTER [VALUES=B,C] Groupfactors
AGRAPH [METHOD=line] XFACTOR=A; GROUPS=Groupfactors
```

to plot a line for every combination of the levels of factors B and C. Similarly, the TRELLISGROUPS option can specify one or more factors to define a trellis plot. For example,

```
AGRAPH [METHOD=line] XFACTOR=A; GROUPS=B; TRELLISGROUPS=C
```

will produce a plot for each level of C, in a trellis arrangement; each plot will again have factor A on the x-axis, and a line for each level of the factor B. Likewise, the PAGEGROUPS parameter can specify factors whose combinations of levels are to be plotted on different pages. So

```
AGRAPH [METHOD=line] XFACTOR=A; GROUPS=B; PAGEGROUPS=C
```

will produce a plot for each level of C, but now on separate pages. Multi-way tables can plotted even if the corresponding model term was not in the ANOVA analysis. For example you can plot a two-way table even if the analysis contained only the main effects of the two factors; however, the lines will then all be parallel and no standard errors or LSDs can be included.

The NEWXLEVELS parameter enables different levels to be supplied for XFACTOR if the existing levels are unsuitable. If XFACTOR has labels, these are used to label the x-axis unless you set option XFREPRESENTATION=levels.

The TITLE, YTITLE and XTITLE parameters can supply titles for the graph, the y-axis and the x-axis, respectively. The symbols, colours and line styles that are used in a high-resolution plot are usually set up by AGRAPH automatically. If you want to control these yourself, you should use the PEN directive to define a pen with your preferred symbol, colour and line style, for each of the groups defined by combinations of the GROUPS factors. The pen numbers should then be supplied to AGRAPH, in a variate with a value for each group, using the PENS parameter.

The YTRANSFORM option allows you to include additional axis markings, transformed onto another scale, on the right-hand side of the y-axis. Suppose, for example, suppose you have analysed a variate of percentages that have been transformed to logits. You might then set YTRANSFORM=ilogit (the inverse-logit transformation) to include markings in percentages alongside the logits. The settings are the same as those of the TRANSFORM parameter of AXIS (which is used to add the markings). You can control the colours of the transformed marks and labels, by defining a pen with the required properties, and specifying it with the PENYTRANSFORM option. Otherwise, the default is to plot them in blue.

When there is more than one GROUPS factor, the KEYMETHOD controls whether to use the factor names with their labels (or levels for factors with no labels) or just the labels (or levels) in the key descriptions. The default is to use the names and the labels (or levels). Similarly, the PLOTTITLEMETHOD specifies what to use for the titles of the plots when there is more then one TRELLISGROUPS factor, and the PAGETITLEMETHOD specifies what to use for the titles of the plots when there is more then one PAGEGROUPS factor. You can setKEYMETHOD=* to have no key at all.

The USEAXES option allows you to control various aspects of the axes. First you need to use the XAXIS and YAXIS directives to define them for window 1. Then specify which of the aspects of the axes in window 1 are to be used by DTABLE, by specifying USEAXES with the following settings:

limits	y- and x-axis limits (LOWER and UPPER parameters);
marks	location and labelling of the tick marks (MARKS, LABELS, LDIRECTION, LROTATION, DECIMALS, DREPRESENTATION, and VREPRESENTATION parameters);
mpositions	positions of the tick marks (MPOSITION parameter); and
nsubticks	number of subticks per interval (NSUBTICKS parameter).

By default none are used.

For compatibility with previous releases, AGRAPH allows you to plot predicted means from an analysis by the AUNBALANCED procedure (which uses the Genstat regression commands). However, procedure AUGRAPH (new in Release 13) is now recommended instead. Also, in Release 13, a new procedure DTABLE was included to plot a user-supplied table. Previously this could be done using the MEANS parameter of AGRAPH, which has now been withdrawn.

Options: GRAPHICS, METHOD, XFREPRESENTATION, PSE, LSDLEVEL, DFSPLINE, YTRANSFORM,

PENYTRANSFORM, KEYMETHOD, PLOTTITLEMETHOD, PAGETITLEMETHOD, USEAXES, SAVE.
Parameters: XFACTOR, GROUPS, TRELLISGROUPS, PAGEGROUPS, NEWXLEVELS, TITLE, YTITLE, XTITLE, PENS.

See also
Directive: ANOVA.
Procedures: APLOT, AUGRAPH, RGRAPH, VGRAPH.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AGREFERENCE

Generates reference-level designs e.g. for microarray experiments (R.W. Payne).

Option

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGREFERENCE will ask whether the design is to be printed, in a batch run the default is not to print the design

Parameters

LEVELS = *scalars*	Number of treatments
REFLEVEL = *scalars*, *variates* or *pointers*	Reference level(s); if this is unset in an interactive run you will be asked which reference level or levels you want, in a batch run the default is level 1
REFUNIT = *scalars*, *variates* or *pointers*	Unit(s) to which to allocate the reference level(s); if this is unset in an interactive run you will be asked which reference level or levels you want, in a batch run the default is to choose the unit at random within each block
SEED = *scalars*	Seed for randomization; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor
BLOCKS = *factors*	Identifier for the block (plate) factor
UNITS = *factors*	Identifier for the factor for the units within each block (or colours in a microarray experiment)
STATEMENT = *texts*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGREFERENCE)

Description

Reference-level designs can be useful in experiments where the main aim is to compare new treatments with a control, or reference, treatment. The design is made up of blocks of size two, each of which compares the control with one of the new treatments. So, if there are four treatment and the reference treatment is treatment 1, the basic design would have three blocks containing the pairs of treatments (1, 2), (1, 3) and (1, 4). The design is particularly relevant to two-colour microarray experiments, where each slide compares a pair of treatments, one of which is stained with a red dye and the other with a green dye.

AGREFERENCE is easiest to use interactively. It then asks questions to determine the necessary information to form the design: for example, the number of treatments, and which of the treatments is the control. The parameters allow you to anticipate questions, or to define all the necessary information if you want to use AGREFERENCE in batch. If, however, you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The number of treatments (including the reference treatment) can be defined using the LEVELS parameter. Similarly, the REFLEVEL parameter can define the reference treatment or treatments. You can supply a scalar to define a single reference treatment, or a variate, or a pointer containing several scalars, to define several. The REFUNIT similarly indicates which unit is to be used for the reference treatment within each block. (In a microarray experiment, the "unit" would be the colour, red or green, and each block would be a slide.) The numbers specified for the reference unit should be either 1 to use the first unit, or 2 to use the second, or 0 to use a unit

selected at random for each block.

You can thus construct several versions of the basic design, each using a different reference level and/or unit. For example

```
VARIATE     [VALUES=1,2] V12
AGREFERENCE 4; REFLEVEL=1; REFUNIT=V12
```

would define a design with two blocks to compare the reference treatment with each of the other three treatments. In one of the blocks the reference treatment would be on unit one (e.g. colour red on a microarray plate) and in the other it would be on unit two (e.g. colour green). Similarly

```
AGREFERENCE 4; REFLEVEL=V12; REFUNIT=1
```

would generate two versions of the basic design. The first would have treatment one as the reference, and the second would have treatment two as the reference (both allocated to unit one).

```
AGREFERENCE 4; REFLEVEL=V12; REFUNIT=V12
```

would generate two versions of the basic design. The first would have treatment one as the reference (allocated to unit 1), and the second would have treatment two as the reference (allocated to unit 2).

The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is −1, to suppress randomization. If you do not set SEED when running interactively AGREFERENCE will ask for a seed, and again a negative value suppresses any randomization. Note that the randomization takes account of the settings of the REFUNIT parameter.

The remaining parameters, TREATMENTS, BLOCKS and UNITS, allow you to specify identifiers for the factors representing treatments, blocks (or plates in a microarray experiment) and units within blocks (or colours in a microarray experiment). If these are not specified in a batch run, AGREFERENCE will use identifiers that are local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGREFERENCE will ask you to provide identifiers, and these will remain available after AGREFERENCE has finished running.

AGREFERENCE has a PRINT option which can be set to design to print the plan of the design. By default, if you are running Genstat in batch, neither are printed. If you do not set PRINT when running interactively, AGREFERENCE will ask whether or not you wish to print the design.

Option: PRINT.
Parameters: LEVELS, REFLEVEL, REFUNIT, SEED, TREATMENTS, BLOCKS, UNITS, STATEMENT.

Method

The QUESTION procedure is used to obtain the necessary details of the design. The design is then using the standard Genstat directives for calculation and manipulation.

See also

Procedures: AGBIB, AGLOOP.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Microarray data.

AGSEMILATIN

Generates semi-Latin squares (W. van den Berg).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGSEMILATIN will ask whether the design is to be printed, in a batch run the default is not to print anything
METHOD = *string token*	Method to use to construct the semi-Latin square (Trojan, interleaving, inflated); if unset in an interactive run AGSEMILATIN will ask what type is required, in a batch run the default is Trojan
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run

Parameters

NROWS = *scalars*	Number of rows and columns of the semi-Latin square
NUNITS = *scalars*	Number of units (i.e. treatments) within each block
SEED = *scalars*	Seed for randomization; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor
ROWS = *factors*	Identifier for the row factor
COLUMNS = *factors*	Identifier for the column factor
UNITS = *factors*	Identifier for the unit factor
PSEUDOFACTOR = *factors*	Identifier for the pseudo-factor
STATEMENT = *texts*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGSEMILATIN)

Description

AGSEMILATIN generates the factors and pseudo-factor required to define a semi-Latin square. It also sets the block and treatment formulae (using the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives) to allow the design, if balanced, to be analysed by ANOVA.

An $(n \times n)/k$ semi-Latin square is like an $n \times n$ Latin square except that there are k letters in each cell. The combinations of the rows and columns of a semi-Latin square are called blocks. Each of the $n \times k$ letters occurs once in each row and once in each column. The design thus has n rows and columns, k (sub-) units within each row × column combination (or block), and $n \times k$ treatments. The analysis should contain strata for rows, columns, rows.columns and rows.columns.units, as well as treatment effects which may be estimated in either the rows.columns or the rows.columns.units strata. AGSEMILATIN enables you to construct three types of semi-Latin square.

Trojan squares: a Trojan square consist of a set of k mutually orthogonal $n \times n$ Latin squares, on k disjoint sets of treatments. Each block of the semi-Latin square contains the treatments which occur in the corresponding cell of all the individual squares (Bailey 1988). AGSEMILATIN can construct Trojan squares for any value of n for which a Graeco-Latin square exists. Thus, for example, no Trojan square exists for $n = 6$. In a Trojan square k must be greater than 1 and less than n (Edmondson 1998), and for some values of n, k must be less than that. The maximum values of k for n up to 15 for a Trojan square are

n:	3	4	5	7	8	9	11	12	13	14	15
k:	2	3	4	6	6	8	10	2	12	2	2

In a Trojan square, some treatment effects are estimated in both the rows.columns and the rows.columns.units strata, while others (which need to be represented by a pseudo-factor) are estimated only in the rows.columns.units stratum. Trojan squares are optimal semi-Latin squares (Bailey 1992).

Inflated Latin squares: an $(n \times n)/k$ inflated Latin square consists of an $n \times n$ Latin square with each letter replaced by k new symbols (Bailey 1988). AGSEMILATIN can construct inflated Latin squares for any value of n greater than 2, and any value of k greater than 1. The analysis requires a pseudo-factor to distinguish the treatment contrasts that are estimated in the rows.columns stratum from those estimated in the rows.columns.units stratum.

Interleaving Latin squares: these are formed similarly to the Trojan square, except that there is no longer the requirement for the k Latin squares to be orthogonal (Bailey 1988). If the squares are orthogonal, the design is a Trojan square and can be analysed by ANOVA with the help of a pseudo-factor as described above. For n=2 the design is an inflated Latin square and can be analysed by ANOVA, again with the help of a pseudo-factor. Otherwise, the design is unbalanced. It is possible to generate a balanced analysis by omitting the row.column stratum, but this is not reasonable and Yates (1935) advises against such an analysis. AGSEMILATIN can construct interleaving Latin squares for any value of n or k greater than 1.

The type of semi-Latin square can be chosen using the METHOD option with setting either Trojan, inflated, or interleaving. In a batch run the default is Trojan, while in an interactive run AGSEMILATIN will ask what type you want. AGSEMILATIN has two other options. The PRINT option can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGSEMILATIN will ask whether or not you wish to print the design. Similarly the ANALYSE option governs whether or not AGSEMILATIN produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGSEMILATIN assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run.

The information required to select the design and give identifiers to its factors can be defined using the parameters of AGSEMILATIN. The number of rows and columns of the design (n) can be defined using the parameter NROWS. Similarly, the number of units (k) for each row-column combination (that is, the number of treatments per block) can be defined by the parameter NUNITS. Parameters TREATMENTS, ROWS, COLUMNS, UNITS and PSEUDOFACTOR allow you to specify identifiers for the treatment, row, column and unit factors, and for the pseudo-factor. The SEED parameter allows you to specify a seed to randomize the design. In a batch run, this has a default of –1, to suppress randomization. If SEED is unset in an interactive run, you will be asked to provide a seed (and again a negative value will leave the design unrandomized). If one of the other parameters is unset in an interactive run, you will be asked to provide a name.

The STATEMENT parameter allows you to save a Genstat text structure containing a command to recreate the design. This is particularly useful when you are running AGSEMILATIN interactively, and specifying the information in response to questions.

Options: PRINT, METHOD, ANALYSE.

Parameters: NROWS, NUNITS, SEED, TREATMENTS, ROWS, COLUMNS, UNITS, PSEUDOFACTOR, STATEMENT.

Method

The QUESTION procedure is used to obtain the details of the required design.

Trojan squares are formed by constructing k orthogonal Latin squares with procedure AGLATIN.

For constructing an inflated Latin square, first one of the possible orthogonal Latin squares is generated with procedure AGLATIN. The generated treatment factor provides the "plot" factor. Each cell is split in k units with a corresponding "unit" treatment factor. Procedure FACPRODUCT then forms the treatment factor from the $n \times k$ combinations of the plot and unit factors.

If an interleaving Latin square is chosen which fulfils the restrictions of a Trojan square, then a Trojan square is generated because Trojan squares are optimal semi-Latin squares. When n and k do not fulfil the restrictions of a Trojan square, interleaving Latin squares are generated by first generating a Trojan square with k as large as possible. After that the generated Latin squares are duplicated (inflated) until the required interleaving Latin square is obtained. For an interleaving Latin square with n equal to 2, the $n \times k$ treatment levels are laid out from 1 to $n \times k$ in the first row, and from $n \times k$ to 1 in the second row.

The randomization is performed with BLOCKSTRUCTURE=(rows*columns)/units and, in addition, the treatment levels are permuted at random.

References

Bailey, R.A. (1988). Semi-Latin squares. *Journal of Statistical Planning and Inference*, **8**, 299-312.

Bailey, R.A. (1992). Efficient semi-Latin squares. *Statistica Sinica*, **2**, 413-437.

Edmondson, R.N. (1998). Trojan square and incomplete Trojan square designs for crop research. *Journal of Agricultural Science, Cambridge*, **131**, 135-142.

Yates, F. (1935). Complex experiments (with Discussion). *Supplement to the Journal of the Royal Statistical Society*, **2**, 181-247. [Reprinted (without Discussion) in Yates, F. (1970). *Experimental Design: Selected Papers*, 69-117. Griffin, London.

See also

Procedures: AGCROSSOVERLATIN, AGLATIN, AGQLATIN.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGSPACEFILLINGDESIGN

Generates space filling designs (B.M. Parker).

Options

PRINT = *string tokens*	Controls whether to print the design and its properties (design, properties, monitor); default * i.e. none
METHOD = *string token*	Specifies the method to use (latinhypercube, random, quasirandom); default rand
AUGMENT = *string token*	Indicates whether to augment an existing design (yes, no); default no
CENTRED = *string token*	For the Latin hypercube method, determines whether the design should be centred (yes, no); default no
CRITERION = *string token*	For the Latin hypercube method, determines which criterion should be used to assess space filling; (none, L2, maximin, entropy); default none
QRSEQUENCE = *string token*	Specifies which sequence to use with the quasi-random method; (sobol, niederreiter, faure); default sobol
NUNITS = *scalars*	Specifies the number of design points
NDIMENSIONS = *scalars*	Specifies the number of dimensions of each of the design points
NTIMES = *scalars*	Specifies the number of times to run the ESE algorithm; default 10
DISCREPANCY = *scalars*	Saves the discrepancy of the design
SEED = *scalars*	Seed to be used to randomize each design; default 0

Parameter

X = *pointer to variates*	A pointer to a set of variates, each variate representing a column of the design matrix

Description

A space filling design is an experimental design for a number of runs, which each have a number of (usually) continuous factors. They are designed to ensure that the experiment is spread over the entire design space, so that large and potentially important regions are not ignored. They are used in computer experiments, or other areas where exploring the whole design space may be useful.

Various criteria are used to assess how well the space is filled, and different methods are used to try to achieve designs that optimize them. The number of design points (i.e. the number of runs) is specified by the NUNITS parameter. Each of these runs has dimensions (i.e. number of continuous factors) specified by NDIMENSIONS.

For simplicity, AGSPACEFILLINGDESIGN produces design points such that the range for each dimension is in the interval [0,1]. These can be scaled to any interval.

The METHOD option specifies how to form the design, with the following settings.

random	is the simplest method, and selects design points uniformly at random for each dimension.
latinhypercube	aims to produce designs with good projection properties for the design on each dimension. For a two-dimensional design, the design space is divided into a regular grid of size NUNITS × NUNITS. A design is a Latin square if (and only if) there is only one design point in each row and each column. A Latin hypercube generalises this to an arbitrary

number of dimensions.

`quasirandom` Quasi-Random sequences are deterministic (i.e. non-random) space-filling sequences, which aim to fill the multidimensional design space. For designed experiments the aim is to ensure that all areas of the design space are explored. However, unlike the `random` setting, more points can readily be added to the design, using the same sequence, with the property that the augmented design is also space-filling. Three different random sequences are available; these can be selected by the `QRSEQUENCE` option, with settings `sobol`, `niederreiter` or `faure`.

If the `CENTRED` option is set to `yes`, the points of the Latin hypercube design are centred within each area of the grid. If set to `no` (default), the design points are chosen uniformly at random from any point in the grid. (Note: a centred Latin hypercube could be used to generate designs with discrete factors.)

The `AUGMENT` option allows extra runs to be added to an existing design, according to the method selected. Currently, the added runs form a separate space filling design, which is returned with the original design.

For Latin hypercube designs, the `CRITERION` option allows you to request that a stochastic algorithm is run to try to improve the space filling properties of the design, while retaining the desired Latin hypercube properties. The available settings differ according to the way in which they measure the difference between the empirical cumulative distribution function of a design and the uniform cumulative distribution function.

`L2` uses Euclidean distance,
`maximin` uses the maximum distance, and
`entropy` uses the entropy function.

This process is effective, but will not always produce the same design (except when the same `SEED` is set), and the result will not always be the best possible design within the class of Latin hypercube designs. The `NTIMES` option may be used to specify how many loops of the Enhanced Stochastic Evolutionary (ESE) algorithm are carried out, with each loop looking for a potential improvement.

The `SEED` parameter allows you to specify a seed to randomize the design. The default is 0, to continue an existing sequence of random numbers.

The `PRINT` option controls printed output, with the following settings:

`design` prints the design,
`properties` prints the space filling discrepancy criterion for the design, and
`monitor` when a criterion has been selected, this prints the best discrepancy criterion found after each ESE loop.

By default, nothing is printed.

The discrepancy can be saved, in a scalar, using the `DISCREPANCY` option.

Options: `PRINT`, `METHOD`, `AUGMENT`, `CENTRED`, `CRITERION`, `QRSEQUENCE`, `NUNITS`, `NDIMENSIONS`, `NTIMES`, `DISCREPANCY`, `SEED`.
Parameter: `X`.

Method

When the `CRITERION` option is set with a Latin hypercube design, `AGSPACEFILLINGDESIGN` uses the Enhanced Stochastic Evolutionary (ESE) algorithm described by Jin *et al*. (2005) in order to minimize the discrepancy of designs. This makes use of the `AFDISCREPANCY` procedure. The quasi-random sequences are formed using the `NAG` directive, with option

NAME=G05YAF.

Reference

Jin, R., Wei C. & Agus S. (2005). An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference,* **134**, 268-287.

See also

Directive: NAG.

Procedure: AFDISCREPANCY.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AGSQLATTICE

Generates square lattice and lattice square designs (R.W. Payne).

Options

PRINT = *string token*	Controls whether or not to print a plan of the design (design); if unset in an interactive run AGSQLATTICE will ask whether the design is to be printed, in a batch run the default is not to print the design
ANALYSE = *string token*	Controls whether or not to analyse the design, and produce a skeleton analysis-of-variance table using ANOVA (no, yes); default is to ask if this is unset in an interactive run, and not to analyse if it is unset in a batch run
DESIGNTYPE = *string token*	What type of design to form (squarelattice, latticesquare); default squa

Parameters

LEVELS = *scalars*	Number of treatments in each design
NREPLICATES = *scalars*	Number of replicates in each design, taken by default to be the maximum number available in a batch run
SEED = *scalars*	Seed for randomization; a negative value implies no randomization
TREATMENTS = *factors*	Identifier for the treatment factor for each design
PSEUDOFACTORS = *pointers*	Identifier for the pseudo-factors required if the design is not a balanced lattice
REPLICATES = *factors*	Identifier for the replicate factor for each design
BLOCKS = *factors*	Identifier for the factor to index the blocks within replicates of a square lattice
ROWS = *factors*	Identifier for the factor to index the rows within replicates of a lattice square
COLUMNS = *factors*	Identifier for the factor to index the columns within replicates of a lattice square
UNITS = *factors*	Identifier for the factor to index the units (or plots) within the blocks of a square lattice
STATEMENT = *texts*	Saves a command to recreate the design (useful if the design information has been specified in response to questions from AGSQLATTICE)
EXCLUDEREPLICATES = *scalars* or *variates*	Replicates to exclude during randomization

Description

AGSQLATTICE can form either square lattice or lattice square designs. These are designs for a single treatment factor with a number of levels that is the square of some integer *k*. The square lattice has replicates, each containing *k* blocks of *k* units (or plots), and different treatment contrasts are confounded with blocks in each replicate. The block structure of the design is thus

```
Replicates / Blocks / Units
```

Alternatively, the lattice square has a row-by-column structure, with *k* rows and *k* columns within each replicate. So the block structure is now

```
Replicates / (Rows * Columns)
```

Lattices are used, for example, in variety trials where there are many treatments to examine and

the variability of the units is such that the block size needs to be kept reasonably small. For some numbers of treatments, it is possible to generate enough different replicates so that every treatment contrast is confounded with blocks in one of the replicates of a square lattice, or with rows and with columns in one of the replicates in a lattice square. The design is then balanced. If insufficient replicates are available, or if you choose to use less than the full set available, the design is unbalanced and needs pseudo-factors for its analysis by the ANOVA directive. However, AGSQLATTICE can generate these for you automatically.

AGSQLATTICE is easiest to use interactively. It then asks questions to determine the information required to generate the design. Its options and parameters allow you to anticipate questions, or to define all the necessary information if you want to use AGSQLATTICE in batch. However, if you wish to recreate the same design later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the same information.

The DESIGNTYPE option controls whether a square lattice or a lattice square is generated. By default, if you are running Genstat in batch, a square lattice is generated. If you do not set DESIGNTYPE when running interactively, AGSQLATTICE will ask what sort of design you want.

The number of treatments can be defined using the LEVELS parameter. Similarly, the NREPLICATES parameter can define the number of replicates; by default, in a batch run, the maximum available number of replicates is formed. The SEED parameter allows you to specify a seed to be used to randomize the design. In batch the default seed is –1, to suppress randomization. If you do not set SEED when running interactively AGSQLATTICE will ask for a seed, and again a negative value suppresses any randomization. You can use the EXCLUDEREPLICATES parameter to specify a scalar or variate giving numbers of replicates that you do not wish to randomize. (This can be useful in "demonstration experiments", when the treatments may need to be kept in a systematic order in some parts of the trial, but it is not a good idea in more normal situations.)

The TREATMENTS and REPLICATES parameters allow you to specify identifiers for the treatment and replicate factors, and the PSEUDOFACTORS parameter allows you to specify a pointer to represent the pseudo-factors if these are required. The BLOCKS and UNITS parameters specify identifiers for the block-within-replicate and unit-within-block factors of a square lattice, while the ROWS and COLUMNS parameters specify identifiers for the row- and column-within-replicate factors of a lattice square. If any of these parameters is not specified in a batch run, AGSQLATTICE will use an identifier that is local within the procedure and thus lost at the end of the procedure. If you are running interactively, AGSQLATTICE will ask you to provide identifiers, and these will remain available after it has finished running.

AGSQLATTICE has a PRINT option which can be set to design to print the plan of the design. By default, if you are running Genstat in batch, the plan is not printed. If you do not set PRINT when running interactively, AGSQLATTICE will ask whether or not you wish to print the design. Similarly the ANALYSE option governs whether or not AGSQLATTICE produces a skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Again AGSQLATTICE assumes that this is not required if ANALYSE is unset in a batch run, and asks whether it is required if ANALYSE is unset in an interactive run.

Options: PRINT, ANALYSE, DESIGNTYPE.
Parameters: LEVELS, NREPLICATES, SEED, TREATMENTS, PSEUDOFACTORS, REPLICATES, BLOCKS, ROWS, COLUMNS, UNITS, STATEMENT, EXCLUDEREPLICATES.

Method

The design is formed by arranging the $k \times k$ treatments in a square array. For a square lattice, the blocks of the first replicate are formed from the rows of the array and those of the second replicate from the columns. The blocks for other replicates, if required, are formed using the treatment factors of a set of (NREPLICATES–1) mutually orthogonal k by k Latin squares

constructed using procedure AGLATIN. The rows and columns of the $k \times k$ array and the treatment factors of the mutually orthogonal Latin squares are used similarly, in pairs, to form the rows and the columns within each of the replicate of the lattice square.

See also

Procedures: AGALPHA, AGCYCLIC.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

AKAIKEHISTOGRAM

Prints histograms with improved definition of groups (A. Keen).

Options

CHANNEL = *scalar*	Channel number of output file; default is the current output file
TITLE = *text*	General title; default 'Histogram of ...', where ... is the identifier of the structure specified by DATA
LOWER = *scalar*	Lowest class limit
WIDTH = *scalar*	Interval width
SCALE = *scalar*	Number of units represented by each symbol; default 1 (or more if the page width is not sufficient)

Parameters

DATA = *identifiers*	Data for the histograms (variate, table, factor or matrix)
NOBSERVATIONS = *tables*	One-way table to save numbers in the groups
GROUPS = *factors*	Factor to save groups defined, with LEVELS the midpoints of the intervals and LABELS as LEVELS, but as text-vector
SYMBOLS = *texts*	Characters to be used to represent the bars of each histogram
DESCRIPTION = *texts*	Annotation for key

Description

The procedure AKAIKEHISTOGRAM has been designed as an alternative for the Genstat directive HISTOGRAM, for cases where the default settings are not optimal. Such cases may arise due to the following disadvantages of HISTOGRAM:

- HISTOGRAM does not take into account the round-off of the data. The round-off defines a minimal interval width, say *dy*, for the observations. A sensible interval width must be a multiple of *dy*, because otherwise the actual width is not equal for all intervals. An extreme example of this is the case where the interval width is smaller than *dy*; this causes artificial "holes" in the histogram.
- The default number of groups equals the square root of the number of observations, irrespective of the shape of the distribution. In some situations (for instance if the number of observations is large) the number of groups is unnecessarily large; in other situations (for instance if the shape of the distribution is complex) the number of groups can be too small. If the number of groups is too large, then differences in numbers of observations between neighbouring classes may be just random fluctuations, while if the number of groups is too small, valuable information is lost.
- The specification of own class limits (in a variate) can be rather cumbersome, especially if many histograms have to be produced.

AKAIKEHISTOGRAM aims to avoid these disadvantages of HISTOGRAM. By default an "optimal" number of groups is determined using Akaike's Information Criterion.

Alternatively, own class limits can be specified using options LOWER and WIDTH instead of the option LIMITS of HISTOGRAM. In a FOR loop different values for the lower limit and/or for the interval width can be specified for different quantitative structures. Scalars with missing values can be used to specify default values for these options. Option LOWER is especially important if the observations have a "natural" lower limit, for example the value 0; then 0 is taken as the lower limit of the first group and the first group has the same interval width as the following groups.

The option TITLE and the parameters of HISTOGRAM have been transferred to

AKAIKEHISTOGRAM. However, options NGROUPS and LABELS from HISTOGRAM have been omitted, because they are not in line with the style of AKAIKEHISTOGRAM.

Options: CHANNEL, TITLE, LOWER, WIDTH, SCALE.
Parameters: DATA, NOBSERVATIONS, GROUPS, SYMBOLS, DESCRIPTION.

Method
The optimality criterion used is Akaike's Information Criterion (AIC), which is twice the number of free parameters of the model (that is, the number of groups minus 1) minus the maximal log likelihood of the observations under the multinomial model. The starting histogram is a histogram with equal length intervals and more than sufficient groups. From this histogram, new histograms are derived with interval length r times the interval length of the starting histogram, $r = 2$ ... etc. The "optimal" histogram is the one with minimal AIC. The basic idea for the method is obtained from Sakamoto, Ishiguro & Kitagawa (1986); also see Taylor (1987).

The starting histogram is obtained as follows. First the range of the observations is divided into five equal length intervals from which the apparent number of observations *Na* is calculated as five times the number of observations in the interval with the largest frequency. *Na* is then used as the number of observations instead of the true number, and the number of groups *Ng* is calculated as five times the number obtained from Sturgess' formula (see, for example, Sakamoto, Ishiguro & Kitagawa (1986), page 117.):

$$Ng = 5 \times (1 + \log_{10}(Na/2))$$

The final limits of the starting histogram are obtained by a relatively strong rounding-off of the class limits (comparable with that in HISTOGRAM), where the width is always a multiple of the rounding-off interval.

Action with RESTRICT
The structures in DATA can be restricted, and in different ways; AKAIKEHISTOGRAM uses only those units that are not excluded by their respective restrictions.

References
Sakamoto, Y., Ishiguro, M & Kitagawa, G. (1986). *Akaike Information Statistics*. D. Reidel Publishing Company. Dordrecht.

Taylor, C.C., (1987). Akaike's Information Criterion and the Histogram. *Biometrika*, **74**, 636-639.

See also
Directives: DHISTOGRAM, LPHISTOGRAM.

AKEY

Generates values for treatment factors using the design key method (R.W. Payne).

Options

PRINT = *string token*	Allows the generated TREATMENTFACTOR values to be printed, tabulated by the BLOCKFACTORS (design); default * i.e. no printing
BLOCKFACTORS = *factors*	Defines the block factors for the design; default is to take those in the formula already specified by the BLOCKSTRUCTURE directive, in the order in which they occur there
KEY = *matrix*	Matrix (number of treatment factors × number of block factors) key for the design
BASEVECTOR = *variate*	Base vector (length = number of treatment factors) for the design; default is a variate of zeros
ROWPRIMES = *variate*	Prime numbers for the rows of the KEY matrix
COLPRIMES = *variate*	Prime numbers for the columns of the KEY matrix
ROWMAPPINGS = *variate*	Mappings from the rows of the KEY to the TREATMENTFACTORS
COLMAPPINGS = *variate*	Mappings from the columns of the KEY to the BLOCKFACTORS

Parameter

TREATMENTFACTORS = *factors*	Defines the treatment factors for the design; default is to take those in the formula already specified by the TREATMENTSTRUCTURE directive, in the order in which they occur there

Description

AKEY generates the values of the block factors, if necessary, in systematic order and then generates the treatment factors from the block factors using a design key. It then allows you to print the design.

The design key method, described by Patterson (1976) and Patterson & Bailey (1978), provides a very flexible way of specifying the allocation of treatments in an experimental design. The method assumes that the units are identified by a set of what are termed "plot" factors. Generally these will be the same factors that are used in the block formula. Thus, in the procedure, they are specified by an option called BLOCKFACTORS which will take the factors from the formula already set by the BLOCKSTRUCTURE directive (outside the procedure) as its default. However, if any of these factors has a non-prime number of levels, it may need to be specified instead as the combination of two or more (pseudo) factors: for example, in a block design with blocks of size eight, the plots might need to be indexed by three factors with two levels (see Example 4). The method can also be used to set up pseudo-factors for use in the treatment formula, and then the "plot" factors may be the treatment factors themselves (Example 3). If these "plot" factors do not already have values, they will be generated in "standard order" using the GENERATE directive.

The factors whose values are to be generated are specified by the TREATMENTFACTORS parameter. Again this can be omitted, and AKEY will take the factors from the existing setting of the TREATMENTSTRUCTURE directive, in the order in which they occur there.

The generated values of the factors can be printed by setting option PRINT=design. The other options define how the values are generated. The KEY option specifies a matrix known as the design key, which indicates how the values of each treatment factor are to be calculated from

the plot factors. The matrix has a row for each treatment factor and a column for each plot factor; below K_{ij} represents the element in row i and column j. (This is the transpose of the form used by Patterson 1976, but in Genstat it seems more convenient to specify the treatments by rows.) There is also an option called BASEVECTOR, which can specify a variate with an element B_i for each treatment factor to allow the levels of the factor to be shifted cyclically; by default this is a variate of zeros.

The calculation assumes that the values of the plot factors are represented by the integers zero upwards (and AKEY will perform this mapping automatically if necessary). The value $q[i]_u$ in unit u of treatment factor i is then given by

$$q[i]_u = b_i + k_{i1} \times p[1]_u + k_{i2} \times p[2]_u + \ldots + k_{in} \times p[n]_u \quad \text{modulo } t_i$$

where $p[1]_u \ldots p[n]_u$ are the values of the plot factors in unit u, and t_i is the number of levels of treatment factor i. The calculated values are integers in the range 0, 1 ... $t_i - 1$, but AKEY will again map these to the defined levels if necessary. However, all this takes place behind the scenes, within AKEY. The numbers of levels t_i must be prime numbers. They need not all be equal, but the key will usually be zero in any element where the row and column factors have different numbers of levels: that is, each treatment factor will usually be generated only from "plot" factors with the same number of levels as the treatment factor itself.

To illustrate the process, the treatments to be allocated (before randomization) to the plots of an $N \times N$ Latin Square may be calculated as

Latin-factor-value = Row-factor-value + Column-factor-value modulo N

The values of the extra factor in a Graeco-Latin square can then be formed as

Graeco-factor-value = Row-factor-value + 2 × Column-factor-value modulo N

The design key thus has rows (1,1) and (1,2); as shown in Example 1, this generates the following 5 × 5 Graeco-Latin square.

Column	0	1	2	3	4
Row					
0	0 0	1 2	2 4	3 1	4 3
1	1 1	2 3	3 0	4 2	0 4
2	2 2	3 4	4 1	0 3	1 0
3	3 3	4 0	0 2	1 4	2 1
4	4 4	0 1	1 3	2 0	3 2

If any of the block or treatment factors has a non-prime number of levels, it must be specified as the combination of two or more (pseudo) factors: for example, in a block design with blocks of size eight, the plots would need to be specified by three factors with two levels (see Example 4). Thus the COLPRIMES option allows you to supply a variate listing the prime numbers for each column of the key, and the COLMAPPINGS option then a variate to indicate the "plot" factor corresponding to each column. So, in Example 4, where we have

```
AKEY  [BLOCKFACTORS=Block,Plot; KEY=HRkey; \
      COLPRIME=!(4(2)); COLMAP=!(1,2,2,2)]
```

COLPRIME specifies that the prime for each column is 2, COLMAP specifies that the first column corresponds to the first "plot" factor (Block in the example) and that columns 2-4 correspond to the second "plot" factor (Plot in the example). The default for COLMAP is a variate containing the integers 1 up to the number of "plot" factors, so it can be omitted if no pseudo-factors are required. If COLPRIME is omitted, the primes for the columns are provided by the numbers of levels of the "plot" factors, as already explained. Options ROWPRIME and ROWMAP similarly allow you to specify pseudo-factors to generate the treatment factors.

The design key thus provides a very convenient way of defining treatment factors. Patterson & Bailey (1978) show a range of examples of keys, which are used to form the worked examples below. Essentially, the key identifies each factor i with the set of contrasts (in the usual terminology)

P[1]**K_{i1} P[2]**K_{i2} ... P[n]**K_{in}

and the skill when forming a design is in selecting the best set for each factor. The Genstat design system has a repertoire of keys, and these are used by procedures DESIGN and AGDESIGN to generate a range of designs, including factorials, fractional factorials, Latin squares and Lattices.

Options: PRINT, BLOCKFACTORS, KEY, BASEVECTOR, ROWPRIMES, COLPRIMES, ROWMAPPINGS, COLMAPPINGS.
Parameter: TREATMENTFACTORS.

Method
The FCLASSIFICATION and FORMULA directives are used, if necessary, to form lists of factors from the block or treatment formulae. The factor levels are then generated using the standard Genstat facilities for calculations and manipulation.

Action with RESTRICT
If any of the factors is restricted, only the part of the design not excluded by the restriction will be generated.

References
Patterson, H.D. (1976). Generation of factorial designs. *Journal of the Royal Statistical Society Series B*, **38**, 175-179.
Patterson, H.D. & Bailey, R.A. (1978). Design keys for factorial experiments. *Applied Statistics*, **27**, 335-343.

See also
Directives: AFMINABERRATION, GENERATE, FKEY, FPSEUDOFACTORS.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

ALIAS

Finds out information about aliased model terms in analysis of variance (R.W. Payne).

Options

TREATMENTSTRUCTURE = *formula*	Treatment model for the design
BLOCKSTRUCTURE = *formula*	Block model for the design
FACTORIAL = *scalar*	Value used in the FACTORIAL option of ANOVA if not the default
DESIGN = *pointer*	Design structure for the analysis

Parameter

TERM = *factors*	Factors defining the aliased model term

Description

When a term is aliased in an analysis of variance, it is listed in the Information summary (produced by ANOVA [PRINT=information]) under the heading "Aliased model terms" (see the *Guide to the Genstat Command Language*, Part 2, Section 4.7.1). However ANOVA does not indicate the terms with which it is aliased. This information can be obtained using procedure ALIAS.

The aliased term is specified by setting the TERM parameter to the list of factors that define it. The structure of the design can be specified either by options BLOCKSTRUCTURE and TREATMENTSTRUCTURE (together with option FACTORIAL, if necessary); alternatively you can save the design structure from the original analysis and supply this using the DESIGN option – this is the only way of specifying the design if there are weights or if the analysis is restricted. If an undeclared structure is specified for DESIGN (and BLOCKSTRUCTURE and TREATMENTSTRUCTURE are also specified), it will be set to the design structure for the analysis.

Note: this procedure has been replaced by FALIASTERMS, but is retained for use in earlier programs. FALIASTERMS has a more convenient syntax, and allows you to save details of the aliased terms.

Options: TREATMENTSTRUCTURE, BLOCKSTRUCTURE, FACTORIAL, DESIGN.
Parameter: TERM.

Method

The procedure calculates a set of dummy effects for the aliased model term, and then forms and analyses a variate in which only these effects are present. The analysis detects the model terms to which the term is aliased as those that have non-zero sums of squares.

Action with RESTRICT

None of the options nor parameters are vectors. To indicate that the analysis is of a restricted set of units you must use the DESIGN option to specify the design structure from the original analysis.

See also

Directives: ANOVA, FPSEUDOFACTORS.
Procedure: AEFFICIENCY, FALIASTERMS.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, Design of experiments.

ALIGNCURVE

Forms an optimal warping to align an observed series of observations with a standard series (D.B. Baird).

Options

PRINT = *string tokens*	What to print (criterion, ss, warps); default * i.e. nothing
PLOT = *string tokens*	What to plot (series, warping); default * i.e. no plots
WARPPENALTY = *scalar*	The relative penalty to add to the criterion when jumping a unit in one series but not the other; default 1
MAXSTEP = *scalar*	The largest jump that can be made between the two series at a single point; default 1
MAXDIFFERENCE = *scalar*	Sets a limit on size of difference between the series to be squared and added to the criterion (differences greater than this are truncated to MAXDIFFERENCE, thus allowing the effects of outliers to be down-weighted); default * i.e. no limit
USEMEANS = *string token*	Whether to use the means of points covered in one step, rather than the final value, when calculating the sums of squares between the two series (yes, no); default no
FORCEENDALIGNMENT = *string token*	Whether to force the ends of the two series to align, so that warping happens only in the middle of the series (yes, no); default no
WINDOW = *scalar*	Window number for the plots; default 1
KEYWINDOW = *scalar*	Window for the key (zero for no key); default 2

Parameters

Y = *variates*	Series to be aligned with the standard series
STANDARD = *variates*	Standard series for each Y
WEIGHTS = *variates*	Weights for the contribution of each point to the criterion; default * no weighting
UWARP = *variates*	The warped positions of the unit numbers, required to align Y with STANDARD
YWARP = *variates*	The warped series for Y, i.e. the optimally aligned y-values
CRITERIONVALUE = *scalars*	The criterion value (as optimized during the alignment)
TITLE = *text*	Title for the plots

Description

ALIGNCURVE is useful when you have a series of observations of a theoretical curve in which the x-axis may have been distorted by stretching or compression. The aim is to "warp" the units of the curve so that it optimally matches a standard series. (This can be used, for example, to align electrophoresis gels.) The observed and standard series are specified, in variates, by the Y and STANDARD parameters respectively.

The warping takes the form of one of the series jumping values, so it that makes several steps for a single step of the other series. There is no limitation on the number of times that this can happen along the series, but you can set option FORCEENDALIGNMENT=yes, to require the two series to match at their final values.

The dynamic warping algorithm that is used, seeks to optimize a criterion that is based on the sum of squares of residuals between the warped observed series and the standard series at each

point, plus a penalty that depends on the amount of warping.

The WEIGHTS parameter can supply a variate of weights to use for the contribution of each point to the criterion. If the standard curve has non-constant variance around it, typically you would set the weights to be the reciprocal of the variance or equivalently, one over the square of the standard deviation. Suppose, for example, if the curve is the mean of a Poisson process. the mean is equal to the variance for a Poisson distribution, so the you could use weights of 1/STANDARD to allow for the non-constant variance. Alternatively, if the points on the curves were means of differing numbers of replicated observations, you could weight by the recriprocals of the replications. By default there is no weighting.

The MAXDIFFERENCE option sets a limit on contribution of each residual: differences greater than this are truncated (to MAXDIFFERENCE), thus allowing the effects of outliers to be down-weighted. By default there is no limit.

The WARPPENALTY option defines the relative penalty to be added to the criterion when the series are warped by one step in either direction; default 1. The penalty is given by WARPPENALTY multiplied by a scale factor equal to the average mean square of the initial (unwarped) differences between Y and STANDARD. (It is thus scale independent, so multiplying both Y and STANDARD by a constant will still give the same solution.) Setting WARPPENALTY to zero finds the optimally aligned series with no constraint on warping. Conversely, setting WARPPENALTY to a large value discourages warps, as a jump will be taken only if the decrease in the sum of squares is greater than the penalty.

The MAXSTEP option sets a limit on the number of units that can be jumped at any one point (default 1).

By default, the y-value after a jump is taken to be the value at the end of the jump, but you can set option USEMEANS=yes to use the mean of the values at the end and intervening (jumped) points.

The UWARP parameter can save a variate containing the warped units, i.e. the unit of Y that corresponds (after warping) to each unit of STANDARD. The YWARP parameter can save the y-values of the optimally aligned series, and the CRITERIONVALUE parameter can save the optimal criterion value.

The PRINT option controls printed output, with settings:

criterion	to print the smoothing weight, maximum step size and optimal criterion value,
warps	to print the location of the warps,
ss	to print sums of squares before and after alignment.

By default nothing is printed.

The PLOT option controls the plots that are displayed, with settings:

series	plots the aligned and standard series,
warping	plots the steps in the warping that have been used to align the two series.

By default nothing is plotted. The WINDOW option specifies the window to use for the plots (default 1), and the KEYWINDOW option specifies the window for their keys (default 2). You can supply a title for the plots using the TITLE parameter.

Options: PRINT, PLOT, WARPPENALTY, MAXSTEP, MAXDIFFERENCE, USEMEANS, FORCEENDALIGNMENT, WINDOW, KEYWINDOW.

Parameters: Y, STANDARD, WEIGHTS, UWARP, YWARP, CRITERIONVALUE, TITLE.

Action with RESTRICT

Any restrictions are ignored.

See also
Procedures: BASELINE, PEAKFINDER.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

ALLDIFFERENCES

Shows all pairwise differences of values in a variate or table (A.R.G. McLachlan).

Options

PRINT = *string token*	What to print (differences); default diff
CLPRINT = *string token*	How to print column labels (labels, integers); default labe
SORT = *string token*	How to sort the DATA values (ascending, descending); default * i.e. not sorted
MVREMOVE = *string token*	Whether to remove missing values (yes, no); default no
RCMETHOD = *string token*	Which differences to calculate i.e. column-row, row-column, or absolute values (column, row, absolute); default colu
DIAGONAL = *string token*	Whether to put the data values into the diagonal of the symmetric matrices of results (values); default * i.e. diagonal left as missing values

Parameters

DATA = *variates* or *tables*	Data values whose pairwise differences are required
DIFFERENCES = *symmetric matrices* or *pointers*	Saves the pairwise differences in a symmetric matrix if GROUPS is unset, otherwise in a pointer to several symmetric matrices
GROUPS = *factors* or *pointers*	Defines groupings of the data values
LABELS = *texts*	Labels for the rows (and columns) of the symmetric matrices of differences
NEWLABELS = *texts* or *pointers*	Saves the row labels of the symmetrix matrices of differences in a text if GROUPS is unset, otherwise in a pointer to several texts

Description

ALLDIFFERENCES prints out a symmetric matrix of all pairwise differences between values in a variate or table. That is, every value is subtracted once from every other value and the results of these subtractions are arranged in a symmetric matrix.

The DATA parameter supplies the data values in either a variate or a table. If a DATA table has margins, these are ignored and the marginal values not used in the differences calculations. If DATA is set to a variate, this must have at least two unrestricted values for differences to be calculated.

The data can be subdivided into groups by using the GROUPS parameter. This can be set to a single factor or to a pointer containing several factors. When it is a pointer, groups are formed for each combination of the factor levels. Each factor must either be of the same length as the DATA variate, or be one of the factors classifying a DATA table. If GROUPS is specified, then at least one group must have two or more unrestricted values in it.

Labels for the rows (and columns) of the symmetric matrix of differences can be provided, using the LABELS parameter, by supplying a text with a value for each DATA value. The unrestricted number of labels must be the same as the number of unrestricted data values. If LABELS are not supplied for a DATA variate with *n* values, the integers from 1 to *n* are used for labels. If LABELS are not supplied for a DATA table, labels are created from the table factors using labels if factor labels are present, or levels if a factor does not have labels. The labels that are actually used for the rows of the symmetric matrices of differences can be obtained from the NEWLABELS parameter which will either be a text if GROUPS is not set, or a pointer to texts if

GROUPS is specified.

The pairwise differences can be saved using the using DIFFERENCES parameter. If there are no groups, they are saved in a symmetric matrix. Alternatively, if there are groups, they are saved in pointer with a symmetric matrix for each group. The suffixes of the pointer are the ordinal levels of a single GROUPS factor. For multiple GROUPS factors they are the integers 1...*n*, where *n* is the number of factor combinations. The saved symmetric matrices each have an extra text defined that gives details of the contents. This text can be seen by setting option IPRINT=extra when printing the matrices using the PRINT directive.

The differences are printed by default, but you can set option PRINT=* to suppress this if you just want to store the differences for further calculation or later printing. The format of the printed column labels can be controlled using the CLPRINT option. The default, CLPRINT=labels, prints both row labels and column labels i.e. it is equivalent to using the PRINT directive with options RLPRINT=labels and CLPRINT=labels. The alternative setting CLPRINT=integers is useful when printing results that have long labels. The columns are then labelled with integers instead of text labels, and the rows are labelled with both text and integers (where the column integers match those of the rows). This is equivalent to using PRINT with options RLPRINT=labels,integers and CLPRINT=integers. At the same time, ALLDIFFERENCES also changes the field width so that it just accommodates the widest value. Usually, this means that the columns are printed closer together, so that the output will be much more compact. If further control is needed over the printing of the results, it is suggested that you save the differences, and then use PRINT with your own preferred settings.

The DATA values can be sorted into either ascending or descending order by specifying the SORT option. (Note though, that any labels supplied by the LABELS parameter must be in the original unsorted order – these will be sorted automatically by ALLDIFFERENCES together with the data values.) By default, the DATA values are not sorted.

By default, when missing values are present in the DATA, these will create missing values in the symmetric matrix of differences. If groups have been specified, then any group whose differences are all missing will be omitted from the printed output, although its symmetric matrix (of missing values) will still be saved by the DIFFERENCES parameter. Alternatively, you can remove the missing values by setting option MVREMOVE=yes. Groups with only missing differences are then neither printed nor saved.

The order of the subtraction in the symmetric matrix of results is controlled by the RCMETHOD option. The default, column, calculates the difference as

difference = column value – row value

but this can be reversed to give

difference = row value – column value

by setting RCMETHOD=row. Essentially, the choice of RCMETHOD determines the sign of the differences. If instead you wish all of the differences to be positive values, you can use RCMETHOD=absolute. This is equivalent to calculating the differences by either method, and then taking their absolute values.

By default, the diagonal of the symmetric matrix of differences will contain missing values. Alternatively, you can replace these by the row values (which are also the column values) by setting option DIAGONAL=value.

Options: PRINT, CLPRINT, SORT, MVREMOVE, RCMETHOD, DIAGONAL.
Parameters: DATA, DIFFERENCES, GROUPS, LABELS, NEWLABELS.

Method

Each value in DATA is subtracted from every other value and the result stored in a symmetric matrix. If restrictions are applied, or MVREMOVE=yes, then procedure SUBSET is first used to remove any restricted values or missing values.

Action with RESTRICT
Restrictions are honoured but are relevant only when the data values are in a variate. In this case, any restrictions on the DATA variate, the GROUP factor and the LABELS text are all combined and honoured. Thus, you can exclude some data values not just by restricting the DATA variate, but also by restricting the GROUPS factor or the LABELS text, or both. Only unrestricted values are used in the differences calculations. Since restrictions are not possible on a table, when the DATA are a table, any restrictions on the LABELS text and the GROUPS factor are then ignored.

See also
Procedures: PAIRTEST, RPAIR.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

ALLPAIRWISE

Performs a range of all pairwise multiple comparison tests (D.M. Smith).

Options

METHOD = *string token*	Test to be performed (Tukey, SNK, REGWMR, Duncan, Scheffe, FPLSD, FULSD, Bonferroni, Sidak); default *
DIRECTION = *string token*	How to sort means (ascending, descending); default asce
PROBABILITY = *scalar*	The required significance level; default=0.05
ALSD = *string token*	Whether to use the alternative LSD test where the Studentized Range statistic is used instead of Student's t (yes, no); default no

Parameters

MEANS = *variates* or *tables*	Mean values
REPLICATIONS = *scalars* or *tables* or *variates*	Number(s) of observations per mean
VARIANCE = *scalars*	Estimate of variance
DF = *scalars*	Degrees of freedom
LABELS = *texts*	Identifiers of mean values

Description

ALLPAIRWISE performs a range of all pairwise multiple comparison tests (see Hsu 1996 and Bechhofer, Santner & Goldsman 1995). The methodology implemented in the procedure closely follows that described in Chapter 5 of Hsu (1996).

The means are input using the MEANS parameter, either in a table saved e.g. from AKEEP, or in a variate. The replication (or number of observations in each mean) is supplied by the REPLICATIONS parameter, either in a scalar (if all the replications are equal) or in a structure of the same type as the means. The estimate of the variance (usually a pooled estimate as given by the residual mean square in ANOVA, and accessible using the VARIANCE parameter of AKEEP) and its corresponding degrees of freedom are input as scalars using the VARIANCE and DF parameters respectively. The LABELS parameter can be used to supply labels for the means.

The type of test to be performed is specified by the METHOD option, with settings Tukey, SNK (Student-Newman-Keuls), REGWMR (Ryan/Einot-Gabriel/Welsch multiple range test), Duncan, Scheffe, FPLSD (Fisher's Protected Least Significant Difference), FULSD (Fisher's Unprotected Least Significant Difference), Bonferroni and Sidak.

The DIRECTION option allows the means to be arranged in ascending or descending order. The PROBABILITY option allows the pair-wise significance level for the intervals from the Fisher tests to be changed from the default 0.05 (e.g. to 0.01). For the other tests, it changes the experiment-wise significance level. The ALSD allows the LSD test asked for (FPLSD or FULSD) to use the Studentized Range statistic rather than Student's t (for further information see Hsu, 1996, page 139).

Options: METHOD, DIRECTION, PROBABILITY, ALSD.
Parameter: MEANS, REPLICATION, VARIANCE, DF, LABELS.

Method

The methodology implemented is based on that described and reviewed in Hsu (1996), and Bechhofer, Santner & Goldsman (1995). For specific details of the tests these books should be referred to.

References

Bechhofer, R.E., Santner, T.J. & Goldsman, D.M. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. Wiley, New York.

Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also

Procedures: AMCOMPARISON, AUMCOMPARISON, AMDUNNETT, CONFIDENCE, MCOMPARISON, VMCOMPARISON.

Genstat Reference Manual 1 Summary section on: Analysis of variance.

AMCOMPARISON

Performs pairwise multiple comparison tests for ANOVA means (D.M. Smith).

Options

PRINT = *string tokens*	Controls printed output (comparisons, critical, description, lines, letters, plot, mplot, pplot); default lett
METHOD = *string token*	Test to be performed (tukey, snk, regwmr, duncan, scheffe, fplsd, fulsd, bonferroni, sidak); default fpls
FACTORIAL = *scalar*	Limit on the number of factors in each term; default 3
DIRECTION = *string token*	How to sort means (ascending, descending); default asce
PROBABILITY = *scalar*	The required significance level; default 0.05
STUDENTIZE = *string token*	Whether to use the alternative LSD test where the Studentized Range statistic is used instead of Student's t (yes, no); default no
SAVE = *ANOVA save structure*	Save structure to provide the tables of means and associated information; default uses the save structure from the most recent ANOVA

Parameters

TERMS = *formula*	Treatment terms whose means are to be compared
MEANS = *pointer* or *variate*	Saves the (sorted) means
DIFFERENCES = *pointer* or *symmetric matrix*	Saves differences between the (sorted) means
LABELS = *pointer* or *text*	Saves labels for the (sorted) means
LETTERS = *pointer* or *text*	Saves letters indicating groups of means that do not differ significantly
SIGNIFICANCE = *pointer* or *symmetric matrix*	Indicators to show significant comparisons between (sorted) means
CIWIDTH = *pointer* or *symmetric matrix*	Saves the width of the confidence interval for the absolute differences between the (sorted) means

Description

AMCOMPARISON performs a range of all pairwise multiple comparison tests (see Hsu 1996 and Bechhofer, Santner & Goldsman 1995). The methodology implemented in the procedure closely follows that described in Chapter 5 of Hsu (1996).

The TERMS parameter specifies a model formula to define the treatment terms whose means are to be compared. The means (and the necessary associated information) are usually taken from the most recent analysis of variance (performed by ANOVA), but you can set the SAVE option to a save structure from another ANOVA if you want to examine means from an earlier analysis. As in ANOVA, the FACTORIAL option sets a limit on the number of factors in each term (default 3).

Printed output is controlled by the PRINT option, with settings:

comparisons	prints the differences between the pair of means, upper and lower confidence limits for the differences, t-statistics and an indication of whether or not they are significant;
critical	gives critical values for the t-statistic for situations where these do not vary amongst the comparisons (i.e. for the

	Scheffe, Bonferroni and Sidak methods, as well as the Fisher LSD methods provided all the comparisons have the same number of residual degrees of freedom);
`description`	provides a description including information such as the experiment-wise and compartment-wise error rates;
`lines`	gives the means, with lines joining those that do not differ significantly;
`letters`	gives the means, with identical letters (a, b etc.) alongside those that do not differ significantly;
`mplot`	does a mean-mean scatter plot (synonym `plot`);
`pplot`	displays the probabilities in a shade plot.

By default, `PRINT=letters`.

The means are usually sorted into ascending order, but you can set option `DIRECTION=descending` for descending order, or `DIRECTION=*` to leave them in their original order. Note, though, that the lines joining means with non-significant differences may then be broken.

If the standard errors for the differences between the means are unequal (as will happen, for example, if the means have unequal replication), the memberships of the groups defined by the lines or letters may be inconsistent. Suppose, for example, you have ordered means A, B and C. If the s.e.d. for A vs. C is large compared to those for A vs. B and B vs C, you might find that there is no significant difference between A and C, but there are significant differences between A and B, and between B and C. So treatments A and B and treatments B and C would be in different groups. However, treatments A and C (which are further apart) would be in the same group. This contradicts the idea behind multiple comparisons, where you expect that if two means are in the same group, than any mean between them should be in that group too. If `AMCOMPARISON` finds inconsistencies like this, it gives a diagnostic and suppresses the printing of lines and letters (but not the other types of output).

The mean-mean scatter plot allows you to assess the confidence region for the difference between each pair of means visually. It has grid lines from both the x- and y-axis at the position of each mean, and a diagonal line at 45 degrees marking y=x. The confidence interval for each pair of means is plotted as a line at an angle of −45 degrees and centred on the intersection above the line y=x of the grid lines for the two means (so the y grid line is for the larger of the two means, and the x grid line is for the smaller mean). The difference between the means is significant if their confidence line does not intersect the line y=x. For more details, see Hsu (1996) pages 151-153.

The shade plot displays the probabilities in a symmetric matrix. The colour of each cell represents the probability for the difference between the means for the treatments in the corresponding row and column.

The type of test to be performed is specified by the `METHOD` option, with settings `Tukey`, `SNK` (Student-Newman-Keuls), `REGWMR` (Ryan/Einot-Gabriel/Welsch multiple range test), `Duncan`, `Scheffe`, `FPLSD` (Fisher's Protected Least Significant Difference), `FULSD` (Fisher's Unprotected Least Significant Difference), `Bonferroni` and `Sidak`. The `PROBABILITY` option allows the pair-wise significance level for the intervals from the Fisher tests to be changed from the default 0.05 (e.g. to 0.01). For the other tests, it changes the experiment-wise significance level. The `STUDENTIZE` option can specify that the Fisher's protected or unprotected LSD tests should use the Studentized Range statistic rather than Student's t (for further information see Hsu 1996, page 139).

The `MEANS` parameter can save the means, sorted according to the `DIRECTION` option and omitting any that were non-estimable. If the `TERMS` parameter specifies a single term, `MEANS` should be set to a variate. If `TERMS` specifies several terms, you must supply a pointer which will then be set up to contain as many variates as there are terms. Similarly the `LABELS` parameter

can save labels to identify the means, in either a text (for a single term) or in a pointer of texts (for several). Likewise the LETTERS parameter can save texts with the letters identifying means that do not differ significantly, and the SIGNIFICANCE parameter can save symmetric matrices containing ones or zeros according to whether the various comparisons were significant or non-significant. The DIFFERENCES parameter can save symmetric matrices containing the differences between the (sorted) means, and the CIWIDTH parameter can save symmetric matrices containing the widths of the confidence intervals for the differences.

Options: PRINT, METHOD, FACTORIAL, DIRECTION, PROBABILITY, STUDENTIZE, SAVE.
Parameter: TERMS, MEANS, DIFFERENCES, LABELS, LETTERS, SIGNIFICANCE, CIWIDTH.

Method
The methodology implemented is based on that described and reviewed in Hsu (1996), and Bechhofer, Santner & Goldsman (1995). For specific details of the tests these books should be referred to.

References
Bechhofer, R.E., Santner, T.J. & Goldsman, D.M. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. Wiley, New York.
Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also
Directive: ANOVA.
Procedures: AUMCOMPARISON, AMDUNNETT, CONFIDENCE, STEEL, VMCOMPARISON.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AMDUNNETT

Forms Dunnett's simultaneous confidence interval around a control (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (interval); default inte
METHOD = *string token*	Form of the alternative hypothesis (twosided, greaterthan, lessthan); default twos
CIPROBABILITY = *scalar*	Probability level for the confidence interval; default 0.95, i.e. a 95% confidence interval
LOWER = *scalar*	Saves the lower confidence limit
UPPER = *scalar*	Saves the upper confidence limit
SAVE = *ANOVA save structure*	Save structure to provide the means; default uses the save structure from the most recent ANOVA

Parameters

FACTOR = *factors*	Define the model term whose means are to be compared
CONTROL = *scalars* or *texts*	Scalar or single-valued text for each factor to identify which of the means of the term is the control; default uses the reference level of the FACTOR

Description

AMDUNNETT is useful when you want to compare several treatments with a control treatment, and use a critical value that controls the chance that any one comparison may be found significant when there are no true differences. (It is designed thus to take account of the fact that you are making multiple comparisons with the control.)

The FACTOR parameter lists the factors that define the treatment term whose means are to be compared. The means are usually taken from the most recent analysis of variance (performed by ANOVA), but you can set the SAVE option to a save structure from another ANOVA if you want to examine means from an earlier analysis. The CONTROL parameter specifies a list of scalars to identify the levels of the factors that correspond to the control, or you can use a string (or single-valued text) to identify the level of any factor that has labels. If CONTROL is unset, AMDUNNETT uses the reference level of the FACTOR.

The METHOD option defines the type of interval that is formed. By default AMDUNNETT forms a two-sided interval. If you set METHOD=lowerthan, a lower confidence interval is formed to assess the one-sided test of the null hypothesis that the treatment means are not lower than the control mean. Alternatively, you can set METHOD=greaterthan, to obtain an upper confidence interval to assess the one-sided test of the null hypothesis that the treatment means are not greater than the mean of the control.

The probability for the confidence interval is specified by the CIPROBABILITY option; the default 0.95 gives a 95% interval. The lower and upper values of the interval can be saved (in scalars) using the LOWER and UPPER options, respectively. By default the interval is printed, but this can be suppressed by setting option PRINT=*.

Options: PRINT, METHOD, CIPROBABILITY, LOWER, UPPER, SAVE.
Parameters: FACTOR, CONTROL.

Method

AMDUNNETT uses the methods of Dunnett (1955, 1989); also see Hsu (1996) Chapter 3.

Action with RESTRICT

If the Y variate in the original ANOVA was restricted, only the units not excluded by the restriction will have been analysed.

References

Dunnett, C.W. (1955). A multiple comparison procedure for comparing several treatments with a control. *Journal of the Americal Statistical Association*, **50**, 1096-1121.

Dunnett, C.W. (1989). Algorithm AS251 Multivariate normal probability intervals with product correlation structure. *Applied Statistics*, **38**, 564-579.

Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also

Procedures: AMCOMPARISON, AUMCOMPARISON, EDDUNNETT, CONFIDENCE, STEEL, VMCOMPARISON.

Genstat Reference Manual 1 Summary section on: Analysis of variance.

AMERGE

Merges extra units into an experimental design (R.W. Payne).

Option

SORT = *string token*	Whether to sort the factors afterwards (no, yes); default no

Parameters

FACTOR = *factors*	Factors to which the new units are to be added
NEWUNITS = *factors*, *variates* or *scalars*	Extra units to be added to each factor

Description

AMERGE provides a convenient way of adding extra units into an experimental design. In the simplest case, this can be used to add control treatments to an already generated factorial design. More complicated uses may join together two completely different designs, for example a randomized block design to a balanced incomplete block design. These are both illustrated in the example.

The factors of the design which is to be augmented are specified using the FACTOR parameter, and the units that are to be added to each one are specified by the NEWUNITS parameter. The same number of units must be added to every FACTOR, and their levels (and labels) will be extended, if necessary, according to those defined on the units that are added. New units of a factor that are to receive different levels should be specified in a factor or a variate. Alternatively, if every new unit is to receive the same level of the FACTOR, NEWUNIT can be set to a scalar.

The SORT option can be set to yes to request that the FACTOR values are sorted after the new units have been added. Otherwise, they are simply placed at the end of the existing values.

Option: SORT. Parameters: FACTOR, NEWUNITS.

Method

AMERGE uses the standard Genstat manipulation facilities.

Action with RESTRICT

Any restrictions on the vectors are ignored.

See also

Procedures: APPEND, APRODUCT.

Genstat Reference Manual 1 Summary sections on: Design of experiments, Calculations and manipulation.

AMMI

Allows exploratory analysis of genotype × environment interactions (M. Talbot, K. Brown & M.F. Smith).

Options

PRINT = *string tokens*	Results to be output (aovtable, genotype, environment, estimates, envtable, cluster, stability); default * i.e. none
NROOTS = *scalar*	Number of IPCA scores required; default is to take as many roots as possible up to a maximum of 9
DIMENSIONS = *scalars*	Two numbers specifying the dimensions to display in the biplots; default 1,2
PLOT = *string tokens*	Types of biplot to display (mean, ipca); default * i.e. none
SCALING = *string token*	Scaling to use for the ipca biplot (genotype, environment, symmetric); default envi

Parameters

DATA = *variates* or *tables*	Provides the data to be analysed
GENOTYPES = *factors*	Specifies the genotypes
ENVIRONMENTS = *factors*	Specifies the environments
REPLICATES = *factors*	Replication factor; this should be omitted if the data comprises just the genotype by environment means
GSCORES = *pointers*	Pointer containing a set of variates (each of length equal to the number of genotypes) to save the genotype IPCA scores
ESCORES = *pointers*	Pointer to a set of variates to save the environment IPCA scores
RESIDUALS = *variates*	Saves the residuals from the AMMI model
FITTEDVALUES = *variates*	Saves the fitted values from the AMMI model
TITLEPREFIX = *texts*	Specifies a prefix to use for the titles of the plots
AOVTABLE = *pointers*	Saves the analysis-of-variance table
STABILITY = *variates*	Saves the AMMI stability values

Description

AMMI is a procedure for fitting, to data classified by two factors, a model which involves the Additive Main effects of ANOVA along with the Multiplicative Interaction effects of principal components analysis (PCA). The method is used when analysing data from a series of trials with crop genotypes.

A principal components model is fitted to the residuals from the ANOVA and the resulting scores, called the I (for interaction) PCA are calculated for both the genotypes and the trials or environments.

The data to be analysed can be supplied in a variate using the DATA parameter. The associated genotype and environment factors are specified using the GENOTYPES and ENVIRONMENTS parameters, respectively. You can also use the REPLICATES parameter to specify a factor defining replicates within environments. When constructing the analysis-of-variance table, AMMI assumes that these replicates arise from the use of a randomized block design within each environment. There must be equal replication. If you have a more complicated structure, you can form the genotype × environment means (for example using ANOVA and AKEEP. or REML and VKEEP), and supply this instead. If the GENOTYPES and ENVIRONMENTS are not specified as well as the table, it is assumed that the rows of the table correspond to the genotypes, and the columns

correspond to the environments.

The NROOTS option allows the number of roots (sets of scores) for the principal component analysis to be specified.

The PRINT option allows a choice of results to be requested by settings:

aovtable	analysis-of-variance table summarising the contribution of each component to the interaction term,
genotype	genotype means and scores and stability,
environment	environment means and scores,
envtable	table of environment means and variances,
estimates	genotype estimates for each environment,
cluster	hierarchical clustering of AMMI genotype estimates over environments (using the average link method and Euclidean test for the similarity matrix),
stability	AMMI stability values (Purchase, Hatting & van Deventer 2000).

The PLOT option controls the biplots that are displayed. The setting mean produces a biplot of the genotype and environment means against their corresponding IPCA scores. The setting ipca produces a biplot of the IPCA scores.

The scaling used for the ipca biplot is controlled by the SCALING option. The settings environment and genotype multiply the environment or genotype scores, respectively, by their corresponding eigenvalues. The symmetric multiplies both the environment and the genotype scores by the square roots of their corresponding eigenvalues.

By default, the plots are produced using the first two dimensions of IPCA scores, but you can specify other dimensions using the DIMENSIONS option.

The default titles for the plots are prefixed using the identifier of the DATA variate or table. However, you can supply an alternative prefix using the TITLEPREFIX parameter.

The genotype and environment IPCA scores can be saved within a pointer to a set of variates, using the GSCORES and ESCORES parameters respectively. The fitted values for the AMMI model can be saved using the FITTEDVALUES parameter, and the simple residuals can be saved using the RESIDUALS parameter.

The AOVTABLE parameter saves the analysis-of-variance table, in a pointer with elements labelled 'Source', 'd.f.', 's.s.', 'm.s.', 'v.r.' and 'F pr'.

The STABILITY parameter saves the AMMI stability values, defined as

$$\sqrt{\{ ((\text{IPCA}_1 \text{ scores}) \times (\text{IPCA}_1 \text{ s.s.}) / (\text{IPCA}_2 \text{ s.s.}))^2 + (\text{IPCA}_2 \text{ scores})^2 \}}$$

by Purchase, Hatting & van Deventer (2000). These are the distances of the genotypes from zero in the 2-dimensional plot of genotype scores, but with an additional weighting for the IPCA_1 scores to take account of their larger contribution to the genotype-by-environment interaction.

Options: PRINT, NROOTS, DIMENSIONS, PLOT, SCALING.

Parameters: DATA, GENOTYPES, ENVIRONMENTS, REPLICATES, GSCORES, ESCORES, RESIDUALS, FITTEDVALUES, TITLEPREFIX, AOVTABLE, STABILITY.

Method

The data are averaged over replicates, and the genotype by environment means are calculated. ANOVA is used to provide the main effects, sums of squares and degrees of freedom. The matrix of residuals from ANOVA are then decomposed by singular value decomposition to generate the AMMI analysis (see, for example, Gauch 1992).

Action with RESTRICT

If the DATA variate is restricted the analysis will involve only the units not excluded by the restriction.

References

Gauch, H.G. (1992). *Statistical Analysis of Regional Yield Trials – AMMI analysis of factorial designs*. Elsevier, Amsterdam.

Purchase, J.L., Hatting, H. & van Deventer, C.S. (2000). Genotype × environment interaction of winter wheat (*Triticum aestivum* L.) in South Africa: II Stability analysis of yield performance. *S. Afr. J. Plant Soil*, **17**, 101-107.

See also

Procedures: GESTABILITY, GGEBIPLOT, RFINLAYWILKINSON, DBIPLOT.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

AMTDISPLAY

Displays further output for multitiered experiments analysed by AMTIER (C.J. Brien & R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, aovpseudotable, design, effects, fittedvalues); default * i.e. none

Parameter

SAVE = *pointers*	Save structure for each analysis (saved from AMTIER); if this is not set the output is from the most recent AMTIER analysis

Description

The AMTIER procedure analyses data from designs that require up to three model formulae to specify their analysis (resulting from three or more tiers for the experiment). Information from AMTIER can be saved by the SAVE parameter, and input to AMTDISPLAY using its own SAVE parameter. Alternatively, if SAVE is not set, AMTDISPLAY will display output from the most recent analysis by AMTIER.

The output is controlled by the PRINT option with settings:

aovtable	to print the analysis-of-variance table,
aovpseudotable	to print the analysis-of-variance table with lines for all the pseudo-terms (generated by pseudo-factors) given explicitly,
design	to display the structure of the design,
effects	to print tables of effects and residuals, and
fittedvalues	to print a table with the y-variate, fitted valued and residuals.

Option: PRINT.
Parameter: SAVE.

Method

Multitiered experiments are defined by Brien (1983), their design is discussed by Brien & Bailey (2006) and Brien *et al*. (2011), and their analysis of variance is described by Brien & Payne (1999), Brien & Bailey (2009) and Bailey & Brien (2013).

Action with RESTRICT

There must not be any restrictions.

References

Bailey, R.A. & Brien C.J. (2013). Randomization-based models for multitiered experiments. I. A chain of randomizations. arXiv preprint arXiv:1310.4132: 30.

Brien, C.J. (1983). Analysis of variance tables based on experimental structure. *Biometrics*, **39**, 53-59.

Brien, C.J. & Bailey, R.A. (2006). Multiple randomizations. *Journal of the Royal Statistical Society, Series B*, **68**, 571-609.

Brien, C.J. & Bailey, R.A. (2009). Decomposition tables for multitiered experiments. I. A chain of randomizations. *The Annals of Statistics*, **36**, 4184-4213.

Brien, C.J., Harch, B.D., Correll, R.L. & Bailey, R.A. (2011). Multiphase experiments with at

least one later laboratory phase. I. Orthogonal designs. *Journal of Agricultural, Biological and Environmental Statistics*, **16**, 422-450.
Brien, C.J. & Payne, R.W. (1999). Tiers, structure formulae and the analysis of complicated experiments. *The Statistician*, **48**, 41-52.

See also
Procedures: AMTIER, AMTKEEP.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AMTIER

Analyses a multitiered design by an analysis of variance specified by up to three model formulae (C.J. Brien & R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, aovpseudotable, design, effects, fittedvalues); default aovt
F1 = *formula*	First model formula
F2 = *formula*	Second model formula
F3 = *formula*	Third model formula
FACTORIAL = *scalar*	Limit on the number of factors in a model term
F2BALANCETYPE = *string token*	Type of balance required for F2 (orthogonal, firstorder); default orth
F3BALANCETYPE = *string token*	Type of balance required for F3 (orthogonal, firstorder); default orth
PSEUDOTERMS = *formula structures*	Specifies pseudo-terms for terms in the F1, F2 or F3 formulae
DESIGN = *tree*	Saves or specifies details of the design and analysis
SEED = *scalar*	Seed for random numbers to generate dummy variate for determining the design; default 13579
TOLERANCE = *variate*	Tolerance for zero sweeps in dummy and y-variate analyses
DPRINT = *string tokens*	Controls debug output (setup, analysis, dummyanalysis); default * i.e. none

Parameters

Y = *variates*	Each of these contains the data values for an analysis
RESIDUALS = *variates*	Saves the residuals from each analysis
FITTEDVALUES = *variates*	Saves the fitted values from each analysis
SAVE = *pointers*	Save structure for each analysis (to use in AMTDISPLAY)

Description

Genstat users are accustomed to the idea that more than one model formula may be required to specify an analysis of variance. For the ANOVA directive, the underlying structure of the data (which indicates the error terms for the analysis) is defined by a model formula specified by the BLOCKSTRUCTURE directive, while the treatment terms to be fitted in the analysis are defined in a model formula specified by the TREATMENTSTRUCTURE directive. However, experiments that involve multiple randomizations (Brien & Bailey 2006), such as two-phase experiments, may require more than two model formulae to define their analysis correctly (Brien & Payne 1999).

For example, Brien (1983) considers a two-phase experiment set up to evaluate a set of wines. These are evaluated at a tasting where several tasters are given the wines over a number of sittings. One wine is presented to each taster at a sitting, and each wine is evaluated only once by each taster. The order of presentation of the wines is randomized for each taster. The basic observational unit is a glass of wine presented to a particular taster in the tasting phase. These have a structure of tasters/sittings. If this phase represented the whole experiment, tasters/sittings would be the block formula, and the treatment formula would be the factor wines. So we would have

```
BLOCKSTRUCTURE tasters/sittings
```

```
TREATMENTSTRUCTURE wines
```

Now suppose that the wines were produced from a field experiment and, in fact, that each one was produced from one of the plots of a randomized-block design. The second model formula would then be `blocks/plots`, and the final formula would be `treatments` (the factor identifying the treatments applied in the field).

`AMTIER` can analyse designs requiring up to three model formulae. It can thus analyse any design with three tiers, and also some with more than three; see, for example, the corn experiment in Brien & Bailey (2006, example 12). The formulae are specified by the options `F1`, `F2` and `F3`, which must not contain the pseudofactorial operator. For the example in Brien (1983), the statement would be

```
AMTIER [F1=tasters/sittings; F2=blocks/plots; \
       F3=treatments] Y
```

The `FACTORIAL` option sets a limit on the number of factor in the model terms generated from the formulae.

The `Y` parameter specifies the response variate. Residuals and fitted values can be saved by the `RESIDUALS` and `FITTEDVALUES` parameters, respectively. The `SAVE` parameter can save a pointer containing the full details of the analysis. This can be used as input to the `AMTDISPLAY` procedure to obtain further output.

The `F2BALANCETYPE` and `F3BALANCETYPE` options control whether the terms from the second and third model formulae are allowed to be first-order balanced rather than orthogonal (Brien & Bailey 2007). The default is that the terms are required to be orthogonal. It is emphasized that this applies only to terms from the same model formula. Even if the terms from a model formula are required to be orthogonal, they may still only be structure balanced in relation to terms from other formulae. However, if terms from any model formula are non-orthogonal, then the experiment is not structure balanced (Brien & Bailey 2007) and so sums of squares for sources differ depending on their order in the model formula.

The `PSEUDOTERMS` option allows you to specify a list of formula structures defining pseudo-terms for some of the terms in the formulae. Each pseudoterm formula is of the form

```
group_term // pseudoterms_formula
```

All pseudo-terms must be defined explicitly as none are generated, for example from relations between the group term and other factors. Furthermore, all marginal terms to a pseudoterm need to be included in its formula, irrespective of whether they themselves are pseudoterms. Those that are not pseudo-terms need to occur in one of the three main model formulae and will not be included in the analysis sequence again as a result of their appearance in the pseudo-term formula. The pseudo-terms are placed immediately before the group term in the analysis sequence. Any repetitions of pseudo-terms are removed.

The `DESIGN` option can save a tree structure representing the design and analysis. You can then specify this as the design in a subsequent `AMTIER` statement, to avoid having to go through the process of determining the design structure with another response variate from the same experiment. The design structure is determined by a similar dummy analysis process as in the standard `ANOVA` directive. The `TOLERANCE` option specifies a variate with two values. The first defines the tolerance multiplier for zero sweeps in the dummy analysis and the second defines the multiplier for use in the analysis of the y-variates. The `SEED` option sets the starting value for the random generator that is used to generate variates to be used in the dummy analysis.

Printed output is controlled by the `PRINT` option with settings:

`aovtable`	to print the analysis-of-variance table,
`aovpseudotable`	to print the analysis-of-variance table with lines for all the pseudo-terms (generated by pseudo-factors) given explicitly,
`design`	to display the structure of the design,
`effects`	to print tables of effects and residuals, and

`fittedvalues`	to print a table with the y-variate, fitted valued and residuals.

The `DPRINT` option controls debug output, with settings:

`setup`	for information from the set-up stage,
`analysis`	for information from the analysis of the y-variates, and
`dummyanalysis`	for information from the dummy analysis.

Options: `PRINT`, `F1`, `F2`, `F3`, `FACTORIAL`, `F2BALANCETYPE`, `F3BALANCETYPE`, `PSEUDOTERMS`, `DESIGN`, `SEED`, `TOLERANCE`, `DPRINT`.
Parameters: `Y`, `RESIDUALS`, `FITTEDVALUES`, `SAVE`.

Method
Multitiered experiments are defined by Brien (1983), their design is discussed by Brien & Bailey (2006) and Brien *et al.* (2011), and their analysis of variance is described by Brien & Payne (1999), Brien & Bailey (2009) and Bailey & Brien (2013).

Action with `RESTRICT`
There must not be any restrictions.

References
Bailey, R.A. & Brien C.J. (2013). Randomization-based models for multitiered experiments. I. A chain of randomizations. arXiv preprint arXiv:1310.4132: 30.

Brien, C.J. (1983). Analysis of variance tables based on experimental structure. *Biometrics*, **39**, 53-59.

Brien, C.J. & Bailey, R.A. (2006). Multiple randomizations. *Journal of the Royal Statistical Society, Series B*, **68**, 571-609.

Brien, C.J. & Bailey, R.A. (2009). Decomposition tables for multitiered experiments. I. A chain of randomizations. *The Annals of Statistics*, **36**, 4184-4213.

Brien, C.J., Harch, B.D., Correll, R.L. & Bailey, R.A. (2011). Multiphase experiments with at least one later laboratory phase. I. Orthogonal designs. *Journal of Agricultural, Biological and Environmental Statistics*, **16**, 422-450.

Brien, C.J. & Payne, R.W. (1999). Tiers, structure formulae and the analysis of complicated experiments. *The Statistician*, **48**, 41-52.

See also
Procedures: `AMTDISPLAY`, `AMTKEEP`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AMTKEEP

Saves information from the analysis of a multitiered design by AMTIER (C.J. Brien & R.W. Payne).

Options

RESIDUALS = *variate*	Saves the residuals
FITTEDVALUES = *variate*	Saves the fitted values
AOVTABLE = *pointer*	Saves the analysis-of-variance table
SKELETON = *string token*	Whether to save only the skeleton analysis-of-variance table (yes, no); default no
PSEUDOLINES = *string token*	Whether to include lines for pseudo-terms in the analysis-of-variance table (yes, no); default no
OMITMISSINGLINES = *string token*	Whether to omit lines of the analysis-of-variance table that contain only missing values (yes, no); default no
SAVE = *pointer*	Save structure for the analysis; if this is not set, information is saved from the most recent AMTIER analysis

No parameters

Description

The AMTIER procedure analyses data from designs that require up to three model formulae to specify their analysis (resulting from three or more tiers for the experiment). Information from an AMTIER analysis can be saved by the SAVE parameter, and input to AMTKEEP using its own SAVE parameter. Alternatively, if SAVE is not set, AMTKEEP will use the information from the most recent AMTIER analysis.

The RESIDUALS and FITTEDVALUES options save the residuals and fitted values, respectively, in variates.

The AOVTABLE option saves the analysis-of-variance table. You can set option PSEUDOLINES=yes to include lines for all the component pseudo-terms of a term; by default lines are included only for the term itself. You can set option SKELETON=yes to obtain a "skeleton" analysis of variance, omitting the columns for sums of squares, mean squares and variance ratios. You can set option OMITMISSINGLINES=yes to omit lines of the analysis-of-variance table, such as stratum headers, that contain only missing values.

Options: RESIDUALS, FITTEDVALUES, AOVTABLE, SKELETON, PSEUDOLINES, OMITMISSINGLINES, SAVE.
Parameters: none.

Method

Multitiered experiments are defined by Brien (1983), their design is discussed by Brien & Bailey (2006) and Brien *et al.* (2011), and their analysis of variance is described by Brien & Payne (1999), Brien & Bailey (2009) and Bailey & Brien (2013).

References

Bailey, R.A. & Brien C.J. (2013). Randomization-based models for multitiered experiments. I. A chain of randomizations. arXiv preprint arXiv:1310.4132: 30.

Brien, C.J. (1983). Analysis of variance tables based on experimental structure. *Biometrics*, **39**, 53-59.

Brien, C.J. & Bailey, R.A. (2006). Multiple randomizations. *Journal of the Royal Statistical*

Society, Series B, **68**, 571-609.
Brien, C.J. & Bailey, R.A. (2009). Decomposition tables for multitiered experiments. I. A chain of randomizations. *The Annals of Statistics*, **36**, 4184-4213.
Brien, C.J., Harch, B.D., Correll, R.L. & Bailey, R.A. (2011). Multiphase experiments with at least one later laboratory phase. I. Orthogonal designs. *Journal of Agricultural, Biological and Environmental Statistics*, **16**, 422-450.
Brien, C.J. & Payne, R.W. (1999). Tiers, structure formulae and the analysis of complicated experiments. *The Statistician*, **48**, 41-52.

See also
Procedures: `AMTIER`, `AMTDISPLAY`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ANTMVESTIMATE

Estimates missing values in repeated measurements (M.G. Kenward & R.W. Payne).

Options

PRINT = *string tokens*	Controls output from the procedure (meanprofiles); default * i.e. none
GROUPS = *factor*	Factor indicating the plot on which each sequence of observations was made
ORDER = *scalar*	Order of ante-dependence structure (i.e. number of past times for which to adjust)

Parameters

DATA = *variates*	Observations at each time
NEWDATA = *variates*	Data variates with missing observations replaced by their estimates
MEANPROFILE = *tables*	Estimated mean profiles at each time

Description

Suppose that we have a set of experimental units, or plots, within which observations are made in several locations at a sequence of times. Data from some of the locations may be missing at various times. The observed data values are specified in separate variates, one for each time point, using the DATA parameter. The factor identifying the experimental unit on which each observation was made is specified using the GROUPS option.

ANTMVESTIMATE assumes that the data have an ante-dependence (AD(r)) covariance structure whose order can be specified using the ORDER option; if this is not set, ANTMVESTIMATE takes the maximum possible order, number of times minus one. Using this assumption, ANTMVESTIMATE estimates the missing values and calculates the mean profiles for each unit. These can be saved, in tables indexed by the GROUPS factor, using the MEANPROFILES parameter, or printed by setting the PRINT option to meanprofiles. Also, the NEWDATA parameter allows new variates to be saved with the missing values replaced by their estimates.

Options: PRINT, GROUPS, ORDER. Parameters: DATA, NEWDATA, MEANPROFILE.

Method

The algorithm in the procedure is a first-order approximation to maximum likelihood estimation which has the advantage of requiring only one pass through the data. At each time point, current plot means are estimated using the equations of maximum likelihood under an AD(r) covariance structure. The calculations required are simply those of analysis of covariance with previous individual measurements as covariates. Where previous measurements are missing they are replaced by previously estimated mean values and if there are no previous missing values the estimated plot means are full maximum likelihood estimates. The procedure uses a single pass through the time points. If the whole cycle were iterated to convergence joint maximum likelihood estimates of all the plot means would be obtained. Full details are given by Kenward (1994).

Action with RESTRICT

Any restriction on the data variates will be cancelled and a warning printed.

Reference

Kenward, M.G. (1994). *The estimation of mean plot profiles and the identification of atypical plots using incomplete sequences of porous cup nitrate levels*. Rothamsted Technical Report

written for ADAS Biometric Unit, Cheltenham.

See also
Procedures: ANTORDER, ANTTEST, MULTMISSING.
Genstat Reference Manual 1 Summary section on: Repeated measurements.

ANTORDER

Assesses order of ante-dependence for repeated measures data (M.S. Ridout & R.W. Payne).

Options

TREATMENTSTRUCTURE = *formula*	Treatment formula for the model at each time; if this is not set, the default is taken from the setting (which must already have been defined) of the TREATMENTSTRUCTURE directive
BLOCKSTRUCTURE = *formula*	Block formula for the model at each time; if this is not set, the default is taken from any existing setting specified by the BLOCKSTRUCTURE directive and if neither has been set the design is assumed to be unstratified (i.e. to have a single error term)
MAXORDER = *scalar*	Maximum order against which to test; default is maximum possible order
FACTORIAL = *scalar*	Limit on the number of factors in a treatment term
TIME = *factor*	Indicates the time of each observation when there is a single DATA variate

Parameter

DATA = *variates*	Data observations either in a list of variates (one for each time), or a single variate (with TIME set to a factor indicating the time of each observation)

Description

A repeated measures experiment is one in which the same set of units, or subjects, is observed at a sequence of times to investigate treatment effects over a period of time. The set of variates observed at the successive times is said to have an ante-dependence structure of order *r* if each *i*th variate (*i*>*r*), given the preceding *r*, is independent of all further preceding variates (Gabriel 1961, 1962). Procedure ANTORDER calculates statistics to assist in the selection of an appropriate order of ante-dependence structure for sets of repeated measures data, using the method of Kenward (1987). Once the order of ante-dependence structure has been established, the individual variates can be analysed individually by analysis of covariance, adjusting for the *r* previous variates, to assess the times at which treatment effects occurred. Also, procedure ANTTEST can be used to perform overall tests of treatment effects.

The model for the analysis is specified by options of the procedure. TREATMENTSTRUCTURE specifies a model formula to define the treatment terms in the analysis; if this is unset, ANTORDER will use the model already defined by the TREATMENTSTRUCTURE directive, or will fail if that too has not been set. BLOCKSTRUCTURE defines the underlying structure of the design, and ANTORDER will use the model (if any) previously defined by the BLOCKSTRUCTURE directive if this is not set; these can both be omitted if there is only one error term (i.e. if the design is unstratified). Option MAXORDER specifies the maximum order of ante-dependence structure to be tested; by default, this is taken as the maximum possible order (the smaller of the number of times minus one or the number of residual degrees at each time; see Kenward 1987). The FACTORIAL option can be used to set a limit on the number of factors in the terms generated from the treatment formula.

The data are specified by the DATA parameter in one of two ways. The first is to supply a list of variates, each one containing the measurements made on the subjects at one of the successive occasions on which they were observed.

The second possibility is to supply a single DATA variate containing the data from all the times. The TIME option must then be set to a factor indicating the time of each observation. The

block and treatment factors must be defined to match the DATA variate, and each subject should be represented by a unique combination of the block factors. If not, Genstat prints a warning and assumes that the subjects occur in the same order within each time.

The data may contain missing values but these should represent "dropouts": that is, once subjects start to record missing values, their observations should continue to be missing at all subsequent times.

Options: TREATMENTSTRUCTURE, BLOCKSTRUCTURE, MAXORDER, FACTORIAL, TIME.
Parameter: DATA.

Method
The procedure uses the method of Kenward (1987) to calculate the statistics using residual sums of squares from analysis of covariance. For further details of ante-dependence see Gabriel (1961, 1962).

Action with RESTRICT
Any restriction on the DATA variates will be applied to all of them.

References
Gabriel, K.R. (1961). The model of ante-dependence for data of biological growth. *Bulletin Institut International Statistique (Paris)*, **39**, 253-264, (33rd session).

Gabriel, K.R. (1962). Ante-dependence analysis of an ordered set of variables. *Annals of Mathematical Statistics*, **33**, 201-212.

Kenward, M.G. (1987). A method for comparing profiles of repeated measurements, *Applied Statistics*, **36**, 296-308.

See also
Directive: VSTRUCTURE.
Procedures: ANTTEST, ANTMVESTIMATE.
Genstat Reference Manual 1 Summary section on: Repeated measurements.

ANTTEST

Calculates overall tests based on a specified order of ante-dependence (R.W. Payne & M.S. Ridout).

Options

TREATMENTSTRUCTURE = *formula*	Treatment formula for the model at each time; if this is not set, the default is taken from the setting (which must already have been defined) of the TREATMENTSTRUCTURE directive
BLOCKSTRUCTURE = *formula*	Block formula for the model at each time; if this is not set, the default is taken from any existing setting specified by the BLOCKSTRUCTURE directive and if neither has been set the design is assumed to be unstratified (i.e. to have a single error term)
ORDER = *scalar*	Number of past times for which to adjust; default is maximum possible order
FACTORIAL = *scalar*	Limit on the number of factors in a treatment term
TIME = *factor*	Indicates the time of each observation when there is a single DATA variate

Parameter

DATA = *variates*	Data observations either in a list of variates (one for each time), or a single variate (with TIME set to a factor indicating the time of each observation)

Description

A repeated measures experiment is one in which the same set of units, or subjects, is observed at a sequence of times to investigate treatment effects over a period of time. The set of variates observed at the successive times is said to have an ante-dependence structure of order *r* if each *i*th variate (*i*>*r*), given the preceding *r*, is independent of all further preceding variates (Gabriel 1961, 1962). Procedure ANTTEST calculates overall tests of treatment terms based on a specified order of ante-dependence structure (see Kenward, 1987).

The model for the analysis is specified by options of the procedure. TREATMENTSTRUCTURE specifies a model formula to define the treatment terms in the analysis; if this is unset, ANTTEST will use the model already defined by the TREATMENTSTRUCTURE directive, or will fail if that too has not been set. BLOCKSTRUCTURE defines the underlying structure of the design, and ANTTEST will use the model (if any) previously defined by the BLOCKSTRUCTURE directive if this is not set; these can both be omitted if there is only one error term (i.e. if the design is unstratified). Option ORDER specifies the order of ante-dependence structure to be assumed for the tests; by default, this is taken as the maximum possible order (the smaller of the number of times minus one or the number of residual degrees at each time). A suitable order can be established using the ANTORDER procedure. The FACTORIAL option can be used to set a limit on the number of factors in the terms generated from the treatment formula.

The data observations are specified by the DATA parameter in one of two ways. The first is to supply a list of variates, each one containing the measurements made on the subjects at one of the successive occasions on which they were observed.

The second possibility is to supply a single DATA variate containing the data from all the times. The TIME option must then be set to a factor indicating the time of each observation. The block and treatment factors must be defined to match the DATA variate, and each subject should be represented by a unique combination of the block factors. If not, Genstat prints a warning and assumes that the subjects occur in the same order within each time.

The data may contain missing values but these should represent "dropouts": that is, once subjects start to record missing values, their observations should continue to be missing at all subsequent times.

Options: TREATMENTSTRUCTURE, BLOCKSTRUCTURE, ORDER, FACTORIAL, TIME.
Parameter: DATA.

Method
The procedure uses the method of Kenward (1987) to calculate the statistics using residual sums of squares from analysis of covariance. For further details of ante-dependence see Gabriel (1961, 1962).

Action with RESTRICT
Any restriction on the DATA variates will be applied to all of them.

References
Gabriel, K.R. (1961). The model of ante-dependence for data of biological growth. *Bulletin Institut International Statistique (Paris)*, **39**, 253-264, (33rd session).
Gabriel, K.R. (1962). Ante-dependence analysis of an ordered set of variables. *Annals of Mathematical Statistics*, **33**, 201-212.
Kenward, M.G. (1987). A method for comparing profiles of repeated measurements, *Applied Statistics*, **36**, 296-308.

See also
Directive: VSTRUCTURE.
Procedures: ANTORDER, ANTMVESTIMATE.
Genstat Reference Manual 1 Summary section on: Repeated measurements.

AN1ADVICE

Aims to give useful advice if a design that is thought to be balanced fails to be analysed by ANOVA (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (advice, suspects); default advi
FACTORIAL = *scalar*	Limit on number of factors in a treatment term; default 3
METHOD = *string tokens*	Method to use to predict the correct pattern of replication (median, mode, proportional); default mode
WEIGHTS = *variate*	Weights for the analysis; default * i.e. all units have weight one
SUSPECTS = *variate*	Saves the numbers of the units whose factor values are suspected to be incorrect

Parameter

Y = *variates*	Data values to be analysed (this is needed only if the analysis is to take place on a restricted set of units)

Description

The ANOVA directive analyses "balanced" designs. These include most of the commonly occurring experimental designs such as randomized blocks, Latin squares, split plots and other orthogonal designs, as well as designs with balanced confounding like balanced incomplete blocks or balanced lattices. Details of the necessary condition of *first-order balance* are given by Wilkinson (1970), Payne & Wilkinson (1976) and Payne & Tobias (1992). However, ANOVA can itself detect whether or not a design can be analysed, so if you are not sure whether or not a particular design is analysable, you can run it through ANOVA and see whether it succeeds or fails with an "AN 1" diagnostic. Sometimes the design will genuinely be unbalanced, but on other occasions it may be that errors have been made in entering the data. So the aim of AN1ADVICE is to give useful advice if you find that a set of data that you had expected to be balanced fails to be analysed by ANOVA.

The use of AN1ADVICE is very similar to ANOVA. You must first define the model that is to be fitted in the analysis, using the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives. As in ANOVA, the treatment terms to be included in the model are controlled by the FACTORIAL option, and the WEIGHT option can specify weights for a weighted analysis of variance.

AN1ADVICE has a parameter Y to specify the variate whose values are being analysed. However, this is required only if you are analysing a subset of the units. (You would then have used the RESTRICT directive, directly or through a menu, to restrict Y to the units concerned.)

In a balanced design, the joint replications of sets of factors in the design will usually have a systematic pattern. Often there will be equal replication. Then, for example, if you look at the replication table for any pair of factors, it will contain a single value (the number of times each pair of their levels occurs in the design). Alternatively, the replications may have a proportional pattern. For example, you may have a "control" level of one of the factors with perhaps twice as many replicates as the other, "test", levels. Then, in every replication table involving that factor, the cells for the "control" level will have values twice as large as those in the corresponding "test" cells. So AN1ADVICE examines the factors in the model terms that ANOVA has found to be unbalanced, and examines their replications to try to identify cells whose values seem to be too small or too large.

The METHOD option controls how AN1ADVICE works out what the replication in each table ought to be. The default setting, mode, assumes that the values should all be equal, and that the

non-zero value that occurs most often in the table is the correct one. The setting `median` is similar except that the right value is assumed to be the median of the non-zero values. Finally, the `proportional` setting estimates the correct values for each table by assuming that the replication has a proportional pattern.

The `PRINT` option controls the printed output, with settings:

`advice`	prints advice including replication tables of the factors that seem to be incorrect, highlighting the cells that seem to be too small or too large, and
`suspects`	prints the units with the combinations of factor levels that seem to occur too often in the design.

The default is `PRINT=advice`. The list of suspect units can also be saved, in a variate, using the `SUSPECTS` option.

If you believe that the design should be balanced, you may find that the factor values (or weights) of some of suspect units have been entered incorrectly. Alternatively, you may find that some units with the factor combinations whose replication has been highlighted as too low have been accidentally omitted from the data. If these mistakes can be corrected, the design may become balanced. Alternatively, if you cannot find any mistakes in the data, you will need to use regression or `REML` instead of `ANOVA`.

Options: `PRINT`, `FACTORIAL`, `METHOD`, `WEIGHTS`, `SUSPECTS`.
Parameter: `Y`.

Action with `RESTRICT`
You can restrict the set of units used for the analysis by applying a restriction to the y-variate.

References
Payne, R.W. & Wilkinson, G.N. (1977). A general algorithm for analysis of variance. *Applied Statistics*, **26**, 251-260.

Payne, R.W. & Tobias, R.D. (1992). General balance, combination of information and the analysis of covariance. *Scandinavian Journal of Statistics*, **19**, 3-23.

Wilkinson, G.N. (1970). A general recursive algorithm for analysis of variance. *Biometrika*, **57**, 19-46.

See also
Directives: `ANOVA`, `BLOCKSTRUCTURE`, `TREATMENTSTRUCTURE`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AONEWAY

Performs one-way analysis of variance (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis of variance (aovtable, information, covariates, effects, residuals, contrasts, means, cbeffects, cbmeans, stratumvariances, %cv, missingvalues, homogeneity, permutationtest); default aovt, mean, miss
GROUPS = *factor*	Defines the treatments for the analysis
COVARIATES = *variates*	Covariates (if any) for analysis of covariance
PLOT = *string tokens*	Which residual plots to provide (fittedvalues, normal, halfnormal, histogram, absresidual); default fitt, norm, half, hist
GRAPHICS = *string token*	Type of graphs (lineprinter, highresolution); default high
FPROBABILITY = *string token*	Probabilities for variance ratio (yes, no); default no
PSE = *string tokens*	Types of standard errors to be printed with the means (differences, lsd, means); default diff
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
NTIMES = *scalar*	Number of random allocations to make when PRINT=perm; default 999
SEED = *scalar*	Seed for the random number generator used to make the allocations; default 0 continues from the previous generation or (if none) initializes the seed automatically

Parameters

Y = *variates*	Each of these contains the data values for an analysis
RESIDUALS = *variates*	Saves the residuals from each analysis
FITTEDVALUES = *variates*	Saves the fitted values from each analysis

Description

AONEWAY provides customized facilities and output for one-way analysis of variance. For example, if the treatments have unequal replication, a standard error is printed for each mean, rather than the summary for comparisons of means with minimum and maximum replication as given by ANOVA. Similarly, any missing values are excluded from the analysis by AONEWAY. Conversely, in ANOVA they need to be included, to ensure balance in the more general situations that it covers, and are estimated as part of the analysis. In addition, AONEWAY provides residual plots directly, instead of requiring you to use procedure APLOT after the analysis, and it can test the homogeneity of the variances within the groups.

The Y parameter supplies a variate containing the data values to be analysed. The factor defining the groups to be compared is supplied by the GROUPS option. You can either specify just the factor to produce a simple one-way anova, or you can put it within a POL, REG or COMPARISON function to fit some contrasts at the same time. There is also a COVARIATES option which can supply one or more variates to be used as covariates in an analysis of covariance.

Printed output is requested by listing the required components with the PRINT option. The most relevant settings are:

aovtable	to print the analysis-of-variance table;
means	to print the table of means;

effects	to print the effects (means minus grand mean);
%cv	to print the coefficient of variation;
missingvalues	to print estimates for missing values (if any);
homogeneity	to print tests for the homogeneity of the variances within the groups; and
permutationtest	analysis-of-variance table with the probabilities calculated by a random permutation test.

However, for compatibility, all the settings of the PRINT option of ANOVA are included. By default, when PRINT=perm, AONEWAY makes 999 random allocations of the data to the two samples (using a default seed), and determines the probabilities of the variance ratios from their distribution over these randomly generated datasets. (It therefore makes no assumptions about the distribution of the data values.) The NTIMES option allows you to request another number of allocations, and the SEED option allows you to specify another seed. AONEWAY checks whether NTIMES is greater than the number of possible ways in which the data values can be allocated. If so, it does an exact test instead, which takes each possible allocation once.

The FPROBABILITY option can be set to yes to print of probabilities for variance ratios in the analysis-of-variance table. The PSE option controls the standard errors printed with the tables of means. The default setting is differences, which gives standard errors of differences of means. The setting means produces standard errors of means, LSD produces least significant differences, and by setting PSE=* the standard errors can be suppressed altogether. The significance level to use in the calculation of the least significant differences can be changed from the default of 5% using the LSDLEVEL option.

The PLOT option allows up to four of the following residual plots to be requested:

fittedvalues	for a plot of residuals against fitted values;
normal	for a Normal plot;
halfnormal	for a half-Normal plot;
histogram	for a histogram of residuals; and
absresidual	for a plot of the absolute values of the residuals against the fitted values.

By default the first four are produced. The GRAPHICS option determines the type of graphics that is used, with settings highresolution (the default) and lineprinter.

Variates of residuals and fitted values can be saved using the RESIDUALS and FITTEDVALUES parameters, respectively. Directive AKEEP can be used to save other information from the analysis of the last data variate to be analysed by the procedure (see the *Guide to the Genstat Command Language*, Part 2, Section 4.6.1 for details).

Options: PRINT, GROUPS, COVARIATES, PLOT, GRAPHICS, FPROBABILITY, PSE, LSDLEVEL.
Parameters: Y, RESIDUALS, FITTEDVALUES .

Method

AONEWAY uses the standard Genstat facilities for analysis of variance, except that the standard errors and lsd's for the means are saved by AKEEP and then printed, rather than being printed directly (as just a summary) by ADISPLAY. Permutation tests are performed by APERMTEST, residual plots are produced by APLOT, and the homogeneity of variances is tested by VHOMOGENEITY.

Action with RESTRICT

If the Y variate is restricted, only the units not excluded by the restriction will be analysed.

See also
Directive: ANOVA.
Procedure: A2WAY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AOVANYHOW

Performs analysis of variance using ANOVA, regression or REML as appropriate (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, information, means, residuals); default aovt, info, mean
METHOD = *string token*	Whether to complete the analysis or just form a recommendation (analyse, recommend); default anal
FACTORIAL = *scalar*	Limit on number of factors in a treatment term; default 3
FPROBABILITY = *string token*	Printing of probabilities for variance ratios in the analysis-of-variance table (yes, no); default no
PLOT = *string tokens*	Which residual plots to provide (fittedvalues, normal, halfnormal, histogram); default * i.e. none
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
PSE = *string tokens*	Types of standard errors to be printed with the predicted means (differences, alldifferences, lsd, alllsd, means; default diff
WEIGHTS = *variate*	Weights for each unit; default * i.e. all units with weight one
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
EFLOSS = *scalar*	Maximum loss of efficiency occurring on any treatment contrast if the analysis is done by regression
EFLIMIT = *scalar*	Limit on the loss of efficiency for the analysis to be done by regression; default 0.1
EXIT = *scalar*	Exit code indicating the recommended method of analysis

Parameters

Y = *variates*	Data values to be analysed
RESIDUALS = *variates*	Variate to save the residuals from each analysis
FITTEDVALUES = *variates*	Variate to save the fitted values from each analysis
SAVE = *identifiers*	To save details of each analysis to use subsequently with the AOVDISPLAY procedure

Description

AOVANYHOW assesses a data set to select the most appropriate method for analysis of variance. If the design is orthogonal or balanced it uses the ANOVA directive. Otherwise, if there is no blocking in the design (i.e. there is only one random term) it uses the Genstat regression facilities through procedure A2WAY or AUNBALANCED. Finally, if there are additional random terms, it looks to see if these contain any useful information about the treatments in order to choose between regression and REML. The EFLIMIT option sets a limit on the amount of information that may be lost on any of the treatment contrasts if the analysis to be done by regression instead of REML; the default of 0.1 implies that no more than 10% of the information on any contrast may be estimated between the random terms.

The method of use is similar to that for ANOVA. The treatment terms to be fitted must be specified, before calling the procedure, by the TREATMENTSTRUCTURE directive. Similarly, any

covariates must be indicated by the COVARIATE directive. Any blocking structure must be specified by the BLOCKSTRUCTURE directive.

The parameters of the procedure are identical to those of ANOVA. The variates to be analysed are specified by the Y parameter. Residuals and fitted values can be saved using the RESIDUALS and FITTEDVALUES parameters respectively. Finally, the SAVE parameter allows details of the analysis to be saved so that further output can be obtained using the AOVDISPLAY procedure.

Printed output is controlled by the PRINT option. The settings are limited to those that can produce analogous output from any of the analysis methods:

aovtable	analysis-of-variance table from ANOVA or regression, or Wald and F tests for fixed effects from REML,
information	design type, efficiency factors and name of the command used for the analysis,
means	tables of (predicted) means, and
residuals	residuals (fitted values are printed too for analyses by regression or REML).

Probabilities can be printed for variance ratios by setting option FPROBABILITY=yes.

The SAVE parameter allows you to save a pointer containing information about the analysis. You can use this as the input for the SAVE parameter of the AOVDISPLAY procedure to print (or reprint) any of the information provided by the PRINT option above. Alternatively, the first element of the pointer is the save structure from the command that was used for the analysis. So, if you use this with the display commands associated with that analysis command, you can display the more specialized output from the command (for example, variance components from REML).

Tables of means from regression and REML are calculated using the PREDICT and VPREDICT directives, respectively. The first step (A) of their calculations forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, for regression analyses, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data.

The PSE option controls the types of standard errors that are produced to accompany the tables of means, with settings:

differences	summary of standard errors for differences between pairs of means,
alldifferences	standard errors for differences between all pairs of means,
lsd	summary of least significant differences between pairs of means,
alllsd	least significant differences between all pairs of means,
means	effective standard errors for analyses by ANOVA, or approximate effective standard errors for analyses by regression or REML – these are formed by procedure SED2ESE with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula of $sed_{i,j} = \sqrt{(ese_i^2 + ese_j^2)}$.

The default is differences. The LSDLEVEL option sets the significance level (as a percentage) for the least significant differences.

The PLOT option allows various residual plots to be requested: fittedvalues for a plot of residuals against fitted values, normal for a Normal plot, halfnormal for a half Normal plot, and histogram for a histogram of residuals.

The FACTORIAL option sets a limit on the number of factors that a higher-order term, such as an interaction, can contain; any terms with more factors are deleted from the analysis. The WEIGHTS option allows a variate of weights to be specified for a weighted analysis of variance.

You can save a scalar indicating the recommended method of analysis by using the EXIT option. The scalar can take values with the following meanings.

0. The design is orthogonal. Analyse by ANOVA.
1. The design is balanced. Analyse by ANOVA.
2. The design unbalanced. It has 1 or 2 treatment factors and no blocking. Analyse by A2WAY.
3. The design unbalanced and has 1 or 2 treatment factors. No more than a proportion defined by the EFLIMIT option of the information on any treatment contrast is estimated between block terms. Analyse by A2WAY.
4. The design unbalanced, and there are either weights or more than 2 treatment factors. There is no blocking. Analyse by AUNBALANCED.
5. The design is unbalanced, and there either are weights or more than 2 treatment factors. No more than a proportion defined by the EFLIMIT option of the information on any treatment contrast is estimated between block terms. Analyse by AUNBALANCED.
6. The design unbalanced with several block (i.e. random) terms. Analyse by REML.

The EFLOSS option can save the maximum loss of efficiency that would occur on any treatment contrast if the analysis is done by regression.

You can set option METHOD=recommend to request that AOVANYHOW will just form a recommendation for the command to be used if the analysis cannot be done by ANOVA. The only available PRINT option is then information, which tells you which command is recommended. You can still use the EXIT and EFLOSS options, but residuals and fitted values will be saved (by the RESIDUALS and FITTEDVALUES parameters) only if the analysis should be done by ANOVA.

Options: PRINT, METHOD, FACTORIAL, FPROBABILITY, PLOT, COMBINATIONS, ADJUSTMENT, PSE, WEIGHTS, LSDLEVEL, EFLOSS, EFLIMIT, EXIT.
Parameters: Y, RESIDUALS, FITTEDVALUES, SAVE.

Method

The EXIT option of the ANOVA directive is used to ascertain whether or not the design is orthogonal or balanced; if so it can be analysed by ANOVA. (For details, see the *Guide to the Genstat Command Language*, Part 2 Statistics, Section 4.7.) If the design is not orthogonal or balanced and there are several random terms, the AEFFICIENCY procedure is used to calculate the efficiency factors for the treatment terms, in order to decide whether to use regression of REML.

Action with RESTRICT

If the Y variate or any of the factors or covariates is restricted, only the units not excluded by the restriction will be analysed.

See also
Directives: ANOVA, REML.
Procedures: AOVDISPLAY, AN1ADVICE, AUNBALANCED, A2WAY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AOVDISPLAY

Provides further output from an analysis by AOVANYHOW (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, information, means, residuals); default aovt, info, mean
FPROBABILITY = *string token*	Printing of probabilities for variance ratios in the analysis-of-variance table (yes, no); default no
PLOT = *string tokens*	Which residual plots to provide (fittedvalues, normal, halfnormal, histogram); default * i.e. none
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
PSE = *string tokens*	Types of standard errors to be printed with the predicted means (differences, alldifferences, lsd, alllsd, means; default diff
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
EFLOSS = *scalar*	Maximum loss of efficiency occurring on any treatment contrast if the analysis is done by regression
EXIT = *scalar*	Code indicating the method of analysis

Parameters

SAVE = *identifiers*	Save structure from AOVANYHOW; default uses the save structure from the most recent AOVANYHOW analysis

Description

The AOVANYHOW procedure assesses a data set to select the most appropriate method for analysis of variance. If the design is orthogonal or balanced it uses the ANOVA directive. Otherwise, if there is no blocking in the design (i.e. there is only one random term) it uses the Genstat regression facilities through procedure A2WAY or AUNBALANCED. Finally, if there are additional random terms, it looks to see if these contain any useful information about the treatments in order to choose between regression and REML.

This procedure, AOVDISPLAY, allows further output to be produced from an analysis by AOVANYHOW. By default, the output is from the most recent analysis done by AOVANYHOW. However, you can print the output from an earlier analysis by setting the SAVE parameter to a pointer containing the analysis information, saved earlier using the SAVE parameter of AOVANYHOW.

The printed output is controlled by the PRINT option. The settings are limited to those that can produce analogous output from any of the analysis methods:

aovtable	analysis-of-variance table from ANOVA or regression, or Wald and F tests for fixed effects from REML,
information	design type, efficiency factors and name of the command used for the analysis,
means	tables of (predicted) means, and
residuals	residuals (fitted values are printed too for analyses by regression or REML).

Probabilities can be printed for variance ratios by setting option FPROBABILITY=yes.

Tables of means from regression and REML are calculated using the PREDICT and VPREDICT

directives, respectively. The first step (A) of their calculations forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the `table of means`. The `COMBINATIONS` option specifies which cells of the full table are to be formed in Step A. The default setting, `estimable`, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting `COMBINATIONS=present` excludes the cells for factor combinations that do not occur in the data. The `ADJUSTMENT` option then defines how the averaging is done in Step B. The default setting, `marginal`, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting `equal` weights all the combinations equally. Finally, for regression analyses, the setting `observed` uses the `WEIGHTS` option of `PREDICT` to weight each factor combination according to its own individual replication in the data.

The `PSE` option controls the types of standard errors that are produced to accompany the tables of means, with settings:

`differences`	summary of standard errors for differences between pairs of means,
`alldifferences`	standard errors for differences between all pairs of means,
`lsd`	summary of least significant differences between pairs of means,
`alllsd`	least significant differences between all pairs of means,
`means`	effective standard errors for analyses by `ANOVA`, or approximate effective standard errors for analyses by regression or `REML` – these are formed by procedure `SED2ESE` with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula of $sed_{i,j} = \sqrt{(ese_i^2 + ese_j^2)}$.

The default is `differences`. The `LSDLEVEL` option sets the significance level (as a percentage) for the least significant differences.

The `PLOT` option allows various residual plots to be requested: `fittedvalues` for a plot of residuals against fitted values, `normal` for a Normal plot, `halfnormal` for a half Normal plot, and `histogram` for a histogram of residuals.

You can save a scalar indicating the recommended method of analysis by using the `EXIT` option. The scalar can take values with the following meanings.

0. The design is orthogonal. Analyse by `ANOVA`.
1. The design is balanced. Analyse by `ANOVA`.
2. The design unbalanced. It has 1 or 2 treatment factors and no blocking. Analyse by `A2WAY`.
3. The design unbalanced and has 1 or 2 treatment factors. No more than a proportion defined by the `EFLIMIT` option of the information on any treatment contrast is estimated between block terms. Analyse by `A2WAY`.
4. The design unbalanced, and there are either weights or more than 2 treatment factors. There is no blocking. Analyse by `AUNBALANCED`.
5. The design is unbalanced, and there either are weights or more than 2 treatment factors. No more than a proportion defined by the `EFLIMIT` option of the information on any treatment contrast is estimated between block terms. Analyse by `AUNBALANCED`.
6. The design unbalanced with several block (i.e. random) terms. Analyse by `REML`.

The `EFLOSS` option can save the maximum loss of efficiency that would occur on any treatment contrast if the analysis is done by regression.

Options: PRINT, FPROBABILITY, PLOT, COMBINATIONS, ADJUSTMENT, PSE, LSDLEVEL, EFLOSS, EXIT.
Parameter: SAVE.

Action with RESTRICT
If the Y variate or any of the factors or covariates was restricted, only the units not excluded by the restriction will have been analysed.

See also
Procedure: AOVANYHOW.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

†APAPADAKIS

Analysis of variance with an added Papadakis covariate, formed from neighbouring residuals (D.B. Baird).

Options

`PRINT` = *string tokens*	Output from the analysis of the y-variates, adjusted for covariates (`aovtable`, `information`, `covariates`, `effects`, `residuals`, `contrasts`, `means`, `cbeffects`, `cbmeans`, `stratumvariances`, `%cv`, `missingvalues`); default `aovt`, `info`, `cova`, `mean`, `miss`
`PLOT` = *string token*	Whether to plot the residuals against the average of neighbouring residuals (`residuals`); default `*` i.e. no plot
`NEIGHBOURS` = *string token*	The neighbours whose residuals are averaged to form the residual covariate (`adjacent`, `rows`, `columns`, `all`); default `adja`
`TREATMENTSTRUCTURE` = *formula*	Defines the treatment structure of the model; default given by the most recent `TREATMENTSTRUCTURE` directive
`BLOCKSTRUCTURE` = *formula*	Defines the blockings structure of the model; default given by the most recent `BLOCKSTRUCTURE` directive
`COVARIATE` = *variates*	Specifies any covariates in addition to the residual (Papadakis) covariate; default given by the most recent `COVARIATE` directive
`FACTORIAL` = *scalar*	Limit on number of factors in a treatment term; default 3
`CONTRASTS` = *scalar*	Limit on the order of a contrast of a treatment term; default 4
`DEVIATIONS` = *scalar*	Limit on the number of factors in a treatment term for the deviations from its fitted contrasts to be retained in the model; default 9
`PSE` = *string token*	Standard errors to be printed with tables of means, `PSE=*` requests s.e.'s to be omitted (`differences`, `lsd`, `means`); default `diff`
`LSDLEVEL` = *scalar*	Significance level (%) to use in the calculation of least significant differences; default 5

Parameters

`Y` = *variates*	Variates to be analysed
`ROWS` = *factors* or *variates*	Factor giving the row location of each plot
`COLUMNS` = *factors* or *variates*	Factor giving the column location of each plot
`UNITS` = *factors* or *variates*	Factor giving the plot location of each unit
`RCOVARIATE` = *variates*	Saves the covariate formed from the mean of the neighbouring residuals
`TITLE` = *texts*	Title for the graph; default i.e. title created from the `Y` variate name and the neighbouring plots that are used
`WINDOW` = *scalars*	Window number for the graph; default 3
`PEN` = *scalars*, *variates* or *factors*	Pen number for the graph; default 1
`SCREEN` = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (`clear`, `keep`); default `clea`

Description

The APAPADAKIS procedure analyses balanced designs with an added covariate formed from the neighbouring residuals from the initial analysis of variance (Papadakis 1937, Bartlett 1938, Wilkinson *et al.* 1983). This method was the first and simplest nearest-neighbour adjustment for removing the effects of spatial trends within a trial. If there is a smooth trend in the trial, the plot's residual will be correlated with the neighbouring plots' residuals. Fitting the average of the neighbouring residuals as a covariate can then adjust the treatment means for the trend and reduce their standard errors. This technique has been superceded by spatial REML analyses, but may still be useful for comparison.

The model to be fitted in the analysis has three parts. The TREATMENTSTRUCTURE specifies the treatment (or *systematic*, or *fixed*) terms for the analysis. The BLOCKSTRUCTURE defines the "underlying structure" of the design or, equivalently, the *error* terms for the analysis; in the simple cases where there is only a single error term this can be omitted. The COVARIATE option specifies any covariates to be included, in addition to the residual (Papadakis) covariate. These can be specified as options in the procedure, or defined by previous TREATMENTSTRUCTURE, BLOCKSTRUCTURE and COVARIATE directives.

The Y parameter lists the variates to be analysed. The ROWS and COLUMNS parameters can define the 2-dimensional spatial layout of the design. Alternatively, the UNITS parameter defines a 1-dimensional spatial layout. If UNITS is not specified for a 1-dimensional layout, APAPADAKIS assumes (with a warning) that the y-values are in plot order.

The NEIGHBOURS option controls which neighbours are averaged to form the residual (Papadakis) covariate. The settings rows, columns and all require a 2-dimensional layout. The neighbours for rows are the two plots on either side in the same row, for columns they are the two plots on either side in the same column, for adjacent they are the 4 plots with an edge in common, and for all they are the eight plots with a side or corner in common. For a 1-dimensional layout, adjacent is the only relevant setting. This uses the plots on either side of the given plot as neighbours. Note: edge plots will have fewer neighbours.

The PRINT option selects which components of output are to be displayed:

aovtable	analysis-of-variance table;
information	information summary, giving details of aliasing and non-orthogonality or of any large residuals;
covariates	estimates of covariate regression coefficients;
effects	tables of estimated treatment parameters;
residuals	tables of estimated residuals;
contrasts	estimated contrasts of treatment effects;
means	tables of predicted means for treatment terms;
cbeffects	estimated effects of treatment terms combining information from all the strata in which each term is estimated;
cbmeans	predicted means for treatment terms combining information from all the strata in which each term is estimated;
stratumvariances	estimated variances of the units in each stratum and stratum variance components;
%cv	coefficients of variation and standard errors of individual units; and
missingvalues	estimates of missing values.

The default is intended to give the output that you will require most often from a full analysis: aovtable, information, covariates, means and missingvalues. However, as with ANOVA, the settings information and missingvalues will not produce any output unless

there is something definite to report.

The PSE option controls the standard errors printed with the tables of means. The default setting is differences, which gives standard errors of differences of means. The setting means produces standard errors of means, LSD produces least significant differences, and you can suppress the standard errors altogether by setting PSE=*. The significance level to use for calculating the least significant differences can be changed from the default of 5% with the LSDLEVEL option.

The treatment terms to be included in the model are controlled by the FACTORIAL option. This sets a limit (by default 3) on the number of factors in a treatment term. Terms containing more than that number are deleted.

The CONTRASTS option places a limit (by default 4) on the order of contrast to be fitted. (Contrasts are defined by using the functions POL, REG, COMPARISON, POLND or REGND in the treatment formula.) For a term involving a single factor, the orders of the successive contrasts run from one upwards, with the deviations term (if any) numbered highest. In interactions between contrasts, the order is the sum of the orders of the component parts.

If your design has few or no degrees of freedom for the residual, you may wish to regard the deviations from some of the fitted contrasts as error components, and assign them to the residual of the stratum where they occur. You can do this by the DEVIATIONS option; its value sets a limit on the number of factors in the terms whose deviations are to be retained in the model. For example, by putting DEVIATIONS=1, the deviations from the contrasts fitted to all terms except main effects will be assigned to error. When deviations have been assigned to error, they will not be included in the calculation of tables of means, which will then be labelled "smoothed". However the associated standard errors of the means are not adjusted for the smoothing.

The RCOVARIATE parameter saves the covariate formed from the neighbouring residuals. Other results from the analysis can be saved with the AKEEP directive, as for the ANOVA directive.

You can set option PLOT=residuals to plot the residuals against the average of neighbouring residuals. The TITLE parameter gives the title for the graph; if this is this not set, an automatic title will be created from the Y variate name and the neighbouring plots that are used. The WINDOW parameter defines the window in which the graph is drawn (default 3). The PEN parameter specifies the pen to use (default 1). Finally, the SCREEN parameter controls whether the graphical display is cleared before the graph is plotted.

Options: PRINT, PLOT, NEIGHBOURS, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE, FACTORIAL, CONTRASTS, DEVIATIONS, PSE, LSDLEVEL.
Parameters: Y, ROWS, COLUMNS, UNITS, RCOVARIATE, TITLE, WINDOW, PEN, SCREEN.

Action with RESTRICT

You can restrict the set of units used for the analysis by applying a restriction to any of the y-variates. Only these units are included in the analysis of each y-variate.

References

Bartlett, M.S. (1938). The approximate recovery of information from replicated field experiments with large blocks. *Journal of Agricultural Science*, **28**, 418-427.

Papadakis, J.S. (1937). Méthode statistique pour les expériences en champ. *Bulletin Institute de L'Ameloration Des Plantes à Salonique,* **23**.

Wilkinson, G.N., Eckert, S.R., Hancock, T.W. and Mayo O. (1983). Nearest neighbour (NN) analysis of field experiments. *Journal of the Royal Statistical Society B*, **45**, 151-178.

See also

Directives: ANOVA, BLOCKSTRUCTURE, TREATMENTSTRUCTURE, ADISPLAY, AKEEP.

Procedures: ACHECK, AGRAPH, APLOT, AFIELDRESIDUALS, APERMTEST, AMCOMPARISON, ARESULTSUMMARY, ASPREADSHEET.

Genstat Reference Manual 1 Summary sections on: Analysis of variance, Design of experiments, REML analysis of linear mixed models.

APERMTEST

Does random permutation tests for analysis-of-variance tables (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (aovtable, critical); default aovt
PLOT = *string*	What to plot (histogram); default *
NTIMES = *scalar*	Number of permutations to make; default 999
EXCLUDE = *factors*	Factors in the block model of the design whose levels are not to be randomized
SEED = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically
AOVTABLE = *pointer*	Saves the aov-table, with permutation probabilities
CRITICAL = *pointer*	Saves the aov-table, with critical values
SAVE = *ANOVA save structure*	Save structure from the analysis of variance; default uses the save structure from the most recent ANOVA

No parameters

Description

Random permutation tests provide an alternative to using the F probabilities printed for variance ratios in an analysis-of-variance table in situations where the assumptions of the analysis are not satisfied. These assumptions can be assessed by studying the residual plots produced by APLOT. In particular, the use of the F distribution to calculate the probabilities is based on the assumption that the residuals from each stratum have Normal distributions with equal variances, and so the histogram of residuals produced by APLOT should look reasonably close to the Normal, bell-shaped curve. Experience shows the analysis is robust to small departures from Normality. APERMTEST can be useful if the histogram looks very non-Normal (and you are unable to redefine the analysis as a generalized linear model; see FIT).

The simplest form of use is simply to specify the command

```
APERMTEST
```

straight after the ANOVA. APERMTEST recovers the necessary information about the analysis automatically, and performs 999 random permutations (made using a default seed). The probability for each variance ratio is then determined from its distribution over the randomly permuted datasets.

The NTIMES option of APERMTEST allows you to request another number of permutations, and the SEED option allows you to specify another seed. APERMTEST checks whether NTIMES is greater than the number of possible permutations available for the data set. If so, APERMTEST does an exact test instead, which uses each possible permutation once.

The information about the analysis is obtained from the save structure of the most recent ANOVA (which is stored automatically within Genstat). You can save the information from any analysis of variance explicitly using the SAVE parameter of ANOVA. You can then perform permutation tests for that analysis by using the save structure as the setting of the SAVE option of APERMTEST. The EXCLUDE option allows you to restrict the randomization so that one or more of the factors in the block model is not randomized. The most common instance where this is required is when one of the treatment factors involves time-order, which cannot be randomized.

Output is controlled by the PRINT option, with settings:

aovtable	for an analysis-of-variance table with the usual F probabilities replaced by those from the permutation test;

and

`critical` for a table giving critical values for each variance ratio.

These can be saved using the `AOVTABLE` and `CRITICAL` parameters.

You can set `PLOT=histogram` to plot histograms showing the variance ratios obtained for each treatment term in the original analysis and the analyses of the permuted data sets.

Options: `PRINT`, `PLOT`, `NTIMES`, `EXCLUDE`, `SEED`, `AOVTABLE`, `CRITICAL`, `SAVE`.
Parameters: none.

Method

If there is no blocking and the treatment combinations have more than one replicate, `APERMTEST` uses `SETALLOCATIONS` to determine the number of unique permutations so that it can see whether an exact test is possible. If so, the permutations are formed using `SETALLOCATIONS` too. Otherwise the number of possible permutations is calculated using the `FACTORIAL` function, and the permutations for the exact test are formed using the `PERMUTE` procedure. For a permutation test, `RANDOMIZE` is used to perform the permutations, taking account of the block structure of the design. The `AOVTABLE` option of `AKEEP` is used to save the variance ratios, and the `QUANTILES` function is used to calculate the critical values.

Action with `RESTRICT`

`APERMTEST` takes account of any restrictions on the y-variate in the analysis of variance (i.e. the variate specified as the `Y` parameter in the earlier `ANOVA` command).

See also

Directive: `ANOVA`.
Procedures: `CHIPERMTEST`, `RPERMTEST`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

APLOT

Plots residuals from an ANOVA analysis (R.W. Payne & A.D. Todd).

Options

RMETHOD = *string token*	Type of residuals to plot (simple, standardized); default simp
INDEX = *variate* or *factor*	X-variable for an index plot; default !(1,2...)
STRATUM = *formula*	The stratum (or error term) whose residuals are to be plotted; the default is to plot the residuals from the final stratum
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	Overall title for the plots; if unset, the identifier of the y-variate is used
SAVE = *ANOVA save structure*	Specifies the analysis from which the residuals and fitted values are to be taken; by default they are taken from the most recent ANOVA

Parameters

METHOD = *string tokens*	Type of residual plot (fittedvalues, normal, halfnormal, histogram, absresidual, index); default fitt, norm, half, hist
PEN = *scalars*, *variates* or *factors*	Pen(s) to use for each plot

Description

Procedure APLOT provides up to four types of plots of residuals from an ANOVA analysis. These are selected using the METHOD parameter, with settings: fitted for residuals versus fitted values, normal for a Normal plot, halfnormal for a half-Normal plot, histogram for a histogram of residuals, absresidual for a plot of the absolute values of the residuals versus the fitted values, and index for a plot against an "index" variable (specified by the INDEX option). The PEN parameter can specify the graphics pen or pens to use for each plot. The TITLE option can supply an overall title. If this is not set, the identifier of the y-variate is used.

The residuals and fitted values are accessed automatically from the structure specified by the SAVE option. If the SAVE option is not set, they are taken from the SAVE structure of the last y-variate to have been analysed by ANOVA. By default, simple residuals are plotted, but you can set option RMETHOD=standardized to plot standardized residuals instead.

If your design has several strata (or error terms), you can set the STRATUM option to plot the residuals from one of the higher strata. If STRATUM is not set, the residuals from the final stratum are plotted.

By default, high-resolution graphics are used. Line-printer graphics can be used by setting option GRAPHICS=lineprinter.

Options: RMETHOD, INDEX, STRATUM, GRAPHICS, TITLE, SAVE.
Parameters: METHOD, PEN.

Method

Residuals and fitted values are accessed, using AKEEP, from the latest ANOVA or from that specified by the SAVE option. The plots are produced using the DRESIDUALS procedure.

Action with RESTRICT
If the y-variate in the ANOVA was restricted, only the units not excluded by the restriction will be included in the graphs.

See also
Directive: ANOVA.
Procedures: ACHECK, AGRAPH, AFIELDRESIDUALS, RCHECK, VPLOT.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

APOLYNOMIAL

Forms equations for single-factor polynomial contrasts fitted by ANOVA (R.W. Payne).

Options

PRINT = *string token*	Whether to print the equation of the polynomial (equation); default equa
SAVE = *ANOVA save structure*	Save structure (from ANOVA) to provide details of the analysis from which the equations are to be formed; default uses the save structure from the most recent ANOVA

Parameters

TERMS = *formula*	Model terms whose polynomial equations are required
COEFFICIENTS = *pointers*	Saves the coefficients of each polynomial

Description

The ANOVA directive fits polynomial contrasts of the effects of a factor by forming orthogonal polynomials (see Section 4.5 of the *Guide to the Genstat Command Language, Part 2 Statistics*). This allows the sums of squares for the factor to be partitioned into the amount that can be explained by a linear relationship, then the extra amount that can be explained if the relationship is quadratic, then the extra amount given by a cubic relationship, and so on. As a result, though, the estimates that are produced by ANOVA are the regression coefficients of the orthogonal polynomials, not the coefficients of the polynomial equation. ANOVA can also estimate interactions between the (orthogonal) polynomial contrasts and other factors.

The polynomial coefficients can, however, be obtained using procedure APOLYNOMIAL. The TERMS parameter specifies the treatment terms whose equations are required. Each term must contain no more than one factor with a polynomial function (POL or POLND), and no factors with regression or comparison functions (REG, REGND or COMPARISON); otherwise it is ignored. If TERMS is not set, APOLYNOMIAL takes the full treatment model (see TREATMENTSTRUCTURE).

APOLYNOMIAL usually prints the equation, but you can set option PRINT=* to suppress this. The COEFFICIENTS parameter can supply a pointer to save the coefficients of the equations. The pointer will contain a pointer for each term. These are given suffixes 0 upwards, corresponding to the powers of the factor in each polynomial.

By default, the equation is formed for the contrasts estimated in the most recent analysis performed by ANOVA, but the SAVE option can be used to supply the save structure from an earlier analysis to use instead.

Option: PRINT.
Parameters: FACTOR, LEVELS, GROUPS, COEFFICIENTS, SAVE.

Method

APOLYNOMIAL first needs to duplicate the process of forming the orthogonal polynomials, regressing each power of the factor levels on the lower powers. Suppose, for example, a fourth-order polynomial was fitted, and the orthogonal polynomials were given by

$p_1 = y$
$p_2 = y^2 - b_{21} \times y$
$p_3 = y^3 - b_{31} \times y - b_{32} \times y^2$
$p_4 = y^4 - b_{41} \times y - b_{42} \times y^2 - b_{41} \times y^3$

and that the estimated coefficients of the orthogonal polynomials were e_1, e_2, e_3 and e_4. The coefficients of the polynomial equation are then calculated as

$c_1 = e_1 - b_{21} \times e_2 - b_{31} \times e_3 - b_{41} \times e_4$

$c_2 = e_2 - b_{32} \times e_3 - b_{42} \times e_4$
$c_3 = e_3 - b_{43} \times e_4$
$c_4 = e_4$

See also
Directives: ANOVA, TREATMENTSTRUCTURE.
Procedure: ADPOLYNOMIAL.
Functions: POL, POLND.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

APOWER

Calculates the power (probability of detection) for terms in an analysis of variance (R.W. Payne).

Options

PRINT = *string token*	Prints the power (power); default powe
TERM = *formula*	Treatment term to be assessed in the analysis
TREATMENTSTRUCTURE = *formula*	Treatment structure of the design; determined automatically from an ANOVA save structure if TREATMENTSTRUCTURE is unset or if SAVE is set
BLOCKSTRUCTURE = *formula*	Block structure of the design; determined automatically from an ANOVA save structure if BLOCKSTRUCTURE is unset or if SAVE is set
FACTORIAL = *scalar*	Limit on the number of factors in treatment terms; default 3
PROBABILITY = *scalar*	Significance level at which the response is required to be detected (assuming a one-sided test); default 0.05
TMETHOD = *string token*	Type of test to be made (onesided, twosided, equivalence, noninferiority, fratio); default ones
XCONTRASTS = *variate*	X-variate defining a contrast to be detected
CONTRASTTYPE = *string token*	Type of contrast (regression, comparison) default rege
SAVE = *asave*	ANOVA save structure to provide the information about the design

Parameters

RESPONSE = *scalars*, *variates* or *tables*	Size of the difference or contrast between the effects of TERM that is to be detected, or (for TMETHOD=fratio) pattern of effects or means to be detected
RMS = *scalars*	Anticipated residual mean square corresponding to TERM; can be omitted if a SAVE structure is available
POWER = *scalars* or *variates*	Power (i.e. probability of detection) for RESPONSE

Description

When assessing an experimental design, it can be useful to know how likely a treatment response of a specified size may be detected. This probability of detection, known as the *power* of the design with respect to the response of interest, helps to determine whether the experiment is sufficiently large or accurate to achieve its purpose.

The treatment term to test is specified using the TERM option of APOWER, and the difference that you want to detect between its effects is given by the RESPONSE parameter. As an alternative to detecting a difference between treatment effects, you can ask to detect a contrast. However, here the treatment term must be a main effect (that is, TERM must involve just one factor). The XCONTRASTS option then species a variate containing the coefficients defining the contrast, and the CONTRASTTYPE option indicates whether this is a regression contrast (as specified by the REG function) or a comparison (as specified by COMPARISON).

The PROBABILITY option specifies the significance level that you will be using in the analysis to detect the treatment difference or contrast; the default is 0.05, i.e. 5%. By default, APOWER assumes that a one-sided t-test is to be used, but you can set option TMETHOD=twosided to take a two-sided t-test instead.

Other settings of TMETHOD enable you to test for equivalence or for non-inferiority. With equivalence (TMETHOD=equivalence), RESPONSE defines a threshold below which the treatments can be assumed to be equivalent. If the treatments have effects e_1 and e_2, the null hypothesis that the treatments are not equivalent is that either

$(e_1 - e_2) \leq -$RESPONSE

or

$(e_1 - e_2) \geq$ RESPONSE

with the alternative hypothesis that they are equivalent, i.e.

$-$RESPONSE $< (e_1 - e_2) <$ RESPONSE

(For further details see the Method information for procedure ASAMPLESIZE.) With non-inferiority (TMETHOD=noninferiority), RESPONSE again specifies the threshold for the effect of one treatment to be superior to another. So, for example, to demonstrate non-inferiority of treatment 1 compared to treatment 2, the null hypothesis becomes

$(e_1 - e_2) \geq -$RESPONSE

which represents a simple one-sided t-test.

You can also set TMETHOD=fratio, to assess the power of the F test in the analysis of variance table to detect a pattern of effects for TERM. You can specify the pattern by setting RESPONSE to a table containing the anticipated effects or means. Alternatively, you can set it to a y-variate containing, in each unit, the value of the effect or mean for the treatment (or treatment combination) to be applied to that unit of the design.

To determine the power, you need to define the design and specify the anticipated residual mean square for the stratum where the treatment term is estimated. This is most easily obtained by taking the analysis of a design with similar units and the same block and treatment structures as those that are to be used in the new design. To do this, you should analyse the earlier set of data with the ANOVA directive in the usual way. First define the strata (or error terms) for the design using the BLOCKSTRUCTURE directive, and the treatment model to be fitted using the TREATMENTSTRUCTURE directive. Then analyse the y-variate using the ANOVA directive. Provided you do not give any other ANOVA commands in the interim, APOWER will pick up the information automatically from the save information held within Genstat about the most recent ANOVA analysis. Alternatively, you can save the information explicitly in an ANOVA save structure, using the SAVE parameter of ANOVA, and then use this same save structure as the setting of the SAVE option of APOWER.

If you do not have a suitable earlier set of data, you should set up the design factors to contain the values required to define the units of the design. Then use the BLOCKSTRUCTURE and TREATMENTSTRUCTURE options of APOWER to define the strata and the treatment model, and the RMS option to specify the anticipated residual mean square for the stratum where TERM is estimated. There is also the compromise possibility that you can take the information about the design, the strata and treatment model from an ANOVA save structure (generated for example by the analysis of an artificial data set), but use the RMS parameter to specify a different residual mean square from the one in the analysis in the save structure. The treatment terms to be included are controlled by the FACTORIAL option; this sets a limit (by default 3) on the number of factors in a treatment term: terms containing more than that number are deleted.

The POWER parameter can save the power. This is printed by default, but you can set option PRINT=* to stop this.

Options: PRINT, TERM, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, FACTORIAL, PROBABILITY, TMETHOD, XCONTRASTS, CONTRASTTYPE, SAVE.

Parameters: RESPONSE, RMS, POWER.

Method

The standard error of difference between two treatment effects is

$$\sqrt{(s^2 \times 2 / (r \times e))}$$

where s^2 is the residual mean square of the stratum where the treatment term is estimated, e is the efficiency factor, and r is the replication of each effect. For a regression contrast the standard error is

$$\sqrt{(s^2 \times 2 / (r \times sdiv \times e))}$$

where sdiv is the sum of squares of the XCONTRASTS variate, and for a comparison contrast the standard error is

$$\sqrt{(s^2 \times sdiv / (r \times e))}$$

APOWER assumes that the treatment effects have equal replication. Unequal replication can be studied by defining a comparison between the effects. For example, to allow for a control level with two replicates, you could assume that the first two levels are for the control, and then study comparisons between their mean and the other levels.

See also

Directive: ANOVA.

Procedures: ADETECTION, ASAMPLESIZE, RPOWER, VPOWER.

Genstat Reference Manual 1 Summary sections on: Analysis of variance, Design of experiments.

APPEND

Appends a list of vectors of the compatible types (R.W. Payne).

Options

NEWVECTOR = *variate*, *factor* or *text*
: Vector to store the appended values; by default uses the first vector of the OLDVECTOR list

FREPRESENTATION = *string token*
: How to match the values of old factors (levels, labels, ordinals, renumbered); default leve

GROUPS = *factor*
: Factor to represent the OLDVECTOR to which each unit originally belonged

Parameter

OLDVECTOR = *variates*, *factors*, *texts* or *scalars*
: Values to be appended

Description

APPEND provides a convenient way of taking the values from several variates, factors, scalars or texts and appending (i.e. copying) them into a single variate, factor or text. The variates, factors scalars and texts whose values are to be appended are specified by the OLDVECTOR parameter, and the NEWVECTOR option supplies the variate, factor or text to store the appended values. If NEWVECTOR is omitted, the values are placed into the first OLDVECTOR (but it must not be a scalar). Also, the type of the NEWVECTOR is taken from the first OLDVECTOR, if it has not already been defined.

The NEWVECTOR will contain all the values of the first OLDVECTOR, then all those from the second, and so on. The old vectors can thus contain different numbers of values, but they must be of compatible types. Texts can receive values from any type of OLDVECTOR, with the values of variates, factors scalars first being formed into texts using the TXCONSTRUCT directive. However, variates cannot receive values from texts. Factors can receive values from any type, subject to the setting of the FREPRESENTATION option, described below. Variates, texts and scalars are first formed into factors, using the GROUPS directive, and the values are then transferred into the new factor. A factor formed from a text will therefore have both levels and labels, but those formed from variates or scalars will have only levels.

The FREPRESENTATION option indicates how the levels of factors are matched amongst the old factors. If this is set to labels and the levels of the old factors are compatible (that is if each label corresponds to the same level in all the old factors), then the level definitions are transferred to the new factor; if not, the levels are defined to be the default values 1, 2... and a warning is printed. Similarly, with the default setting FREPRESENTATION=levels, the labels are retained if they are compatible, but no warning is printed if they are not. For FREPRESENTATION=ordinals, the levels of all the factors are taken as the ordinal values 1, 2... (and no labels are defined). Finally, the renumbered setting assumes that the old factors all have independent sets of levels, and renumbers these from one upwards for the first factor, from number of levels of the first factor plus one upwards for the second factor, and so on; the new factor will thus have a different level for every level of the original factors.

The GROUPS option allows a factor to be formed indicating the OLDVECTOR to which each unit of the appended vector originally belonged. The levels are labelled by the identifier of the corresponding OLDVECTOR. This factor could be used in the CONDITION option of the SUBSET procedure subsequently to recover the values of the original vectors.

Options: NEWVECTOR, FREPRESENTATION, GROUPS.
Parameter: OLDVECTOR.

Action with `RESTRICT`
Any restrictions on the vectors are ignored.

See also
Directive: EQUATE.
Procedures: RESHAPE, STACK, VEQUATE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

APRODUCT

Forms a new experimental design from the product of two designs (R.W. Payne).

Options

PRINT = *string token*	Controls printing of the design (design); default desi
ANALYSE = *string token*	Whether to analyse the design by ANOVA (yes, no); default no
METHOD = *string token*	How to combine the designs (cross, nest); default nest
BF1 = *formula*	Block formula for design 1
TF1 = *formula*	Treatment formula for design 1
BF2 = *formula*	Block formula for design 2
TF2 = *formula*	Treatment formula for design 2

No parameters

Description

APRODUCT forms an experimental design by taking the product of two other designs. The METHOD option controls whether the product is formed by nesting the second design within the first, or by crossing the two designs together. For example, suppose that the first design has a single factor Units in the block structure and a single treatment factor A, while the second design is a Latin square with block structure Rows*Colums and treatment factor B. If we nest the second design within the first, we would obtain a design with block structure Units/(Rows*Columns) in which each unit of the first design has been subdivided into a row by column array of subplots to contain a Latin square of the sort defined by the second design. Nesting is thus useful when you want to subdivide the units of a design and apply further treatments (in this case those defined by the factor B) to the resulting subplots. Similarly, if we cross the two designs, the new design will have a block structure of Units*(Rows*Columns), or Units*Rows*Columns, in which we have duplicated the second design for every level of Units. Crossing is useful if you need to introduce a new blocking structure into an existing design. For example, the Units factor might represent different time periods or different locations in which the latin square design was to be used, and the factor A the different systematic conditions that might apply on each occasion.

With both nesting and crossing, the new design will contain a unit for every combination of the block factors in the two original designs, and so every combination of the treatment factors in the first design will occur with every combination of the treatment factors in the second design. The treatment structure is thus defined for the new design by crossing the treatment structures of the two original designs, to estimate all the original treatment terms and their interactions. So, in the example above, the treatment structure is defined to be A*B.

APRODUCT redefines the values of the factors as required for the new design, and executes BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives with the new block and treatment formulae. The new formulae can then be accessed, outside the procedure, using the GET directive or procedure ASTATUS. The PRINT option can be set to design to print the new design, and the ANALYSE option can be set to yes to produce a skeleton analysis of variance from ANOVA. Options BF1, TF1, BF2, and TF2 define the block structure and treatment structure of the first and then the second design.

Options: PRINT, ANALYSE, METHOD, BF1, TF1, BF2, TF2. Parameters: none.

Method

APRODUCT uses the standard Genstat manipulation directives such as FCLASSIFICATION,

CALCULATE and DUPLICATE. Procedure PDESIGN is used to print the design.

Action with RESTRICT
None of the factors must be restricted, and any existing restrictions will be cancelled.

See also
Procedure: AMERGE.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Calculations and manipulation.

ARANDOMIZE

Randomizes and prints an experimental design (R.W. Payne).

Options

PRINT = *string token*	Allows the (randomized) design to be printed; (design); default *
BLOCKSTRUCTURE = *formula*	Defines the block factors according to which the randomization is to be carried out; default takes the existing specification as defined by the BLOCKSTRUCTURE directive
EXCLUDE = *factors*	(Block) factors whose levels are not to be randomized
SEED = *scalar*	Seed to generate the random numbers used to define the randomization; default 0
LPERMUTE = *string token*	Whether to randomly permute treatment factor levels (no, yes); default no

Parameters

OLDVECTOR = *factors* or *variates*	Vectors whose values are to be randomized; default is to use the factors occurring in the formula (if any) specified by the most recent TREATMENTSTRUCTURE directive
NEWVECTOR = *factors* or *variates*	Vectors to store the randomized values; by default these overwrite the values in the original vectors

Description

ARANDOMIZE provides a convenient way of randomizing the treatment allocations in an experimental design. It has several advantages over the RANDOMIZE directive (which is used inside the procedure).

First of all, the BLOCKSTRUCTURE option, which (as in RANDOMIZE) specifies the block model formula to indicate how the randomization is to take place, will use any setting that has already been defined by the BLOCKSTRUCTURE directive as its default. Moreover, the formula need not index all the units of the design, as would be required by RANDOMIZE; if necessary ARANDOMIZE will set up an extra factor _units_ simular to the factor *units* used by ANOVA.

ARANDOMIZE allows the original (unrandomized) values to be retained. There are two parameters: OLDVECTOR to specify the factors or variates to be randomized, and NEWVECTOR to allow new structures to be supplied to store the randomized values. If no NEWVECTOR is specified, the randomized values replace the original values of the corresponding OLDVECTOR. By default, NEWVECTOR is assumed to contain the list of factors in the model formula (if any) specified by the previous TREATMENTSTRUCTURE directive.

The levels of the treatment factors can be randomized by setting option LPERMUTE=yes; ARANDOMIZE then randomly permutes the numbering of the levels of each treatment factor on the units of the design. There is also a PRINT option which can be set to design to print the design. The other two options, EXCLUDE and SEED, are as in RANDOMIZE. EXCLUDE lists block factors whose levels are not to be permuted during the randomization; for example the period factor might need to be excluded in the randomization of a trial to study carry over effects. SEED defines the seed used to generate the random numbers used for the randomization. The default of zero continues the existing sequence of random numbers if RANDOMIZE has already been used in the current Genstat job. If RANDOMIZE has not yet been used, Genstat picks a seed at random.

Options: PRINT, BLOCKSTRUCTURE, EXCLUDE, SEED, LPERMUTE.
Parameters: OLDVECTOR, NEWVECTOR.

Method
The GET directive is used to access any existing settings defined by the BLOCKSTRUCTURE or TREATMENTSTRUCTURE directives. AFUNITS, if necessary, forms the extra _units_ factor, and DUPLICATE generates new copies of the original vectors, if these are to be kept, before RANDOMIZE is used to produce the randomized values. Finally, if required, PDESIGN is used to print the design.

Action with RESTRICT
RESTRICT can be used, as usual, to restrict the set of units to be randomized.

See also
Directive: RANDOMIZE.
Procedures: APERMTEST, RPERMTEST.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

†ARCSPLITPLOT

Adds extra treatments onto the replicates of a resolvable row-column design, and generates factors giving the row and column locations of the plots within the design (R.W. Payne).

Options

PRINT = *strings*	Controls printed output (design, factors, layout); default * i.e. none
LEVELS = *scalar* or *variate*	Numbers of levels of the extra treatment factors; if unset, takes the numbers of levels declared for the TREATMENTFACTORS
TREATMENTFACTORS = *factors*	Extra treatment factors to be imposed onto the replicates of the original row-column design
REPLICATES = *factor*	Replicates in the modified design (after adding the extra treatments)
WHOLEPLOTS = *factor*	Whole-plots in the modified design
ROWS = *factor*	Factor indexing the rows over the whole design
COLUMNS = *factor*	Factor indexing the columns over the whole design
RCREPLICATES = *factor*	Replicates in the row-column design
RCROWS = *factor*	Rows within replicates of the row-column design
RCCOLUMNS = *factor*	Columns within replicates of the row-column design
REPLOCATIONS = *variate* or *matrix*	Locations of the replicates of the row-column design
METHOD = *string*	How to form the replicates of the modified design (rowserpentine, columnserpentine, given); default rows
SEED = *scalar*	Seed for randomizing the allocation of the extra treatments; default 0
SPREADSHEET = *string*	Whether to put the design factors into a spreadsheet (design); default *

No parameters

Description

ARCSPLITPLOT can be used to superimpose additional treatments onto the replicates of a row-column design, for example formed by CDNROWCOLUMNDESIGN, so that the design becomes a split-plot with the original replicates as the whole plots. It can also be used to generate row and column factors giving the locations of the plots of the row-column design within the entire experiment. (The row and column factors usually generated for the row-column design merely index the rows and columns within each replicate.)

The factor for the replicates of the row-column design is specified by the RCREPLICATES option. The RCROWS and RCCOLUMNS options specify the factors for the rows and the columns within those replicates.

The locations of the replicates within the whole design must be specified by the REPLOCATIONS option. This can supply a variate containing the numbers of replicates in adjacent columns of the design, like the REPLATINGROUPS option of the CDNROWCOLUMNDESIGN procedure. For example, setting REPLATINGROUPS=!(2,2,2) defines three columns of replicates, the first containing replicates 1 and 2, the second containing replicates 3 and 4, and the third column containing replicates 5 and 6. Alternatively, you can supply a matrix with each cell containing the number of the replicate at that location.

This provides the information needed to generate the factors to identify the row and column locations of the plots within the whole design. They can be saved using the ROWS and COLUMNS options, respectively.

To superimpose the extra treatments, ARCSPLITPLOT also needs to know how to define the replicate factor for the extended design. This is specified by the METHOD option, with settings:

rowserpentine	in a serpentine way e.g. left-to-right, then right-to-left, and so on;
columnserpentine	column-by-column in a serpentine way e.g. top-to-bottom, then bottom to top, and so on; or
given	defined by existing values of the REPLICATES factor.

The identifiers of extra treatment factors are specified by the TREATMENTFACTORS option. The LEVELS option can be used to define the numbers of levels of those factors, as a scalar if there is only one factor, or as a variate if there are several. The levels specified in the variate are assumed to be in the same order as the order in which the factors occur in the TREATMENTFACTORS list. LEVELS can be omitted if the factors have already been declared with the right numbers of levels. The REPLICATES option specifies the identifier of the replicate factor, and the WHOLEPLOTS option specifies the identifier of the whole-plot factor.

Printed output is controlled by the PRINT option, with settings:

design	to print the design,
factors	to print the factor values; and
layout	to print the values the replicate, row and column factors of the row-column design in the layout of the whole experiment.

The SEED option allows you to supply a seed for the random numbers used to randomize the allocation of the extra treatments. The default value of zero continues an existing sequence of random numbers if any have already been used in the current Genstat job, or obtains a random seed using the system clock if none have been used already. You can also set SEED=-1 if you want to suppress any randomization.

You can set option SPREADSHEET=design to put the factors into a spreadsheet.

Options: PRINT, LEVELS, TREATMENTFACTORS, REPLICATES, WHOLEPLOTS, ROWS, COLUMNS, RCREPLICATES, RCROWS, RCCOLUMNS, REPLOCATIONS, METHOD, SEED, SPREADSHEET.
Parameters: none.

See also

Procedure: CDNROWCOLUMNDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

AREPMEASURES

Produces an analysis of variance for repeated measurements (R.W. Payne).

Options

PRINT = *string tokens*	Controls output about the covariance structure (vcovariance, correlation, epsilon, test); default epsi, test
APRINT = *string tokens*	Printed output from the analysis of variance (as for the ANOVA PRINT option); default *
TREATMENTSTRUCTURE = *formula*	Defines the treatments given to the subjects; if this is not set, the default is taken from any existing setting defined by the TREATMENTSTRUCTURE directive
BLOCKSTRUCTURE = *formula*	Defines any block structure over the subjects if this is not set, the default is taken from any existing setting defined by the BLOCKSTRUCTURE directive
COVARIATE = *variates*	Specifies any covariates on the subjects if this is not set, the default is taken from any existing setting defined by the COVARIATE directive
FACTORIAL = *scalar*	Limit in the number of factors in the terms generated from the TREATMENTSTRUCTURE formula
TIMEPOINTS = *variate*, *text* or *factor*	When the DATA parameter supplies a separate variate of observations for each time this can specify numbers or labels for the time points, when there is a single DATA variate this must supply a factor to indicate the time of each observation
CONTRASTS = *scalar*	Limit on the order of a contrast of a treatment term; default 4
DEVIATIONS = *scalar*	Limit on the number of factors in a treatment term for the deviations from its fitted contrasts to be retained in the model; default 9
FPROBABILITY = *string token*	Printing of probabilities for variance ratios in the aov table (no, yes); default no
PSE = *string tokens*	Standard errors to be printed with tables of means (differences, lsd, means); default diff
MAXCYCLE = *scalar*	Maximum number of iterations for estimating missing values; default 20
LSDLEVEL = *scalar*	Significance level (%) to use in the calculation of least significant differences; default 5
EPSILON = *scalar*	Saves the correction factor epsilon
SAVEFACTORS = *pointer*	Saves the factors used in the analysis of variance
ASAVE = *identifier*	Saves the ANOVA save structure from the analysis of variance

Parameter

DATA = *variates*	Data observations either in a list of variates (one for each time), or a single variate (with TIMEPOINTS set to a factor indicating the time of each observation)

Description

A repeated-measures design is one in which subjects (animals, people, plots, etc) are observed several times. Each subject receives a randomly allocated treatment, either at the outset, or repeatedly through the experiment. The subjects are observed at successive occasions to see how the treatment effects develop.

The design might thus seem analogous to a split-plot design, with subjects corresponding to whole plots, and the occasions of observation to the sub-plots. There are, however, some important differences between the two situations. With repeated measurements, there is likely to be a greater correlation between observations that are made at adjacent time points than between those that are more greatly spaced. Furthermore, the `Times` factor cannot, by its very nature, be allocated at random to the occasions within subjects. In the customary split-plot situation we can usually assume that there is an equal correlation between the sub-plots of each whole plot and, even if this were not so, the sub-plot treatment should have been allocated at random to the sub-plots within each whole plot. The formal conditions for the validity of the split-plot analysis will be discussed in more detail below, together with advice on how to proceed if they do not hold.

It is worth pointing out first, though, that this problem affects only the `Subjects.Times` stratum. The `Subjects` stratum contains an analysis of variance of the measurements totalled over the subjects, and this part of the analysis will be valid whatever the within-subject correlation structure. A further point is that, when measurements are taken on only two occasions, the analysis in the `Subjects.Times` stratum will also be valid; there can then be only one within-subject correlation, and the analysis in the `Subjects.Times` stratum is of the difference between the observations at time 2 and time 1 on each subject.

Another potential problem arising from the systematic nature of the `Times` factor is that effects arising from the "length of treatment time" will be confounded with any effects arising from the duration of the experiment, such as age of subject (which may be important with short-lived material such as aphids), season of year, time of day, and so on. This does not affect the validity of the analysis, and some of the confusion may be capable of being unravelled by running the experiment during more than one period. Nevertheless, care needs to be taken in drawing conclusions about time-effects.

The `Subjects.Times` information, describing the way in which the treatment effects change differentially with time, is often the aspect of most interest in the study. The formal requirement for the validity of the analysis in the sub-plot stratum of a split-plot design is that all the normalised contrasts in that stratum have an equal variance. The only practical arrangement of covariances between times that satisfies this condition would have a single variance down the diagonal and a single covariance off-diagonal. This pattern is known as a uniform covariance structure or, equivalently, the matrix is said to show compound symmetry; Box (1950) describes how this can be tested. In the usual split-plot analysis, the `Subjects.Times` sum of squares is assumed to be distributed as $\sigma^2 \times \chi^2_r$ where σ^2 is a constant and χ^2_r has a chi-square distribution on r degrees of freedom. Similarly, under the assumption that there is no `Treatments.Times` interaction, the `Treatments.Times` sum of squares is assumed to be distributed as $\sigma^2 \times \chi^2_t$ where χ^2_t has a chi-square distribution on t degrees of freedom. If the variance-covariance structure does not exhibit compound symmetry, it is possible to show that the distributions can still be approximated by chi-square distributions, but the degrees of freedom are instead epsilon $\times r$ and epsilon $\times t$. The correction factor epsilon lies between one, which would give the ordinary split-plot analysis, and 1/(number of times minus one), which would leave just one degree of freedom within each subject (remember that when there are only two observation on each subject, and thus just one within-subject degree of freedom, the analysis is valid). Epsilon can be estimated by maximum likelihood, as described by Greenhouse & Geisser (1959), and the estimated value can be saved by the `EPSILON` option. A further point is that this correction applies to the calculation of least significant differences as well as to the F ratios in the analysis

of variance table. So, instead of a t distribution on r degrees of freedom, these must use the square root of an F distribution on epsilon and epsilon × r degrees of freedom.

The printing of information about the covariances is controlled by the strings listed for the PRINT option: vcovariance variance-covariance matrix, correlation correlation matrix, epsilon Greenhouse-Geisser epsilon, test test for compound symmetry.

The output from the analysis of variance is controlled by the APRINT option, with settings identical to those in the PRINT option of the ANOVA directive. The FPROBABILITY, PSE, MAXCYCLE and LSDLEVEL options also operate exactly as in ANOVA.

The treatments applied to the subjects can be specified (as a model formula) using the TREATMENTSTRUCTURE option, the block structure (if any) on the subjects can be specified by the BLOCKSTRUCTURE option, and the COVARIATE option can be used to list any covariates. If any of these options is unset, the default is taken from any existing setting defined by the directives TREATMENTSTRUCTURE, BLOCKSTRUCTURE or COVARIATE, respectively. The FACTORIAL option can be used to set a limit on the number of factors in the terms generated from the TREATMENTSTRUCTURE option.

Contrasts can be specified by using the functions POL, REG, COMPARISON, POLND or REGND in the TREATMENTSTRUCTURE formula, as in ANOVA. The CONTRASTS option places a limit on the order of contrasts that are fitted. The DEVIATIONS option sets a limit on the number of factors in the terms whose deviations from the fitted contrasts are to be retained in the model. See ANOVA for more details.

The observed data are specified by the DATA parameter in one of two ways. The first is to supply a list of variates, each one containing the measurements made on the subjects at one of the successive occasions on which they were observed. The TIMEPOINTS option can then supply a variate or text to define numbers or labels to use in output to identify the time point corresponding to each DATA variate; if TIMEPOINTS is unset, the labels are formed automatically from the identifiers of the DATA variates themselves. The DATA variates are appended into a single variate for the analysis, and the block and treatment factors are expanded to match. You can specify a pointer using the SAVEFACTORS option to save the expanded factors. The elements of the pointer are labelled by the factor names, and the time factor is also included, with the label 'Time factor'. You would need to use these, for example, if you wanted to plot the means using AGRAPH.

The second possibility is to supply a single DATA variate containing the data from all the times. The TIMEPOINTS option must then be set to a factor indicating the time of each observation. The block and treatment factors must already have been expanded to match the DATA variate, and each subject should be represented by a unique combination of the block factors. If not, Genstat prints a warning and assumes that the subjects occur in the same order within each time. To simplify the use of AREPMEASURES in general programs, the SAVEFACTORS pointer is also formed when the data are in a single variate. (However, it then contains the original factors.)

The ASAVE option allows you to save the save structure from the ANOVA analysis.

Options: PRINT, APRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE, FACTORIAL, TIMEPOINTS, CONTRASTS, DEVIATIONS, FPROBABILITY, PSE, MAXCYCLE, LSDLEVEL, EPSILON, SAVEFACTORS, ASAVE.
Parameter: DATA.

Method

The procedure uses the standard Genstat directives for calculations and manipulation to obtain the various matrices and tests. Formulae for these are given by Box (1950), Greenhouse & Geisser (1959) and Winer (1962) pages 523 and 594-599, although note that equation (1) on page 595 should contain N' & n'_i, not N & n_i.

Action with `RESTRICT`
The procedure does not allow for restrictions, and will cancel any that have been applied.

References
Box, G.E.P. (1950). Problems in the analysis of growth and wear curves. *Biometrics*, **6**, 362-389.
Greenhouse, S.W. & Geisser, S. (1959). On methods in the analysis of profile data. *Psychometrika*, **24**, 95-112.
Winer, B.J. (1962). *Statistical Principals in Experimental Design (second edition)*. McGraw-Hill, New York.

See also
Directive: `ANOVA`.
Genstat Reference Manual 1 Summary section on: Repeated measurements.

ARESULTSUMMARY

Provides a summary of results from an ANOVA analysis (R.W. Payne).

Options

PRINT = *string tokens*	What to print (description, means, significant); default desc, mean, sign
PSE = *string tokens*	Standard errors to be printed with the means (sed, sedsummary, lsd, lsdsummary, dfmeans); default sed, dfme
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
SAVE = *ANOVA save structure*	Save structure for the analysis; default uses the save structure from the most recent ANOVA

No parameters

Description

ARESULTSUMMARY investigates an ANOVA analysis, to provide the information that would be useful for a report. By default, all the information is printed, but you can control this with the PRINT option, whose settings are:

description	prints the name of the y-variate, any covariates and the block and treatment models,
means	prints relevant tables of means, and
significant	lists the significant treatment terms.

The relevant tables of means are those that contain significant treatment effects. Also, each table contains all the significant effects involving any of its factors. In the example for the procedure, terms A, D, S and A.S are significant. Two tables of means are therefore presented, one classified by A and S, and the other by D. However, if the significant terms were A.S and D.S. there would be only one table, classified by factors A, D and S.

The PSE option controls the information provided with the tables of means:

sed	standard errors for differences between means,
sedsummary	summary of the standard errors for differences,
dfmeans	degrees of freedom for the standard errors of differences,
lsd	least significant differences between the means, and
lsdsummary	summary of the least significant differences.

The default is to print the standard errors of differences and their degrees of freedom. Note: if all the differences between means have the same standard error of difference, a summary is printed for the settings sed and lsd, instead of the full symmetric matrices of values.

The LSDLEVEL option specifies the significance level (%) to use in the calculation of least significant differences (default 5%).

Options: PRINT, PSE, LSDLEVEL, SAVE.
Parameters: none.

See also

Directives: ADISPLAY, ANOVA.
Procedure: AFMEANS, A2RESULTSUMMARY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ARETRIEVE

Retrieves an ANOVA save structure from an external file (R.W. Payne).

No options

Parameters

FILENAME = *texts*	Name of the file storing the save structure
EXIT = *scalars*	Scalar that contains the value one if the save structure could not be retrieved successfully, otherwise zero
SAVE = *asave* structures	Save structure that has been retrieved

Description

ARETRIEVE retrieves an ANOVA save structure, stored earlier by the ASTORE procedure in an external file. It can then be used to produce further output from the analysis. (See, for example, directives ADISPLAY and AKEEP, or procedures ACHECK, AGRAPH, APLOT, APERMTEST and ASPREADSHEET.)

The name (and path) of the file that stores the save structure is specified, in a text, by the FILENAME parameter. The save structure is saved by the SAVE parameter. The EXIT parameter can return a scalar containing the value one if the save structure could not be retrieved successfully. Otherwise it contains zero.

Options: none.
Parameters: FILENAME, EXIT, SAVE.

Method

ASTORE stores the save structure in a Genstat backing-store file using the STORE directive, and ARETRIEVE retrieves it using the RETRIEVE directive.

See also

Directive: ANOVA.
Procedure: ASTORE.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ASAMPLESIZE

Finds the replication to detect a treatment effect or contrast (R.W. Payne & P. Brain).

Options

PRINT = *string tokens*	Prints the replication or produces a printed summary of the power etc for the various amounts of replication (power, replication); default powe, repl
TERM = *formula*	Treatment term to be assessed in the analysis
REPLICATES = *factor*	Factor identifying the replication in the design
MINREPLICATION = *scalar*	Minimum number of replicates to try; default 2
MAXREPLICATION = *scalar*	Maximum feasible number of replicates; default * i.e. no limit
TREATMENTSTRUCTURE = *formula*	Treatment structure of the design; determined automatically from an ANOVA save structure if TREATMENTSTRUCTURE is unset or if SAVE is set
BLOCKSTRUCTURE = *formula*	Block structure of the design; determined automatically from an ANOVA save structure if BLOCKSTRUCTURE is unset or if SAVE is set
COMPONENTS = *variate* or *scalar*	Variate of variance components of all the terms in the block structure or, if TERM is estimated in the final stratum of the design, scalar containing only the variance component of the final stratum itself; determined automatically (if possible) from an ANOVA save structure if unset
FACTORIAL = *scalar*	Limit on the number of factors in treatment terms; default 3
PROBABILITY = *scalar*	Significance level at which the response is required to be detected (assuming a one-sided test); default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Type of test to be made (onesided, twosided, equivalence, noninferiority, fratio); default ones
XCONTRASTS = *variate*	X-variate defining a a contrast to be detected
CONTRASTTYPE = *string token*	Type of contrast (regression, comparison) default rege
SAVE = *asave*	ANOVA save structure to provide the information about the design

Parameters

RESPONSE = *scalars*	Size of the difference or contrast between TERM effects that is to be detected
NREPLICATES = *scalars*	Number of replicates required to detect RESPONSE

Description

When designing an experiment, it is often possible to vary the replication of the treatments. For example, in a randomized block design you can adjust the number of blocks, or in a design with no blocking structure you can choose how many units to allocate to each of the treatments.

To decide how many replicates to include, you need to specify the size of difference between treatment effects that you would like the design to be able to detect. The treatment term of interest is specified using the TERM option of ASAMPLESIZE, and the difference that you want

to detect between its effects is given by the RESPONSE parameter. As an alternative to detecting a difference between treatment effects, you can ask to detect a contrast, but here the treatment term must be a main effect (that is, TERM must involve just one factor). The XCONTRASTS option then specifies a variate containing the coefficients defining the contrast, and the CONTRASTTYPE option indicates whether this is a regression contrast (as specified by the REG function) or a comparison (as specified by COMPARISON).

The PROBABILITY option specifies the significance level that you will be using in the future analysis to detect the treatment difference (default 0.05, i.e. 5%). The POWER option specifies the probability with which you want the experiment to be able to detect the difference (that is, the *power* of the test); by default this is 0.9 i.e. 90%. In the language of hypothesis testing, PROBABILITY specifies the type I error rate, and POWER specifies one minus the type II error rate. By default, ASAMPLESIZE assumes a one-sided t-test is to be used, but you can set option TMETHOD=twosided to take a two-sided t-test instead. Alternatively, if you set TMETHOD=fratio, ASAMPLESIZE takes RESPONSE as the maximum difference between the effects of TERM, and uses
an F-test.

Other settings of TMETHOD enable you to test for equivalence or for non-inferiority. With equivalence (TMETHOD=equivalence), RESPONSE provides a threshold below which the treatments can be assumed to be equivalent. If the treatments have effects e_1 and e_2, the null hypothesis that the treatments are not equivalent is that either

$(e_1 - e_2) \leq -$RESPONSE

or

$(e_1 - e_2) \geq$ RESPONSE

with the alternative hypothesis that they are equivalent, i.e.

$-$RESPONSE $< (e_1 - e_2) <$ RESPONSE

With non-inferiority (TMETHOD=noninferiority), RESPONSE again specifies the threshold for the effect of one treatment to be superior to another. So, for example, to demonstrate non-inferiority of treatment 1 compared to treatment 2, the null hypothesis becomes

$(e_1 - e_2) \geq -$RESPONSE

(which, in fact, represents a simple one-sided t-test).

To determine the replication, ASAMPLESIZE needs to know the about the structure of the design, and the likely amount of variability. This is most easily obtained by taking the analysis of a design with similar units and the same block and treatment structures as those that are to be used in the new design. To do this, you should analyse the earlier set of data with the ANOVA directive in the usual way. First define the strata (or error terms) for the design using the BLOCKSTRUCTURE directive, and the treatment model to be fitted using the TREATMENTSTRUCTURE directive. Then analyse the y-variate using the ANOVA. Provided you do not give any other ANOVA commands in the interim, ASAMPLESIZE will pick up the information automatically from the save information held within Genstat about that analysis. Alternatively, you can save the information explicitly in an ANOVA save structure, using the SAVE parameter of ANOVA, and then use this same save structure as the setting of the SAVE option of ASAMPLESIZE.

If you do not have a suitable earlier set of data, you should set up the design factors to contain the values required to define the units of the design for any convenient number of replicates. (It does not matter how many replicates you choose, as the form of the design should be the same in every replicate.) Then use the TREATMENTSTRUCTURE and BLOCKSTRUCTURE options of ASAMPLESIZE to define the treatment model and the block model, and the COMPONENTS option to specify the variance components of the strata. Note: if TERM is estimated in the bottom (or final) stratum of the design, COMPONENTS can be set to a scalar to specify only the variance component of this stratum – which is then equal to its residual mean square.

There is also the compromise possibility that you can take the information about the design

and the block and treatment model from an ANOVA save structure (generated for example by the analysis of an artificial data set), but use the COMPONENTS option to specify different variance components from those in the analysis in the save structure.

The treatment terms to be included are controlled by the FACTORIAL option. This sets a limit (by default 3) on the number of factors in a treatment term. Treatment terms containing more than that number are deleted.

Finally, you must set the REPLICATES option to the factor in the block formula whose number of levels is to be increased or decreased to change the replication of the treatments. You can set the MINREPLICATION option to indicate the minimum number of replicates to try; by default this is 2. You can use the MAXREPLICATION option to define a maximum feasible number of replicates; by default this is no limit. The number of replicates that is required can be saved using the NREPLICATES parameter.

The PRINT option controls the printed output, with settings:

power	prints a table summarising the situation for a range of numbers of replicates (defined by MINREPLICATION and MAXREPLICATION if set, otherwise set automatically to a range covering the required number of replicates) – the table contains the residual degrees of freedom, the residual mean square, the standard error of difference (sed), RESPONSE divided by the sed, the t-value for a difference of RESPONSE, and the detection probability (i.e. power) at the level defined by the PROBABILITY option;
replication	prints the required replication.

By default both are printed.

For example, the following program would determine the number of blocks required to detect a treatment difference of 3 in a randomized block design with an anticipated residual mean square of 2.5 in the final stratum Block.Plot (i.e. within blocks); there is a single treatment factor Treat with 3 levels. We first use AGHIERARCHICAL to define the design for one replicate (or block), and then call ASAMPLESIZE to discover how many blocks are actually needed.

```
AGHIERARCHICAL [PRINT=*; ANALYSE=no; SEED=-1] Block,Plot;\
               TREATMENTFACTORS=*,Treat; LEVELS=1,3
ASAMPLESIZE    [PRINT=power,rep; TERM=Treat;\
               REPLICATES=Block; TREATMENTSTRUCTURE=Treat;\
               BLOCKSTRUCTURE=Block/Plot; COMPONENT=2.5]\
               1; NREPLICATES=Nrep
```

As another example, suppose we wish to have a split-plot design, with block structure Rep/Wplot/Subplot. The factor Variety with 3 levels is applied to whole plots (and is thus estimated in the Rep.Wplot stratum) and the factor Nitrogen with 4 levels is applied to the sub-plots (and is thus estimated in the Rep.Wplot.Subplot stratum). The variance components for Rep, Rep.Wplot and Rep.Wplot.Subplot are anticipated to be 6, 3 and 5 respectively, and we wish to detect varietal differences of 3. Again we first define a split-plot with a single replicate, and then use ASAMPLESIZE to find out how many reps we need.

```
AGHIERARCHICAL [PRINT=*; ANALYSE=no; SEED=-1]\
               Rep,Wplot,Subplot;\
               TREATMENTFACTORS=*,Variety,Nitrogen;\
               LEVELS=1,3,4
ASAMPLESIZE    [PRINT=power,rep; TERM=Variety;\
               REPLICATES=Rep;\
               TREATMENTSTRUCTURE=Variety*Nitrogen;\
               BLOCKSTRUCTURE=Rep/Wplot/Subplot;\
               COMPONENTS=!(6,3,5)] 3; NREPLICATES=Nrep
```

Options: PRINT, TERM, REPLICATES, MINREPLICATION, MAXREPLICATION,

TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COMPONENTS, FACTORIAL, PROBABILITY, POWER, TMETHOD, XCONTRASTS, CONTRASTTYPE, SAVE.
Parameters: RESPONSE, NREPLICATES.

Method

The standard error of difference between two treatment effects is

$$\sqrt{(s^2 \times 2 / (r \times e))}$$

where s^2 is the stratum variance of the stratum where the treatment term is estimated, e is the efficiency factor, and r is the replication of each effect. For a regression contrast the standard error is

$$\sqrt{(s^2 \times 2 / (r \times sdiv \times e))}$$

where *sdiv* is the sum of squares of the XCONTRASTS variate, and for a comparison contrast the standard error is

$$\sqrt{(s^2 \times sdiv / (r \times e))}$$

ASAMPLESIZE assumes that the treatment effects have equal replication, and also that all the effects (or residuals) of each block term have equal replication.

The stratum variance can be calculated as the variance component of the stratum S where the treatment term is estimated multiplied by the replication of its effects (residuals), plus the variance component of each stratum to which the stratum S is marginal, again multiplied by the replication of its effects (residuals). See for example Payne & Tobias (1992).

Comparing the null hypothesis that the treatments are not equivalent, i.e.

$$(m_1 - m_2) \leq -d$$

or

$$(m_1 - m_2) \geq d$$

with the alternative hypothesis that they are equivalent, i.e.

$$-d < (m_1 - m_2) < d$$

defines an *intersection-union* test, in which each component of the null hypothesis must be rejected separately. Here this implies performing two one-sided t-tests (this is known as a *TOST* procedure). If the significance level for the full test is to be α, each t-test must have significance level α (see Berger & Hsu 1996). To obtain a detection probability (or power) of (1 − β), each of the t-tests must have detection probabilities of (1 − β/2).

To demonstrate non-inferiority of treatment 1 compared to treatment 2, the null hypothesis is

$$(m_1 - m_2) \geq -d$$

This is equivalent to a one-sided t-test.

For the F-test, it is assumed that one effect will be −0.5 × RESPONSE, another will be 0.5 × RESPONSE, and the others will be zero. This gives the smallest sum of squares for any table of effects with a maximum pair-wise difference of RESPONSE, which represents the most difficult case that needs to be detected.

References

Berger, M.L. & Hsu, J.C. (1996). Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Statistical Science*, **11**, 283-319.

Payne, R.W. & Tobias, R.D. (1992). General balance, combination of information and the analysis of covariance. *Scandinavian Journal of Statistics*, **19**, 3-23.

See also

Directive: ANOVA.
Procedures: ADETECTION, APOWER, STTEST, VSAMPLESIZE.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

ASCREEN

Performs screening tests for designs with orthogonal block structure (R.W. Payne).

Options

PRINT = *string tokens*	Which tests to print (conditional, marginal, efficiency); default cond, marg
FACTORIAL = *scalar*	Limit on the number of factors in each treatment term; default 3
EXCLUDEHIGHER = *string token*	Whether to exclude higher-order interactions in the initial model for the conditional test of each term (yes, no); default no
FORCED = *formula*	Terms that must be included (together with any covariates) in the initial models for every term; default * i.e. none

Parameter

Y = *variates*	Variates to be analysed

Description

ASCREEN can be used to assess the treatment terms in an analysis of variance when the design is unbalanced but its error terms that are all orthogonal to one another. This includes any design with a hierarchical block structure, for example

```
Blocks / Plots
```

or

```
Replicates / Wholeplots / Subplots
```

ASCREEN thus provides a way of testing treatment terms in designs that cannot be analysed by ANOVA. Once ASCREEN has been used to decided which terms need to be included in the treatment model, the treatment effects and means can be estimated using REML.

Before using ASCREEN, the block and treatment models for the design must be defined by the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives, in exactly the same way as for an analysis by ANOVA. As in ANOVA, the FACTORIAL option sets a limit on number of factors in each treatment term (default 3). You can also define covariates using the COVARIATE directive. The y-variate is specified by the Y parameter of ASCREEN.

ASCREEN forms marginal and conditional tests for the treatment terms similar to those produced by the RSCREEN procedure. These are produced for the analysis of each stratum of the design (i.e. for the variation associated with each error term).

In a marginal test, each term is assessed by adding it to the simplest possible model. So, with a treatment model of

```
A + B + C + D + A.B + A.C + A.D + B.C + C.D + A.B.C + A.B.D +
A.C.D + B.C.D + A.B.C.D
```

the main effect of A is added it to the null model, while the interaction term A.B is added to a model containing only the main effects of A and B.

In a conditional test, each term is added to the most complex possible model. So the main effect A is added to an initial model excluding any term that has A as one of its margins. A is a margin of any term that contains A as one of its factors. So the terms to exclude for A are A.B, A.C, A.D, A.B.C, A.B.D, A.C.D and A.B.C.D. Similarly the interaction A.B is added to a model excluding any term that has A.B as a margin; i.e. any term that contains A and B amongst its factors. So A.B.C, A.B.D and A.B.C.D are excluded with A.B. The other terms to be included in the initial model depend on the setting of the EXCLUDEHIGHER option. With the default setting of no, all other terms are included in the initial model. So, the initial model for

A would be

```
B + C + D + B.C + C.D + B.C.D
```

Alternatively, if EXCLUDEHIGHER=yes, the initial model contains only terms with no more factors than the term being tested. So, the initial model for A would be

```
B + C + D
```

The FORCED option allows you to specify a model formula with terms that must be included in the initial model for the conditional and marginal tests of every treatment term. The forced model automatically includes any covariates.

The PRINT option controls printed output. The settings marginal and conditional control which tests are produced if there is more than one stratum (or error term); by default both types of test are printed. However, if there is only one error term, ASCREEN uses procedure RSCREEN, which always prints both. There is also a setting, efficiency, which prints the minimum, maximum and harmonic mean efficiency factor of the terms in each of the strata if there is more than one. These efficiency factors show the amount of information available to construct the marginal test for each of the terms in the strata where it can be estimated. The harmonic mean is presented, rather than an ordinary average, as this corresponds to the average variance of differences amongst the effects of the term (remember that the variance is proportional to the reciprocal of the efficiency factor).

Options: PRINT, FACTORIAL, EXCLUDEHIGHER, FORCED.
Parameter: Y.

Method

ASCREEN uses RSCREEN if there is only one error term. Otherwise, it first uses ANOVA to check that the design has orthogonal block structure. Then, if so, it calculates the relevant sums of squares by regression with matrices of weights calculated using FPROJECTIONMATRIX. The weight matrix for each stratum is its projection matrix; for further details see Payne & Tobias (1992).

Action with RESTRICT

ASCREEN takes account of any restrictions on the y-variate.

Reference

Payne, R.W. & Tobias, R.D. (1992). General balance, combination of information and the analysis of covariance. *Scandinavian Journal of Statistics*, **19**, 3-23.

See also

Directives: ANOVA, REML.
Procedures: RSCREEN, VSCREEN.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, REML analysis of linear mixed models.

ASPREADSHEET

Saves results from an analysis of variance in a spreadsheet (R.W. Payne).

Options

MEANS = *pointer*	Pointer to tables to contain the treatment means; default means
SEMEANS = *pointer*	Pointer to tables to contain the effective standard errors of treatment means; default ese
SEDMEANS = *pointer*	Pointer to matrices to contain standard errors of differences of treatment means; default sed
EFFECTS = *pointer*	Pointer to tables to contain the treatment effects; default effects
REPLICATIONS = *pointer*	Pointer to tables of treatment replications; default replication
RESIDUALS = *variate*	Variate to save the residuals in the fittedvalues page; default residuals
FITTEDVALUES = *variate*	Variate to save the fitted values in the fittedvalues page; default fittedvalues
AOVTABLE = *pointer*	Pointer to a text and variates containing the information in the analysis-of-variance table; default aovtable
COVINFORMATION = *pointer*	Pointer to a text and variates containing the information about the estimated covariate regression coefficients; default cov
MVINFORMATION = *pointer*	Pointer to a text and variates containing the information the about estimated missing values; default missing
EQFACTORS = *factors*	Factors whose levels are to be assumed to be equal within the comparisons between means, when calculating effective standard errors
RMETHOD = *string token*	Type of residuals to form (simple, standardized); default simp
†LSDMEANS = *pointer*	Pointer to matrices to contain least significant differences for means
†LSDLEVEL = *scalar*	Significance level (as a percentage) for the least significant differences; default 5
†SPREADSHEET = *string tokens*	What to include in the spreadsheet (aovtable, covariates, effects, means, semeans, sedmeans, lsdmeans, replications, fittedvalues, missingvalues); default aovt, cova, mean, sedm, repl, fitt, miss
OUTFILENAME = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create
SAVE = *ANOVA save structure*	Specifies which analysis to save; default * i.e. most recent one

No parameters

Description

ASPREADSHEET puts results from an analysis of variance into a spreadsheet. By default the results are from the most recent ANOVA, but you use the SAVE option to specify the save structure from some other analysis.

The SPREADSHEET option specifies which pages of the spreadsheet to form, with settings:

aovtable	analysis of variance table,

`covariates`	estimated covariate regression coefficients and their standard errors (if any covariates in the analysis),
`effects`	tables of treatment effects,
`means`	tables of treatment means,
`semeans`	tables of effective standard errors of treatment means,
`sedmeans`	matrices of standard errors of differences of treatment means,
`lsdmeans`	matrices of least significant differences of treatment means,
`replications`	replication tables of treatment terms,
`fittedvalues`	y-variate, fitted values and residuals,
`missingvalues`	estimates for missing values (if any).

By default, `SPREADSHEET = aovt, cova, mean, sedm, repl, fitt, miss`.

To help avoid clashes between the columns of the spreadsheets if you want to save results from more than one analysis, the parameters `MEANS`, `SEMEANS`, `SEDMEANS`, `LSDMEANS`, `EFFECTS`, `REPLICATIONS`, `RESIDUALS`, `FITTEDVALUES`, `AOVTABLE`, `COVINFORMATION` and `MVINFORMATION` allow you to specify identifiers for the columns (or sets of columns) that will store the corresponding results in the current spreadsheet.

The `EQFACTORS` option allows you to specify factors within the tables of means whose levels are assumed to be equal for the two means, when calculating effective standard errors.

The `RMETHOD` option controls whether the residuals are simple residuals (like those printed by `ANOVA` – the default) or whether they are standardized according to their variances.

The `LSDLEVEL` option specifies the significance level (as a percentage) for the least significant differences; default 5.

You can save the data in either a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx), by setting the `OUTFILENAME` option to the name of the file to create. If the name is specified without a suffix, `'.gwb'` is added (so that a Genstat workbook is saved). If `OUTFILENAME` is not specified, the data are put into a spreadsheet opened inside Genstat.

Options: `MEANS`, `SEMEANS`, `SEDMEANS`, `EFFECTS`, `REPLICATIONS`, `RESIDUALS`, `FITTEDVALUES`, `AOVTABLE`, `COVINFORMATION`, `MVINFORMATION`, `EQFACTORS`, `RMETHOD`, `LSDMEANS`, `LSDLEVEL`, `SPREADSHEET`, `OUTFILENAME`, `SAVE`.
Parameters: none.

Action with `RESTRICT`
If the `Y` variate is restricted, that restriction will carry over into the fitted-values spreadsheet.

See also
Directive: `SPLOAD`.
Procedures: `ADSPREADSHEET`, `AUSPREADSHEET`, `RSPREADSHEET`, `VSPREADSHEET`, `FSPREADSHEET`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ASTATUS

Provides information about the settings of ANOVA models and variates (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (y, model, weights); default mode

Parameters

Y = *pointers*	Pointer of length 1 to save the identifier of the y-variate of the most recent ANOVA or that used to form INSAVE
TREATMENTSTRUCTURE = *formula structures*	Saves the current setting of TREATMENTSTRUCTURE or the setting used to form INSAVE
BLOCKSTRUCTURE = *formula structures*	Saves the current setting of BLOCKSTRUCTURE or the setting used to form INSAVE
COVARIATE = *pointers*	Saves the current COVARIATE setting or the setting used to form INSAVE
DESIGN = *pointers*	Pointer of length 1 to save the design structure in the most recent ANOVA or the one used to form INSAVE
WEIGHTS = *pointers*	Pointer of length 1 to save the identifier of the variate of weights (if any) in the most recent ANOVA or that used to form INSAVE
SAVE = *asave* structures	Saves the save structure from the most recent ANOVA
INSAVE = *asave* structures	Provides a save structure from which to save Y, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE and WEIGHTS; default * uses the current settings

Description

ASTATUS allows information to be printed and saved about the model settings and other information involved in an ANOVA analysis.

By default ASTATUS prints the current settings defined by the directives TREATMENTSTRUCTURE, BLOCKSTRUCTURE and COVARIATE. This is governed by the default setting, model, of the PRINT option. The y setting prints the name of the y-variate from the most recent ANOVA, and the weights setting prints the identifier of the variate of weights (if any). Alternatively, if the INSAVE parameter is set to the save structure from an ANOVA analysis, the y-variate, weights and model settings will be those used to form the save structure.

If the INSAVE parameter is not set, the Y parameter can be used to save the identifier of the y-variate most recently analysed by ANOVA, in a pointer of length one. The TREATMENTSTRUCTURE parameter saves the current setting defined by the TREATMENTSTRUCTURE directive (in a formula structure), and the BLOCKSTRUCTURE parameter similarly saves the current setting defined by the BLOCKSTRUCTURE directive. The COVARIATE parameter saves the current setting defined by the COVARIATE directive (in a pointer). The DESIGN parameter can save the design structure, which contains the information for the analysis, in a pointer of length one. Finally, the WEIGHTS parameter can save the identifier of the variate of weights in the most recent ANOVA, in a pointer of length one; the pointer is not formed if this was an unweighted analysis.

Alternatively, if INSAVE is set to an ANOVA save structure, the parameters Y, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE. DESIGN and WEIGHTS save the settings used to form INSAVE.

The SAVE parameter saves the save structure from the most recent ANOVA (regardless of the setting of INSAVE).

Option: PRINT.
Parameters: Y, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE, DESIGN, WEIGHTS, SAVE, INSAVE.

Method
ASTATUS uses the GET directive to obtain the current settings of BLOCKSTRUCTURE, TREATMENTSTUCTURE and COVARIATE, and the save structure from the most recent ANOVA. It uses AKEEP, and specialist knowledge of the save structure, to obtain information from an ANOVA save structure.

See also
Directives: AKEEP, ANOVA, BLOCKSTRUCTURE, COVARIATE, TREATMENTSTRUCTURE.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ASTORE

Stores an ANOVA save structure in an external file (R.W. Payne).

No options

Parameters

FILENAME = *texts*	Name of the file to store the save structure
EXIT = *scalars*	Scalar that contains the value one if the save structure could not be stored successfully, otherwise zero
SAVE = *asave* structures	Save structure to be stored; default stores the save structure from the most recent ANOVA

Description

ASTORE stores an ANOVA save structure in an external file. It can then be loaded back into Genstat in a later run, by the ARETRIEVE procedure, so that further output can be produced from the analysis. (See, for example, directives ADISPLAY and AKEEP, or procedures ACHECK, AGRAPH, APLOT, APERMTEST and ASPREADSHEET.)

The name (and path) of the file to store the save structure is specified, in a text, by the FILENAME parameter. The save structure is specified by the SAVE parameter. If this is unset, ASTORE stores the save structure from the most recent ANOVA. The EXIT parameter can return a scalar containing the value one if the save structure could not be stored successfully. Otherwise it contains zero.

Options: none.
Parameters: FILENAME, EXIT, SAVE.

Method

The save structure is stored in a Genstat backing-store file by the STORE directive.

See also

Directive: ANOVA.
Procedure: ARETRIEVE.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

ASWEEP

Performs sweeps for model terms in an analysis of variance (R.W. Payne).

Options

TERM = *formula*	Model term (or terms) involved in the sweep (this need not be specified if EMETHOD=given); default is to sweep for the grand mean
EFFICIENCY = *scalar*	Efficiency factor of the term(s)
EMETHOD = *string token*	Source of the effects (calculated, given); default calc
RMETHOD = *string token*	Method to be used to obtain the residual variate (subtract, replace); default subt

Parameters

Y = *variate*	Working variates to be swept
EFFECTS = *table*	Estimated effects
RESIDUALS = *variate*	New working variates, following the sweep
SS = *scalars*	Sum of squares due to the term(s)
RSS = *scalars*	Sum of squares of the working variate after the sweep

Description

The analysis-of-variance algorithm in the Genstat ANOVA directive involves a series of sweep operations performed on a working variate which initially contains the data values and finally contains the residuals. Sweeps may have two parts. The first involves the estimation of the effects of a particular term. For a term that is orthogonal to the terms that preceed it in the model, the effects are estimated simply by the tables of means for that term, calculated from the working variate; for non-orthogonal terms, the effects are the means divided by an efficiency factor. In the second part, the working variate is modified. Usually this involves subtracting the estimated effects. Alternatively there is a special sweep, known as a pivot, which is used to initiate the analysis within a stratum. In this, the value in each unit of the working variate is replaced by the corresponding effect of the term. Further details can be found in the *Guide to the Genstat Command Language*, Part 2, Section 4.7.5, or in the paper by Payne & Wilkinson (1977). Procedure ASWEEP is provided as a research tool for studying the algorithm and its properties.

The values initially in the working variate are specified by the Y parameter. The procedure can sweep for a single term or, if several terms have the same efficiency factor, these can all be swept together. The TERM option specifies the term (or terms) and the efficiency factor is defined by the EFFICIENCY option. The EFFECTS parameter allows the estimated effects of the term(s) to be stored if option EMETHOD=calculated, or to be supplied if EMETHOD=given. The values in the working variate after the sweep can be saved using the RESIDUALS parameter, and the RMETHOD option indicates whether these are to be formed by an ordinary sweep (RMETHOD=subtract) or by a pivot (RMETHOD=replace). The SS parameter saves the sum of squares due to the term(s), and the RSS parameter saves the sum of squares of the working variate after the sweep.

Options: TERM, EFFICIENCY, EMETHOD, RMETHOD.
Parameters: Y, EFFECTS, RESIDUAL, SS, RSS.

Method

The procedure uses the standard Genstat directives for analysis of variance, calculations and manipulation.

Action with `RESTRICT`
If the working variate (specified by the `Y` parameter) is restricted, the sweep will use only the units not excluded by the restriction.

Reference
Payne, R.W. & Wilkinson, G.N. (1977). A general algorithm for analysis of variance. *Applied Statistics*, **26**, 251-260.

See also
Directive: `ANOVA`.
Procedures: `AEFFICIENCY`, `AMTIER`, `FPROJECTIONMATRIX`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AUDISPLAY

Produces further output for an unbalanced design (after AUNBALANCED) (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, effects, means, residuals, %cv); default aovt, mean
PFACTORIAL = *scalar*	Limit on number of factors in printed tables of predicted means; default 3
FPROBABILITY = *string token*	Printing of probabilities for variance ratios in the analysis-of-variance table (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-tests of effects (yes, no); default no
PLOT = *string tokens*	Which residual plots to provide (fittedvalues, normal, halfnormal, histogram); default * i.e. none
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
PSE = *string tokens*	Types of standard errors to be printed with the predicted means (differences, alldifferences, lsd, alllsd, means, ese); default diff
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
RMETHOD = *string token*	Type of residuals to plot (simple, standardized); default simp
PMEANTERMS = *formula*	Treatment terms for which predicted means are to be printed; default * implies all the treatment terms

Parameter

SAVE = *identifiers*	Save structure (from AUNBALANCED) containing details of the analysis for which further output is required; if omitted, output is from the most recent use of AUNBALANCED

Description

This procedure can be used, following the use of procedure AUNBALANCED, to produce further output for the analysis of variance of an unbalanced design.

The output to be printed is controlled by the PRINT option, with settings: aovtable to print the analysis-of-variance table, effects to print the effects (as estimated by Genstat regression), means to print tables of predicted means with standard errors, residuals to print residuals and fitted values, and %cv to print the coefficient of variation. The default is to print the analysis-of-variance table and tables of means.

The model is fitted sequentially, first any covariates and then the treatments. Thus, the sum of square in each line of the analysis-of-variance table is for the term concerned, eliminating the effects of terms in earlier lines and ignoring the effects of terms lower in the table. In particular, the sums of squares for covariates are ignoring treatments, and not after eliminating treatments (as with the ANOVA directive).

Tables of means are calculated using the PREDICT directive. The first step (A) of the calculation forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The

COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data.

The PSE option controls the types of standard errors that are produced to accompany the tables of means, with settings:

differences	summary of standard errors for differences between pairs of means;
alldifferences	standard errors for differences between all pairs of means;
lsd	summary of least significant differences between pairs of means;
alllsd	least significant differences between all pairs of means;
means	standard errors of the means (relevant for comparing them with zero);
ese	approximate effective standard errors – these are formed by procedure SED2ESE with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula of $sed_{i,j} = \sqrt{(ese_i^2 + ese_j^2)}$.

The default is differences. The LSDLEVEL option sets the significance level (as a percentage) for the least significant differences.

The PFACTORIAL option limits the number of factors in terms for which predicted means are printed. Probabilities can be printed for variance ratios by setting option FPROBABILITY=yes, and probabilities for t-tests of effects by setting option TPROBABILITY=yes. Finally, there is a PLOT option which allows various residual plots to be requested: fittedvalues for a plot of residuals against fitted values, normal for a Normal plot, halfnormal for a half Normal plot, and histogram for a histogram of residuals. By default, simple residuals are plotted, but you can set option RMETHOD=standardized to plot standardized residuals instead. The PMEANTERMS option can be used to specify the treatment terms for which predicted means are to be printed; by default, they are printed for all the treatment terms (subject, of course, to the PFACTORIAL option).

The SAVE parameter can be set to the save structure from the analysis for which further output is required. If SAVE is not set, output will be produced for the most recent analysis from AUNBALANCED; however, none of the Genstat regression directives (MODEL, TERMS, FIT, ADD, DROP and so on) must then have been used in the interim.

Options: PRINT, PFACTORIAL, FPROBABILITY, TPROBABILITY, PLOT, COMBINATIONS, ADJUSTMENT, PSE, LSDLEVEL, RMETHOD.
Parameter: SAVE.

Method
The output is produced mainly using the directives RKEEP and PREDICT.

Action with RESTRICT
If the y-variate originally analysed by AUNBALANCED was restricted, only the units not excluded by the restriction will have been analysed.

See also
Procedures: AUNBALANCED, AUGRAPH, AUPREDICT, AUMCOMPARISON, AUKEEP.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AUGRAPH

Plots tables of means from AUNBALANCED (R.W. Payne).

Options

GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
METHOD = *string token*	What to plot (means, lines, data, barchart, splines); default mean
XFREPRESENTATION = *string token*	How to label the *x*-axis (levels, labels); default labels uses the XFACTOR labels, if available
PSE = *string token*	What to plot to represent variation (differences, lsd, means, allmeans); default diff
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
LSDLEVEL = *scalar*	Significance level (%) to use for least significant differences; default 5
DFSPLINE = *scalar*	Number of degrees of freedom to use when METHOD=splines
YTRANSFORM = *string tokens*	Transformed scale for additional axis marks and labels to be plotted on the right-hand side of the y-axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden i.e. none
PENYTRANSFORM = *scalar*	Pen to use to plot the transformed axis marks and labels; default * selects a pen, and defines its properties, automatically
†KEYMETHOD = *string token*	What to use for the key descriptions when GROUPS specifies more than one factor (labels, namesandlabels); default name
†PLOTTITLEMETHOD = *string token*	What to use for the titles of the plots when TRELLISGROUPS specifies more than one factor (labels, namesandlabels); default name
†PAGETITLEMETHOD = *string token*	What to use for the titles of the pages when PAGEGROUPS specifies more than one factor (labels, namesandlabels); default name
†USEAXES = *string token*	Which aspects of the current axis definitions of window 1 to use (none, limits, marks, mpositions, nsubticks,); default none
SAVE = *regression save structure*	Save structure to provide the table of means; default uses the save structure from the most recent AUNBALANCED analysis (provided no other regression analysis has been done in the interim)

Parameters

XFACTOR = *factors*	Factor providing the *x*-values for each plot
GROUPS = *factors* or *pointers*	Factor or factors identifying groups of points in each plot; by default chosen automatically
TRELLISGROUPS = *factors* or *pointers*	

	Factor or factors specifying the different plots of a trellis plot of a multi-way table
PAGEGROUPS = *factors* or *pointers*	Factor or factors specifying plots to be displayed on different pages
NEWXLEVELS = *variates*	Values to be used for XFACTOR instead of its existing levels
TITLE = *texts*	Title for the graph; default defines a title automatically
YTITLE = *texts*	Title for the y-axis; default is to use the identifier of the y-variate, or to have no title if this is unnamed
XTITLE = *texts*	Title for the x-axis; default is to use the identifier of the XFACTOR
PENS = *variates*	Defines the pen to use to plot the points and/or line for each group defined by the GROUPS factors

Description

AUGRAPH plots tables of predicted means from an analysis by AUNBALANCED. The SAVE option can be set to the save structure from the analysis from which the means should be taken. If SAVE is not set, the means will be from the most recent analysis by AUNBALANCED; however, none of the Genstat regression directives (MODEL, TERMS, FIT, ADD, DROP and so on) must then have been used in the interim.

In its simplest form, the behaviour of AUGRAPH depends on the model. If the treatment model contains only main effects, it plots the means for the first factor in the model. Otherwise it looks for the first treatment term involving two factors; it then plots the means with one of these factors as the x-axis, and the second as a grouping factor with levels identified by different plotting colours and symbols. The means are predicted by the AUKEEP procedure using the averaging and adjustment methods specified by the COMBINATIONS and ADJUSTMENT options; see AUKEEP for details.

Usually, each mean is represented by a point. However, with high-resolution plots, the METHOD option can be set to lines to draw lines between the points, or data to draw just the lines and then also plot the original data values, or barchart to plot the means as a barchart, or splines to plot the points together with a smooth spline to show the trend over each group of points. The DFSPLINE specifies the degrees of freedom for the splines; if this is not set, 2 d.f. are used when there are up to 10 points, 3 if there are 11 to 20, and 4 for 21 or more. The GRAPHICS option controls whether a high-resolution or a line-printer graph is plotted; by default GRAPHICS=high.

The PSE option specifies the type of error bar to be plotted with the means, with settings:

differences	average standard error of difference;
lsd	average least significant difference;
means	average effective standard error for the means;
allmeans	plots plus and minus the effective standard error around every mean.

The LSDLEVEL option sets the significance level (%) to use for the least significant differences (default 5). The allmeans setting is often unsuitable for plots other than barcharts when there are GROUPS, as the plus/minus e.s.e. bars may overlap each other.

You can define the table of means to plot explicitly, by specifying its classifying factors using the XFACTOR, GROUPS, TRELLISGROUPS and PAGEGROUPS parameters. The XFACTOR parameter defines the factor against whose levels the means are plotted. With a multi-way table, there will be a plot of means against the XFACTOR levels for every combination of levels of the factors specified by the GROUPS, TRELLISGROUPS and PAGEGROUPS parameters. The GROUPS parameter specifies factors whose levels are to be included in a single window of the graph. So, for example, if you specify

```
AUGRAPH [METHOD=line] XFACTOR=A; GROUPS=B
```

AUGRAPH will produce plot the means in a single window with factor A on the x-axis, and a line for each level of the factor B. You can set GROUPS to a pointer to specify several factors to define groups. For example

```
POINTER [VALUES=B,C] Groupfactors
AUGRAPH [METHOD=line] XFACTOR=A; GROUPS=Groupfactors
```

to plot a line for every combination of the levels of factors B and C. Similarly, the TRELLISGROUPS option can specify one or more factors to define a trellis plot. For example,

```
AUGRAPH [METHOD=line] XFACTOR=A; GROUPS=B; TRELLISGROUPS=C
```

will produce a plot for each level of C, in a trellis arrangement; each plot will again have factor A on the x-axis, and a line for each level of the factor B. Likewise, the PAGEGROUPS parameter can specify factors whose combinations of levels are to be plotted on different pages. So

```
AUGRAPH [METHOD=line] XFACTOR=A; GROUPS=B; PAGEGROUPS=C
```

will produce a plot for each level of C, but now on separate pages. Multi-way tables can plotted even if the corresponding model term was not in the ANOVA analysis. For example you can plot a two-way table even if the analysis contained only the main effects of the two factors; however, the lines will then all be parallel and no standard errors or LSDs can be included.

The NEWXLEVELS parameter enables different levels to be supplied for XFACTOR if the existing levels are unsuitable. If XFACTOR has labels, these are used to label the x-axis unless you set option XFREPRESENTATION=levels.

The TITLE, YTITLE and XTITLE parameters can supply titles for the graph, the y-axis and the x-axis, respectively. The symbols, colours and line styles that are used in a high-resolution plot are usually set up by AUGRAPH automatically. If you want to control these yourself, you should use the PEN directive to define a pen with your preferred symbol, colour and line style, for each of the groups defined by combinations of the GROUPS factors. The pen numbers should then be supplied to AUGRAPH, in a variate with a value for each group, using the PENS parameter.

The YTRANSFORM option allows you to include additional axis markings, transformed onto another scale, on the right-hand side of the y-axis. Suppose, for example, suppose you have analysed a variate of percentages that have been transformed to logits. You might then set YTRANSFORM=ilogit (the inverse-logit transformation) to include markings in percentages alongside the logits. The settings are the same as those of the TRANSFORM parameter of AXIS (which is used to add the markings). You can control the colours of the transformed marks and labels, by defining a pen with the required properties, and specifying it with the PENYTRANSFORM option. Otherwise, the default is to plot them in blue.

When there is more than one GROUPS factor, the KEYMETHOD controls whether to use the factor names with their labels (or levels for factors with no labels) or just the labels (or levels) in the key descriptions. The default is to use the names and the labels (or levels). Similarly, the PLOTTITLEMETHOD specifies what to use for the titles of the plots when there is more then one TRELLISGROUPS factor, and the PAGETITLEMETHOD specifies what to use for the titles of the plots when there is more then one PAGEGROUPS factor. You can setKEYMETHOD=* to have no key at all.

The USEAXES option allows you to control various aspects of the axes. First you need to use the XAXIS and YAXIS directives to define them for window 1. Then specify which of the aspects of the axes in window 1 are to be used by DTABLE, by specifying USEAXES with the following settings:

limits	y- and x-axis limits (LOWER and UPPER parameters);
marks	location and labelling of the tick marks (MARKS, LABELS, LDIRECTION, LROTATION, DECIMALS, DREPRESENTATION, and VREPRESENTATION parameters);
mpositions	positions of the tick marks (MPOSITION parameter); and

`nsubticks` number of subticks per interval (`NSUBTICKS` parameter). By default none are used.

Options: `GRAPHICS`, `METHOD`, `XFREPRESENTATION`, `PSE`, `COMBINATIONS`, `ADJUSTMENT`, `LSDLEVEL`, `DFSPLINE`, `YTRANSFORM`, `PENYTRANSFORM`, `KEYMETHOD`, `PLOTTITLEMETHOD`, `PAGETITLEMETHOD`, `USEAXES`, `SAVE`.

Parameters: `XFACTOR`, `GROUPS`, `TRELLISGROUPS`, `PAGEGROUPS`, `NEWXLEVELS`, `TITLE`, `YTITLE`, `XTITLE`, `PENS`.

See also

Procedures: `AUNBALANCED`, `AUDISPLAY`, `AUPREDICT`, `AUMCOMPARISON`, `AUKEEP`, `RGRAPH`, `VGRAPH`.

Genstat Reference Manual 1 Summary section on: Analysis of variance.

AUKEEP

Saves output from analysis of an unbalanced design (by AUNBALANCED) (R.W. Payne).

Options

FACTORIAL = *scalar*	Limit on number of factors in the model terms generated from the TERMS parameter; default 3
RESIDUALS = *variate*	To save residuals from the analysis
FITTEDVALUES = *variate*	To save fitted values
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
LSDLEVEL = *scalar*	Significance level (as a percentage) for the least significant differences
RMETHOD = *string token*	Type of residuals to form if the RESIDUALS option is set (simple, standardized); default simp
SAVE = *identifier*	Save structure (from AUNBALANCED) containing details of the analysis for which further output is required; if omitted, output is from the most recent use of AUNBALANCED

Parameters

TERMS = *formula*	Model terms for which information is required
MEANS = *table* or *pointer* to *tables*	Predicted means for each term
SEMEANS = *table* or *pointer* to *tables*	Standard errors of the means for each term
SEDMEANS = *symmetric matrix* or *pointer* to *symmetric matrices*	Standard errors of differences between means
ESEMEANS = *table* or *pointer* to *tables*	Approximate effective standard errors of the means: these are formed by procedure SED2ESE with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula $sed_{i,j} = \sqrt{(ese_i^2 + ese_j^2)}$
LSD = *symmetric matrix* or *pointer* to *symmetric matrices*	Least significant differences

Description

This procedure can be used, following the use of procedure AUNBALANCED, to save output for the analysis of variance of an unbalanced design.

Options are provided to save information about the analysis as a whole. The RESIDUALS and FITTEDVALUES options allow variates to be specified to store the residuals and fitted values, respectively. The RMETHOD option controls whether simple or standardized residuals are saved; by default RMETHOD=simple.

The SAVE option can be set to the save structure from the analysis from which output is to be saved. If SAVE is not set, output will be produced for the most recent analysis from AUNBALANCED; however, none of the Genstat regression directives (MODEL, TERMS, FIT, ADD, DROP and so on) must then have been used in the interim.

The parameters of AUKEEP save information about particular model terms in the analysis. With the TERMS parameter you specify a model formula, which Genstat expands to form the series of model terms about which you wish to save information. As in AUNBALANCED, the FACTORIAL

option sets a limit on the number of factors in each term. Any term containing more than that limit is deleted. The subsequent parameters allow you to specify identifiers of data structures to store various components of information for each of the terms that you have specified. The MEANS parameter saves tables of predicted means, the SEMEANS parameter saves tables of standard errors for the means, the SEDMEANS parameter saves symmetric matrices of standard errors of differences, the ESEMEANS parameter saves tables of approximate effective standard errors, and the LSD parameter saves symmetric matrices of least significant differences. If you have a single term, you can supply a table or symmetric matrix for each of these parameters, as appropriate. However, if you have several terms, you must supply a pointer which will then be set up to contain as many tables or symmetric matrices as there are terms. The LSDLEVEL option sets the significance level (as a percentage) for the least significant differences.

Tables of means are calculated using the PREDICT directive. The first step (A) of the calculation forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data.

Options: FACTORIAL, RESIDUALS, FITTEDVALUES, COMBINATIONS, ADJUSTMENT, LSDLEVEL, RMETHOD, SAVE.
Parameters: TERMS, MEANS, SEMEANS, SEDMEANS, ESEMEANS, LSD.

Method
The output is obtained mainly using the directives RKEEP and PREDICT.

Action with RESTRICT
If the y-variate originally analysed by AUNBALANCED was restricted, only the units not excluded by the restriction will have been analysed.

See also
Procedures: AUNBALANCED, AUDISPLAY, AUGRAPH, AUPREDICT, AUMCOMPARISON.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AUMCOMPARISON

Performs pairwise multiple comparison tests for means from an unbalanced analysis of variance, performed previously by AUNBALANCED (D.M. Smith).

Options

PRINT = *string tokens*	Controls printed output (comparisons, critical, description, lines, letters, plot, mplot, pplot); default lett
METHOD = *string token*	Test to be performed (flsd, bonferroni, sidak); default flsd
FACTORIAL = *scalar*	Limit on the number of factors in each term; default 3
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
WEIGHTS = *table*	Weights classified by some or all of the factors in the model
DIRECTION = *string token*	How to sort means (ascending, descending); default asce
PROBABILITY = *scalar*	The required significance level; default 0.05
STUDENTIZE = *string token*	Whether to use the alternative LSD test where the Studentized Range statistic is used instead of Student's t (yes, no); default no
SAVE = *identifier*	Save structure to provide the table of means; default uses the save structure from the most recent AUNBALANCED analysis

Parameters

TERMS = *formula*	Treatment terms whose means are to be compared
MEANS = *pointer* or *variate*	Saves the (sorted) means
DIFFERENCES = *pointer* or *symmetric matrix*	Saves differences between the (sorted) means
LABELS = *pointer* or *text*	Saves labels for the (sorted) means
LETTERS = *pointer* or *text*	Saves letters indicating groups of means that do not differ significantly
SIGNIFICANCE = *pointer* or *symmetric matrix*	Indicators to show significant comparisons between (sorted) means
CIWIDTH = *pointer* or *symmetric matrix*	Saves the width of the confidence interval for the absolute differences between the (sorted) means

Description

AUMCOMPARISON can be used following an analysis by AUNBALANCED to perform all pairwise multiple comparison tests on tables of predicted means. The methodology implemented in the procedure closely follows that described in Chapter 5 of Hsu (1996).

The TERMS parameter specifies a model formula to define the treatment terms whose means are to be compared. The means are usually taken from the most recent analysis performed by AUNBALANCED, but you can set the SAVE option to a save structure from another AUNBALANCED if you want to examine means from an earlier analysis. The FACTORIAL option sets a limit on the number of factors in each term (default 3).

The predicted means are formed using the AUPREDICT procedure. The COMBINATIONS, ADJUSTMENT and WEIGHTS options control how this is done; see AUPREDICT for more details.

Printed output is controlled by the PRINT option, with settings:

comparisons	prints the differences between the pair of means, upper and lower confidence limits for the differences, t-statistics and an indication of whether or not they are significant;
critical	gives critical values for the t-statistic for situations where these do not vary amongst the comparisons (i.e. for the Scheffe, Bonferroni and Sidak methods, as well as the Fisher LSD methods provided all the comparisons have the same number of residual degrees of freedom);
description	provides a description including information such as the experiment-wise and compartment-wise error rates;
lines	gives the means, with lines joining those that do not differ significantly;
letters	gives the means, with identical letters (a, b etc.) alongside those that do not differ significantly;
mplot	does a mean-mean scatter plot (synonym plot);
pplot	displays the probabilities in a shade plot.

By default, PRINT=letters.

The means are usually sorted into ascending order, but you can set option DIRECTION=descending for descending order, or DIRECTION=* to leave them in their original order. Note, though, that the lines joining means with non-significant differences may then be broken.

In most unbalanced anova's the standard errors for the differences between the means will be unequal, and the memberships of the groups defined by the lines or letters may then be inconsistent. Suppose, for example, you have ordered means A, B and C. If the s.e.d. for A vs. C is large compared to those for A vs. B and B vs C, you might find that there is no significant difference between A and C, but there are significant differences between A and B, and between B and C. So treatments A and B and treatments B and C would be in different groups. However, treatments A and C (which are further apart) would be in the same group. This contradicts the idea behind multiple comparisons, where you expect that if two means are in the same group, than any mean between them should be in that group too. If AUMCOMPARISON finds inconsistencies like this, it gives a diagnostic and suppresses the printing of lines and letters (but not the other types of output).

The mean-mean scatter plot allows you to assess the confidence region for the difference between each pair of means visually. It has grid lines from both the x- and y-axis at the position of each mean, and a diagonal line at 45 degrees marking y=x. The confidence interval for each pair of means is plotted as a line at an angle of −45 degrees and centred on the intersection above the line y=x of the grid lines for the two means (so the y grid line is for the larger of the two means, and the x grid line is for the smaller mean). The difference between the means is significant if their confidence line does not intersect the line y=x. For more details, see Hsu (1996) pages 151-153.

The shade plot displays the probabilities in a symmetric matrix. The colour of each cell represents the probability for the difference between the means for the treatments in the corresponding row and column.

The type of test to be performed is specified by the METHOD option, with settings FLSD (Fisher's Unprotected Least Significant Difference), Bonferroni and Sidak. The PROBABILITY option allows the experiment-wise significance level for the intervals from the Bonferroni and Sidak tests to be changed from the default 0.05 (e.g. to 0.01). For the Fisher's test, it changes the pair-wise significance level. The STUDENTIZE option can specify that the

Fisher's protected or unprotected LSD tests should use the Studentized Range statistic rather than Student's t (for further information see Hsu 1996, page 139).

The MEANS parameter can save the means, sorted according to the DIRECTION option and omitting any that were non-estimable. If the TERMS parameter specifies a single term, MEANS should be set to a variate. If TERMS specifies several terms, you must supply a pointer which will then be set up to contain as many variates as there are terms. Similarly the LABELS parameter can save labels to identify the means, in either a text (for a single term) or in a pointer of texts (for several). Likewise the LETTERS parameter can save texts with the letters identifying means that do not differ significantly, and the SIGNIFICANCE parameter can save symmetric matrices containing ones or zeros according to whether the various comparisons were significant or non-significant. The DIFFERENCES parameter can save symmetric matrices containing the differences between the (sorted) means, and the CIWIDTH parameter can save symmetric matrices containing the widths of the confidence intervals for the differences.

Options: PRINT, METHOD, FACTORIAL, COMBINATIONS, ADJUSTMENT, WEIGHTS, DIRECTION, PROBABILITY, STUDENTIZE, SAVE.
Parameter: TERMS, MEANS, DIFFERENCES, LABELS, LETTERS, SIGNIFICANCE, CIWIDTH.

Method
The methodology implemented is based on that described in Hsu (1996).

Reference
Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also
Procedures: AUNBALANCED, AUDISPLAY, AUGRAPH, AUPREDICT, AUKEEP, AMCOMPARISON, AMDUNNETT, MCOMPARISON, VMCOMPARISON.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AUNBALANCED

Performs analysis of variance for unbalanced designs (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, effects, means, residuals, screen, %cv); default aovt, mean
FACTORIAL = *scalar*	Limit on number of factors in a treatment term; default 3
PFACTORIAL = *scalar*	Limit on number of factors in printed tables of predicted means; default 3
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default * i.e. none
FPROBABILITY = *string token*	Printing of probabilities for variance ratios in the analysis-of-variance table (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-tests of effects (yes, no); default no
PLOT = *string tokens*	Which residual plots to provide (fittedvalues, normal, halfnormal, histogram); default * i.e. none
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
WEIGHTS = *variate*	Weights for each unit; default * i.e. all units with weight one
PSE = *string tokens*	Types of standard errors to be printed with the predicted means (differences, alldifferences, lsd, alllsd, means, ese); default diff
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
RMETHOD = *string token*	Type of residuals to plot (simple, standardized); default simp

Parameters

Y = *variates*	Data values to be analysed
RESIDUALS = *variates*	Variate to save the residuals from each analysis
FITTEDVALUES = *variates*	Variate to save the fitted values from each analysis
SAVE = *identifiers*	To save details of each analysis to use subsequently with the AUDISPLAY procedure

Description

This procedure carries out analysis of variance using the regression directives in Genstat. It is particularly useful for designs that are unbalanced and which thus cannot be analysed by the ANOVA directive.

The method of use is similar to that for ANOVA. The treatment terms to be fitted must be specified, before calling the procedure, by the TREATMENTSTRUCTURE directive. Similarly, any covariates must be indicated by the COVARIATE directive. The procedure also takes account of any blocking structure specified by the BLOCKSTRUCTURE directive. However, it cannot produce stratified analyses like those generated by ANOVA, and is able to estimate treatments and covariates only in the "bottom stratum". So, for example, the full analysis can be produced for a randomized block design, where the treatments are all estimated on the plots within blocks, but

it cannot produce the whole-plot analysis in a split plot design.

The parameters of the procedure are identical to those of ANOVA. The variates to be analysed are specified by the Y parameter. Residuals and fitted values can be saved using the RESIDUALS and FITTEDVALUES parameters respectively. Finally, the SAVE parameter allows details of the analysis to be saved so that further output can be obtained using the AUDISPLAY procedure, or information can be copied into Genstat data structures using the AUKEEP procedure. (Note that this is a regression save structure, not an ANOVA structure, so it cannot be used with the directives ADISPLAY or AKEEP.)

Printed output is controlled by the PRINT option, with settings: aovtable to print the analysis-of-variance table, effects to print the effects (as estimated by Genstat regression), means to print tables of predicted means with standard errors, residuals to print residuals and fitted values, screen to print "screening" tests for treatment terms, and %cv to print the coefficient of variation. The default is to print the analysis-of-variance table and tables of means.

The model is fitted sequentially, first any block terms, then any covariates and then the treatments. Thus, the sum of square in each line of the analysis-of-variance table is for the term concerned, eliminating the effects of terms in earlier lines and ignoring the effects of terms lower in the table. In particular, the sums of squares for covariates are ignoring treatments, and not after eliminating treatments (as with the ANOVA directive). Alternatively, the screen setting calls the RSCREEN procedure to provide screening tests for the treatment terms: marginal tests to assess the effect of adding each term to the simplest possible model (i.e. a model containing any blocks and covariates, and any terms marginal to the term); conditional tests to assess the effect of adding each term to the fullest possible model (i.e. a model containing all terms other than those to which the term is marginal). For example, if we have

```
BLOCKSTRUCTURE Blocks
```

and

```
TREATMENTSTRUCTURE A + B + A.B
```

the marginal test for A will show the effect of adding A to a model containing only Blocks, while the conditional test will show the effect of adding A to a model containing Blocks and B. (The terms A and B are marginal to A.B.)

Tables of means are calculated using the PREDICT directive. The first step (A) of the calculation forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data.

The PSE option controls the types of standard errors that are produced to accompany the tables of means, with settings:

differences	summary of standard errors for differences between pairs of means;
alldifferences	standard errors for differences between all pairs of means;
lsd	summary of least significant differences between pairs of means;
alllsd	least significant differences between all pairs of means;

means	standard errors of the means (relevant for comparing them with zero);
ese	approximate effective standard errors – these are formed by procedure SED2ESE with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula of $sed_{i,j} = \sqrt{(ese_i^2 + ese_j^2)}$.

The default is differences. The LSDLEVEL option sets the significance level (as a percentage) for the least significant differences.

The FACTORIAL option sets a limit on the number of factors that a higher-order term, such as an interaction, can contain; any terms with more factors are deleted from the analysis. Similarly, the PFACTORIAL option limits the number of factors in terms for which predicted means are printed. Probabilities can be printed for variance ratios by setting option FPROBABILITY=yes, and probabilities for t-tests of effects by setting option TPROBABILITY=yes. The WEIGHTS option allows a variate of weights to be specified for a weighted analysis of variance. The NOMESSAGE option allows various warning messages (produced by the FIT directive) to be suppressed, and the PLOT option allows various residual plots to be requested: fittedvalues for a plot of residuals against fitted values, normal for a Normal plot, halfnormal for a half Normal plot, and histogram for a histogram of residuals. By default, simple residuals are plotted, but you can set option RMETHOD=standardized to plot standardized residuals instead.

Options: PRINT, FACTORIAL, PFACTORIAL, NOMESSAGE, FPROBABILITY, TPROBABILITY, PLOT, COMBINATIONS, ADJUSTMENT, PSE, WEIGHTS, LSDLEVEL, RMETHOD.
Parameters: Y, RESIDUALS, FITTEDVALUES, SAVE.

Method
The y-variate is specified using the MODEL directive, along with any variates to save residuals and fitted values. The current settings of the TREATMENTSTRUCTURE and COVARIATE directives are recovered using the SET directive, and used to define the terms in the analysis (using the TERMS directive). The model is then fitted (using FIT), AUDISPLAY is called to print the output and any plots of residuals.

Action with RESTRICT
If the Y variate is restricted, only the units not excluded by the restriction will be analysed.

See also
Directives: ANOVA, REML.
Procedures: AUDISPLAY, AUGRAPH, AUPREDICT, AUMCOMPARISON, AUKEEP.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AUPREDICT

Forms predictions from an unbalanced analysis of variance, performed by AUNBALANCED (R.W. Payne).

Options

PRINT = *string tokens*	What to print (description, predictions, se, sed, sedsummary, ese, lsd, lsdsummary, vcovariance); default pred, sed
MODEL = *formula*	Model to use to calculate the predictions; default * i.e. full model fitted by AUNBALANCED
FACTORIAL = *scalar*	Limit on number of factors or variates in each term specified by MODEL; default 3
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
WEIGHTS = *table*	Weights classified by some or all of the factors in the model
PREDICTIONS = *tables* or *scalars*	Saves predictions; default *
SE = *tables* or *scalars*	Saves standard errors of predictions; default *
SED = *symmetric matrices*	Saves matrices of standard errors of differences between predictions; default *
ESE = *table*	Saves effective standard errors; default *
LSD = *symmetric matrix*	Saves least significant differences between predictions; default *
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
VCOVARIANCE = *symmetric matrices*	Saves variance-covariance matrices of predictions; default *
SAVE = *identifier*	Save structure (from AUNBALANCED) containing details of the analysis for which predictions are required; if omitted, output is from the most recent use of AUNBALANCED

Parameters

CLASSIFY = *vectors*	Variates and/or factors to classify table of predictions
LEVELS = *variates* or *scalars*	To specify values of variates, levels of factors

Description

AUPREDICT can produce predicted means following an analysis of variance of an unbalanced design by AUNBALANCED. The predictions are calculated using the PREDICT directive. The first step (A) of the calculation forms a full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data. The ADJUSTMENT and WEIGHTS options then define how the averaging is done in Step B. The WEIGHTS option allows you to specify your own table of weights to use in the averaging. Alternatively, if WEIGHTS is not set, the weights are formed automatically according

to the setting of the ADJUSTMENT option. The default setting, marginal, of ADJUSTMENT forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data.

Printed output, which extends the output available from PREDICT, is controlled by settings of the PRINT option:

description	standardization policies used when forming the predictions,
predictions	predictions,
se	predictions and standard errors,
sed	standard errors for differences between the predictions,
sedsummary	summary of the standard errors for differences between the predictions,
lsd	least significant differences between the predictions,
lsdsummary	summary of the least significant differences between the predictions,
ese	approximate effective standard errors – these are formed by procedure SED2ESE with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula of $sed_{i,j} = \sqrt{(ese_i^2 + ese_j^2)}$, and
vcovariance	variance and covariances of the predictions.

The default is to print predictions and a summary of the standard errors of differences. The standard errors (and sed's) are relevant for the predictions when considered as means of those data that have been analysed, with the means formed according to the averaging policy defined by the options of PREDICT. The word *prediction* is used because these are predictions of what the means would have been if the factor levels been replicated differently in the data; see Lane & Nelder (1982) for more details. The LSDLEVEL option specifies the significance level (%) to use in the calculation of least significant differences (default 5%).

Another extension in AUPREDICT is that you can produce predictions using a smaller model than the full model that has been fitted by AUNBALANCED. This can be useful if the full model contains many parameters. A substantial amount of time and computer workspace may then be needed to calculate the predictions and standard errors. Very large models may even exceed the capacity of some PCs.

You might choose to omit a term from the full model when forming a particular table of predictions if the term is orthogonal to all the terms involved in the table. For example, you might omit the term blocks when forming an A-by-B table of predictions if each combination of levels of the factors A and B is replicated the same number of times in every block. The justification is that an orthogonal term cannot affect the size of any of the differences between predictions. Different weighting of the levels of the orthogonal term may affect the overall mean of the predictions, but this is usually unimportant. If you omit the term, it is though you had included it with weightings based on the observed replication of its levels in the data set – and in any well-designed data set these should provide a satisfactory outcome. You might also omit a term if it is nearly orthogonal to the terms involved in the table, and you are happy to ignore its effect on the predictions.

The model is specified by the MODEL option. The FACTORIAL option sets a limit on number of factors or variates in each term specified by MODEL; default 3.

The PREDICTIONS, SE, SED, ESE, LSD and VCOVARIANCE options allow the results of the prediction to be save in appropriate Genstat data structures.

The SAVE option allows you to specify save structure from the analysis for which further output is required. If SAVE is not set, output will be produced for the most recent analysis from AUNBALANCED; however, none of the Genstat regression directives (MODEL, TERMS, FIT, ADD, DROP and so on) must then have been used in the interim.

Options: PRINT, MODEL, FACTORIAL, COMBINATIONS, ADJUSTMENT, WEIGHTS, PREDICTIONS, SE, SED, ESE, LSD, LSDLEVEL, VCOVARIANCE, SAVE.
Parameters: CLASSIFY, LEVELS.

Method
The predictions are produced using the PREDICT directive.

Reference
Lane, P.W. & Nelder, J.A. (1982). Analysis of covariance and standardization as instances of prediction. *Biometrics*, **38**, 613-621.

See also
Directive: PREDICT.
Procedures: AUNBALANCED, AUDISPLAY, AUGRAPH, AUMCOMPARISON, AUKEEP.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

AUSPREADSHEET

Saves results from an analysis of an unbalanced design (by AUNBALANCED) in a spreadsheet (R.W. Payne).

Options

MEANS = *pointer*	Pointer to tables to contain the treatment means; default means
SEMEANS = *pointer*	Pointer to tables to contain the standard errors of treatment means; default sem
SEDMEANS = *pointer*	Pointer to matrices to contain standard errors of differences of treatment means; default sed
ESEMEANS = *pointer*	Pointer to matrices to contain effective standard errors of treatment means; default ese
EFFECTS = *pointer*	Pointer to contain the estimated effects, their standard errors, t-statistics and probabilities; default effects
REPLICATIONS = *pointer*	Pointer to tables of treatment replications; default replication
RESIDUALS = *variate*	Variate to save the residuals in the fittedvalues page; default residuals
FITTEDVALUES = *variate*	Variate to save the fitted values in the fittedvalues page; default fittedvalues
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
AOVTABLE = *pointer*	Pointer to a text and variates containing the information in the analysis-of-variance table; default aovtable
RMETHOD = *string token*	Type of residuals to form (simple, standardized); default simp
†LSDMEANS = *pointer*	Pointer to matrices to contain least significant differences for means
†LSDLEVEL = *scalar*	Significance level (as a percentage) for the least significant differences; default 5
†SPREADSHEET = *string tokens*	What to include in the spreadsheet (aovtable, effects, means, semeans, sedmeans, esemeans, lsdmeans, replications, fittedvalues); default aovt, mean, sedm, repl, fitt
OUTFILENAME = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create
SAVE = *identifier*	Save structure (from AUNBALANCED) containing details of the analysis for which further output is required; if omitted, output is from the most recent use of AUNBALANCED

No parameters

Description

AUSPREADSHEET puts results from an analysis, by AUNBALANCED, of an unbalanced design into a spreadsheet. By default the results are from the most recent analysis by AUNBALANCED, but you use the SAVE option to specify the save structure from some other analysis.

The SPREADSHEET option specifies which pages of the spreadsheet to form, with settings:

aovtable	analysis of variance table,

`effects`	estimates of effects, with their standard errors, t-statistics and probabilities,
`means`	tables of treatment means,
`semeans`	tables of standard errors of treatment means,
`sedmeans`	matrices of standard errors of differences of treatment means,
`esemeans`	tables of effective standard errors of treatment means,
`lsdmeans`	matrices of least significant differences of treatment means,
`replications`	replication tables of treatment terms,
`fittedvalues`	y-variate, fitted values and residuals.

By default, `SPREADSHEET = aovt, mean, sedm, repl, fitt`.

To help avoid clashes between the columns of the spreadsheets if you want to save results from more than one analysis, the parameters `MEANS`, `SEMEANS`, `SEDMEANS`, `ESEMEANS`, `LSDMEANS`, `EFFECTS`, `REPLICATIONS`, `RESIDUALS`, `FITTEDVALUES` and `AOVTABLE` allow you to specify identifiers for the columns (or sets of columns) that will store the corresponding results in the current spreadsheet.

Tables of means are obtained from the `AUKEEP` procedure which uses the `PREDICT` directive. The first step (A) of the calculation forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The `COMBINATIONS` option specifies which cells of the full table are to be formed in Step A. The default setting, `estimable`, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting `COMBINATIONS=present` excludes the cells for factor combinations that do not occur in the data. The `ADJUSTMENT` option then defines how the averaging is done in Step B. The default setting, `marginal`, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting `equal` weights all the combinations equally. Finally, the setting `observed` uses the `WEIGHTS` option of `PREDICT` to weight each factor combination according to its own individual replication in the data.

The `LSDLEVEL` option specifies the significance level (as a percentage) for the least significant differences; default 5.

You can save the data in either a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx), by setting the `OUTFILENAME` option to the name of the file to create. If the name is specified without a suffix, `'.gwb'` is added (so that a Genstat workbook is saved). If `OUTFILENAME` is not specified, the data are put into a spreadsheet opened inside Genstat.

Options: `MEANS`, `SEMEANS`, `SEDMEANS`, `ESEMEANS`, `EFFECTS`, `REPLICATIONS`, `RESIDUALS`, `FITTEDVALUES`, `COMBINATIONS`, `ADJUSTMENT`, `AOVTABLE`, `RMETHOD`, `LSDMEANS`, `LSDLEVEL`, `SPREADSHEET`, `OUTFILENAME`, `SAVE`.

Parameters: none.

Action with `RESTRICT`

If the `Y` variate is restricted, that restriction will carry over into the fitted-values spreadsheet.

See also

Directive: `SPLOAD`.

Procedures: `ADSPREADSHEET`, `ASPREADSHEET`, `RSPREADSHEET`, `VSPREADSHEET`, `FSPREADSHEET`.

Genstat Reference Manual 1 Summary section on: Regression analysis.

AU2RDA

Saves results from an unbalanced analysis of variance, by AUNBALANCED, in R data frames (R.W. Payne & Z. Zhang).

Options

TERM = *formula*	Treatment term whose means, effects etc. are to be saved; must be set if any of these are to be saved, unless there is only one treatment term
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
LSDLEVEL = *scalar*	Significance level (%) for least significant differences and multiple comparisons; default 5
RMETHOD = *string token*	Type of residuals to form (simple, standardized); default simp
MCOMPARISON = *string token*	Method to use to make multiple comparisons between the means (flsd, fstudentizedlsd, bonferroni, sidak); default * i.e. none
SAVE = *identifier*	Save structure (from AUNBALANCED) containing details of the analysis for which further output is required; if omitted, output is from the most recent use of AUNBALANCED

Parameters

INFORMATION = *string tokens*	What to save (aovtable, effects, means, semeans, esemeans, sedmeans, lsdmeans, replications, fittedvalues); must be set
OUTFILENAME = *texts*	Name of the R (.rda) file to create for each set of information; must be set
COLUMNNAMES = *texts*	Specifies names for the columns in the file; if this is not set, suitable names are chosen automatically
EXIT = *scalars*	Records the exit status, 0 if the information was saved successfully, 1 otherwise

Description

AU2RDA puts results from an analysis by AUNBALANCED into R data frames. By default the results are from the most recent AUNBALANCED analysis, but you can use the SAVE option to specify the save structure from some other analysis.

The INFORMATION parameter specifies the information to save, with settings:

aovtable	analysis of variance table;
effects	table of treatment effects;
means	table of predicted means;
semeans	table of standard errors of means;
esemeans	table of effective standard errors of means;
sedmeans	symmetric matrix of standard errors of differences of means;
lsdmeans	symmetric matrix of least significant differences of means;
replications	replication table;
fittedvalues	y-variate, fitted values and residuals.

The OUTFILENAME parameter specifies the name of the file to save the information. If this

does not have a `.rda` extension, one will be appended to the name automatically.

The `COLUMNNAMES` parameter allows you to specify a text containing names to use for the columns in the file. Thus may be useful if you want to avoid name clashes when you are saving several sets of output.

The `EXIT` parameter can save a scalar to record the exit status for each set of information. This contains zero if the information was saved successfully, and one otherwise. (For example, it will not be possible to save missing-value estimates if no responses were missing.)

The `TERM` option specifies the treatment term whose means, effects etc. are to be saved. This must be set if any of these are to be saved, unless there is only one treatment term.

The `LSDLEVEL` option specifies the significance level (%) to use in the calculation of least significant differences, or the experiment-wise significance level for multiple comparisons (default 5%).

The `MCOMPARISON` option specifies the method to use for multiple comparisons. The settings are `FLSD` (Fisher's Least Significant Difference), `FSTUDENTIZEDLSD` (Fisher's Protected Least Significant Difference, using the Studentized Range statistic rather than Student's t), `Bonferroni` and `Sidak`. By default, no multiple comparisons are done. The results are saved in a textual column, with the means. This contains identical letters (a, b etc.) alongside the sets of means that do not differ significantly. For more details see procedure `AMCOMPARISON`, which is used to do the calculations.

The `RMETHOD` option controls whether the residuals are simple residuals (like those printed by `ANOVA` – the default) or whether they are standardized according to their variances.

Options: `TERM`, `COMBINATIONS`, `ADJUSTMENT`, `LSDLEVEL`, `RMETHOD`, `MCOMPARISON`, `SAVE`.
Parameters: `INFORMATION`, `OUTFILENAME`, `COLUMNNAMES`, `EXIT`.

Action with `RESTRICT`

If the `Y` variate in the analysis is restricted, that restriction will carry over into the fitted-values spreadsheet.

See also

Procedures: `A2RDA`, `RXGENSTAT`.

Genstat Reference Manual 1 Summary sections on: Analysis of variance and Input and output.

AYPARALLEL

Does the same analysis of variance for several y-variates, and collates the output (R.W. Payne & D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output (summary, monitoring); default * i.e. none
TREATMENTSTRUCTURE = *formula*	Treatment formula for the analysis; if this is not set, the default is taken from the setting (which must already have been defined) of the TREATMENTSTRUCTURE directive
BLOCKSTRUCTURE = *formula*	Block formula for the analysis; if this is not set, the default is taken from any existing setting specified by the BLOCKSTRUCTURE directive and if neither has been set the design is assumed to be unstratified (i.e. to have a single error term)
COVARIATE = *variates*	Defines any covariates
FACTORIAL = *scalar*	Limit on the number of factors in a treatment term
SAVETERMS = *formula*	Treatment terms for which to save information; if this is not set, information is saved for all the treatment terms
REPLICATION = *pointer*	Pointer to tables saving the replication of the SAVETERMS
SPREADSHEET = *string tokens*	What results to save in spreadsheets (aov, means, vcmeans, effects, vareffects, seeffects, contrasts, secontrasts, tcontrasts, prcontrasts); default * i.e. none
CONTRASTSLIMIT = *scalar*	Limit on the order of a contrast of a treatment term; default 4
DEVIATIONSLIMIT = *scalar*	Limit on the number of factors in a treatment term for the deviations from its fitted contrasts to be retained in the model; default 9

Parameters

Y = *variates* or *pointers*	Y-variates for each analysis
VFACTOR = *factors*	Identifies the individual y-variates when they are supplied in a single Y variate
RESIDUALS = *variates* or *matrices*	Saves the residuals
FITTEDVALUES = *variates* or *matrices*	Saves the fitted values
MEANS = *pointers*	Pointer to a matrix for each of the SAVETERMS, saving the means from each analysis
VCMEANS = *pointers*	Pointer to matrices saving variances and covariances for the means
EFFECTS = *pointers*	Pointer to matrices saving effects
VAREFFECTS = *pointers*	Pointer to variates saving unit variances for effects
SEEFFECTS = *pointers*	Pointer to matrices saving effective standard errors of effects
TEFFECTS = *pointers*	Pointer to matrices saving t-statistics for effects
PREFFECTS = *pointers*	Pointer to matrices saving probabilities of t-statistics for effects
DF = *pointers*	Pointer to variates saving degrees of freedom

SS = *pointers*	Pointer to variates saving sums of squares
MS = *pointers*	Pointer to variates saving mean squares
RDF = *pointers*	Pointer to variates saving degrees of freedom for the residual corresponding to each of the SAVETERMS
RSS = *pointers*	Pointer to variates saving residual sums of squares
RMS = *pointers*	Pointer to variates saving residual mean squares
VR = *pointers*	Pointer to variates saving variance ratios
PRVR = *pointers*	Pointer to variates saving probabilities for the variance ratios
CONTRASTS = *pointers*	Pointer to matrices saving estimates of contrasts
SECONTRASTS = *pointers*	Pointer to matrices saving standard errors of contrasts
TCONTRASTS = *pointers*	Pointer to matrices saving t-statistics for contrasts
PRCONTRASTS = *pointers*	Pointer to matrices saving probabilities for t-statistics of contrasts
OUTFILENAME = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create

Description

The AYPARALLEL procedure does a "parallel" analysis of variance for several y-variates, combining and summarizing the information from all the analyses. The procedure operates most efficiently if the y-values for the analyses are in separate variates (with their units in identical orders). The variates should be placed into a pointer, which should then be used as the setting of the Y parameter of AYPARALLEL. The alternative format has the values for all the analyses in a single variate (which should again be used as the setting of the Y parameter). You must then also set the VFACTOR parameter, to a factor to indicate which values are involved in each analysis.

The BLOCKSTRUCTURE and TREATMENTSTRUCTURE options can specify block and treatment formulae (as in ordinary ANOVA) to define the models for the analysis of variance. If the TREATMENTSTRUCTURE option is not set, AYPARALLEL will use the model already defined by the TREATMENTSTRUCTURE directive, or will fail if that too has not been set. Similarly, if the BLOCKSTRUCTURE option is not set, AYPARALLEL will use the model (if any) previously defined by the BLOCKSTRUCTURE directive. The lengths of the block and treatment factors should be the same as the Y variate or variates. Furthermore, if there is a single Y variate, there must be a unique combimation of the levels of the block factors for every unit (this is required so that AYPARALLEL can check that the y-values are in the correct order for each analysis). The FACTORIAL option sets a limit on the number of factors in a treatment term, as in the ANOVA directive. Similarly the CONTRASTSLIMIT and DEVIATIONSLIMIT options operate as the CONTRASTS and DEVIATIONS options of ANOVA.

The COVARIATE option can list any covariates for the analyses; if this is unset, the default is taken from any existing setting defined by the COVARIATE directive. The lengths of the covariates should be the same as the Y variate or variates.

The RESIDUALS and FITTEDVALUES parameters can save the residuals and fitted values, respectively. These will each be in a pointer with a variate for each analysis if the y-values were specified in separate variates, or in a single variate if the y-values were combined in a single variate. The REPLICATION option saves a pointer containing the replication tables for the SAVETERMS. Parameters MEANS and EFFECTS save tables of means and effects from each analysis. The information is stored in a pointer with a matrix for each of the SAVETERMS. The matrices have a row for each analysis, and the columns are labelled to show how they correspond to the cells of the table. (Note that their ordering is the same as the order in which the contents of the REPLICATION table is stored.) Similarly SEEFFECTS saves effective standard errors for the effects, and VCMEANS saves the variances and covariances of the means. VAREFFECTS saves

a pointer of variates storing the unit variances of the effects, obtained by the VARIANCE parameter of AKEEP. Parameters DF, SS, MS, RDF, RSS, RMS, VR and PRVR store information from the analysis of variance table, in pointers with a variate for each term and a unit for each analysis. DF store the number of degrees of freedom for the relevant term (and analysis), SS stores sums of squares, MS stores mean squares, VR stores variance ratios, and PRVR the corresponding probabilities. Similarly the RDF parameter stores the number of degrees of freedom for the appropriate residual for the term, RSS stores the residual sums of squares, and RMS the residual mean square.

Printed output is controlled by the PRINT option, with settings:

monitoring	to print a running total of the number of analyses that have been analysed, and
summary	to print a summary of the significance levels found for the analyses for each of the SAVETERMS.

The SPREADSHEET option allows you to save various output components in spreadsheets. You can save these in either a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx), by setting the OUTFILENAME option to the name of the file to create. If the name is specified without a suffix, '.gwb' is added (so that a Genstat workbook is saved). If OUTFILENAME is not specified, they are put into a spreadsheet opened inside Genstat.

Options: PRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE, FACTORIAL, SAVETERMS, REPLICATION, SPREADSHEET, CONTRASTSLIMIT, DEVIATIONSLIMIT.

Parameters: Y, VFACTOR, RESIDUALS, FITTEDVALUES, MEANS, VCMEANS, EFFECTS, VAREFFECTS, SEEFFECTS, DF, SS, MS, RDF, RSS, RMS, VR, PRVR, CONTRASTS, SECONTRASTS, TCONTRASTS, PRCONTRASTS, OUTFILENAME.

Method

The analyses are performed by the ANOVA directive.

Action with RESTRICT

Any restrictions on the y-variates will be removed.

See also

Directive: ANOVA.

Procedures: MAANOVA, RYPARALLEL.

Genstat Reference Manual 1 Summary section on: Analysis of variance.

A2DISPLAY

Provides further output following an analysis of variance by A2WAY (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, information, covariates, effects, residuals, means, %cv, missingvalues); default *
FPROBABILITY = *string token*	Probabilities for variance ratio (yes, no); default no
PLOT = *string tokens*	Which residual plots to provide (fittedvalues, normal, halfnormal, histogram, absresidual); default *
GRAPHICS = *string token*	Type of graphs (lineprinter, highresolution); default high
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
PSE = *string tokens*	Types of standard errors to be printed with the means (differences, lsd, means, alldifferences, alllsd); default diff
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
RMETHOD = *string token*	Type of residuals to display (simple, standardized); default simp

Parameter

SAVE = *pointers*	Save structure (from A2WAY) for the analysis; if omitted, output is from the most recent A2WAY analysis

Description

The procedure A2WAY provides specialized facilities for analysis of variance with either one or two treatment factors. There can also be a blocking factor. It automatically determines the type of design and uses the appropriate method: the ANOVA directive if the design is balanced, or the regression directives (FIT, ADD and so on) if it is unbalanced.

Procedure A2DISPLAY allows you to display further output from the analysis. By default the output is from the most recent analysis performed by A2WAY. Alternatively, you can set the SAVE parameter to a save structure (saved using the SAVE parameter of A2WAY) to obtain output from an earlier analysis.

Printed output is controlled by the PRINT option, with settings:

aovtable	analysis-of-variance table (probabilities are given for the variance ratios if option FPROBABILITY=yes);
information	information about the design (non-orthogonality &c);
covariates	covariate regression coefficients;
effects	treatment parameters in the linear model;
means	table of means;
%cv	the coefficient of variation;
missingvalues	estimates for any missing values;
residuals	residuals and fitted values.

The PSE option controls the types of standard errors that are produced to accompany the tables of means, with settings:

differences	summary of standard errors for differences between pairs

	of means;
`alldifferences`	standard errors for differences between all pairs of means (unbalanced designs only);
`lsd`	summary of least significant differences between pairs of means;
`alllsd`	least significant differences between all pairs of means (unbalanced designs only);
`means`	standard errors of the means – for unbalanced designs, these are approximate effective standard errors formed by procedure `SED2ESE` with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula of $sed_{ij} = \sqrt{(ese_i^2 + ese_j^2)}$

The default is `differences`. The `LSDLEVEL` option sets the significance level (as a percentage) for the least significant differences.

For unbalanced designs (analysed by `A2WAY` using Genstat regression) the means are produced using the `PREDICT` directive. The first step (A) of the calculation forms the full table of predictions, classified by all the treatment and blocking factors. The second step (B) averages the full table over the factors that do not occur in the table of means. The `COMBINATIONS` option specifies which cells of the full table are to be formed in Step A. The default setting, `estimable`, fills in all the cells other than those that involve parameters that cannot be estimated. Alternatively, setting `COMBINATIONS=present` excludes the cells for factor combinations that do not occur in the data. The `ADJUSTMENT` option then defines how the averaging is done in Step B. The default setting, `marginal`, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting `equal` weights all the combinations equally. Finally, the setting `observed` uses the `WEIGHTS` option of `PREDICT` to weight each factor combination according to its own individual replication in the data.

The `PLOT` option allows up to four of the following residual plots to be requested:

`fittedvalues`	for a plot of residuals against fitted values;
`normal`	for a Normal plot;
`halfnormal`	for a half-Normal plot;
`histogram`	for a histogram of residuals; and
`absresidual`	for a plot of the absolute values of the residuals against the fitted values.

By default the first four are produced. The `GRAPHICS` option determines the type of graphics that is used, with settings `highresolution` (the default) and `lineprinter`.

The `RMETHOD` option controls whether simple or standardized residuals are printed or plotted; by default `RMETHOD=simple`.

Options: `PRINT`, `FPROBABILITY`, `PLOT`, `GRAPHICS`, `COMBINATIONS`, `ADJUSTMENT`, `PSE`, `LSDLEVEL`, `RMETHOD`.
Parameter: `SAVE`.

Method

`A2DISPLAY` uses `ADISPLAY` or `AUDISPLAY` when appropriate. Otherwise, it saves the information, using `AKEEP` or `RKEEP`, and prints the output in the required format.

Action with `RESTRICT`

If the `Y` variate in `A2WAY` was restricted, only the units not excluded by the restriction will have been analysed.

See also
Procedures: A2WAY, A2KEEP, A2RESULTSUMMARY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

A2KEEP

Copies information from an A2WAY analysis into Genstat data structures (R.W. Payne).

Options

FACTORIAL = *scalar*	Sets a limit on the number of factors in the terms formed from the TERMS formula; default 2
RESIDUALS = *variate*	Saves the residuals
FITTEDVALUES = *variate*	Saves the fitted values
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
AOVTABLE = *pointer*	To save the analysis-of-variance table as a pointer with a variate or text for each column (source, d.f., s.s., m.s. etc)
RMETHOD = *string token*	Type of residuals to form if the RESIDUALS option is set (simple, standardized); default simp
EXIT = *scalar*	Saves an exit code indicating the properties of the design
SAVE = *pointer*	Save structure (from A2WAY) for the analysis; if omitted, output is from the most recent A2WAY analysis

Parameters

TERMS = *formula*	Specifies the treatment terms whose means &c are to be saved
MEANS = *table* or *pointer* to *tables*	Saves tables of means for the terms or pointer to tables
SEMEANS = *table* or *pointer* to *tables*	Saves approximate effective standard errors of means
SEDMEANS = *table* or *pointer* to *tables*	Saves standard errors of differences between means
LSD = *table* or *pointer* to *tables*	Saves least significant differences

Description

The procedure A2WAY provides specialized facilities for analysis of variance with either one or two treatment factors. There can also be a blocking factor. It automatically determines the type of design and uses the appropriate method: the ANOVA directive if the design is balanced, or the regression directives (FIT, ADD and so on) if it is unbalanced.

Procedure A2KEEP allows you to copy information from the analysis into Genstat data structures. By default the information is from the most recent analysis performed by A2WAY. Alternatively, you can set the SAVE option to a save structure (saved using the SAVE parameter of A2WAY) to save information from an earlier analysis.

You can use the parameters of A2KEEP to save means, standard errors and least significant differences for the treatment main effects and interactions. The TERMS parameter should be set to a model formula to define the main effects and interactions whose means &c you want to save. The MEANS parameter can then save tables of means. The SEMEANS parameter saves their standard errors (also in a table). The SEDMEANS parameter saves standard errors for differences between the means (in a symmetric matrix), and the LSD parameter saves least significant differences (also in a symmetric matrix). The significance level for the least significant differences can be change from the default of 5% using the LSDLEVEL option. If you have a single term, you can supply a table or symmetric matrix for each of these parameters, as

appropriate. However, if you have several terms, you must supply a pointer which will then be set up to contain as many tables or symmetric matrices as there are terms. The LSDLEVEL option sets the significance level (as a percentage) for the least significant differences.

The FACTORIAL option sets a limit in the number of factors in the terms generated from the TERMS model formula. So

```
A2KEEP [FACTORIAL=1] A*B; MEANS=MA,MB
```

would save only the main effects of A and B. The option is provided for compatibility with the AKEEP directive. However, an alternative (and simpler) way of saving means only for the main effects would be to put

```
A2KEEP [FACTORIAL=1] A+B; MEANS=MA,MB
```

The default for FACTORIAL is 2.

For unbalanced designs (analysed by A2WAY using Genstat regression) the means are produced using the PREDICT directive. The first step (A) of the calculation forms the full table of predictions, classified by all the treatment and blocking factors. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data.

The RESIDUALS option can save the residuals from the analysis, and the FITTEDVALUES option can save the fitted values. The RMETHOD option controls whether simple or standardized residuals are saved; by default RMETHOD=simple. The AOVTABLE option saves the analysis-of-variance table, as a pointer with a variate or a text for each column of the table. The pointer elements are labelled with the column labels of the table, and the variates contain missing values where the table has blanks. These can be printed as blanks by setting option MISSING=' ' in the PRINT directive.

The EXIT option can save an exit code indicating how the analysis was done. For the exact meanings of the values see the ANOVA directive. Essentially, it has the values 0 or 1 if the analysis has been done using ANOVA (0 if design orthogonal and 1 if it is balanced). Other values indicate that it has been done using the regression directives.

Options: FACTORIAL, RESIDUALS, FITTEDVALUES, COMBINATIONS, ADJUSTMENT, LSDLEVEL, AOVTABLE, RMETHOD, EXIT, SAVE.
Parameters: TERMS, MEANS, SEMEANS, SEDMEANS, LSD.

Method

A2KEEP uses AKEEP and AUKEEP.

Action with RESTRICT

If the Y variate in A2WAY was restricted, only the units not excluded by the restriction will have been analysed.

See also
Procedures: A2WAY, A2DISPLAY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

A2PLOT

Plots effects from two-level designs with robust s.e. estimates (Eric D. Schoen & Enrico A.A. Kaul).

Options

PRINT = *string tokens*	Which ANOVA output to print, as in ADISPLAY; default aovt, effe
CHANNEL = *scalar*	What channel to use for anova and line-printer output; default * i.e. the current output channel
FACTORIAL = *scalar*	Limit for factorial expansion of TREATMENT formula; default 3
STRATUM = *formula*	Error strata from which Yates effects are to be plotted; if unset, plots are made for all the strata
GRAPHICS = *string token*	What type of graphics (highresolution, lineprinter); default high
TITLE = *strings*	Separate titles for each of the plots
METHOD = *string token*	Whether to make half-Normal or Normal plots (halfnormal, normal); default half
ROBUSTNESS = *string token*	Robustness of scale estimators against contamination with active effects (low, medium, high); default medi
ALPHALEVEL = *scalar*	Type I error (0.20, 0.15, 0.10, 0.05, 0.01); default 0.05
EXCLUDE = *scalars*	How many of the largest effects to withhold from each of the half-Normal plots; default 0

Parameters

Y = *variates*	Data to be analysed
EFFECTS = *pointers*	To save a variate for each error stratum containing the (sorted) Yates effects estimated there
SE = *pointers*	To save a scalar with the standard error of the Yates effects for each error stratum
SIGNIFICANT = *pointers*	To save formulae containing the significant Yates effects in each stratum

Description

Daniel (1959) shows how contrasts from two-level experiments in single or fractional replication can be evaluated through half-Normal plotting. Box *et al.* (1978) emphasize Normal plotting of the Yates effects. They suggest making separate plots for each error stratum. The Yates definition ensures that the effects from the same error stratum share a common variance. When there is sparsity of effects and Normality of error, most effects will come from a Normal distribution with zero mean and unknown variance. Inactive effects, plotted against quantiles of the Normal or half-Normal distribution, are roughly on a straight line through the origin. Effects not compatible with this line are designated active. Thus (half-)Normal plots will separate the few active effects from the inactive ones.

A well-known problem with the technique is the subjectivity as to which effects constitute the null-line. Many authors, therefore, have developed procedures for getting robust estimates of the standard errors of the Yates effects from unreplicated two-level experiments, see Haaland & O'Connell (1995) for an overview. Based on simulation results for 2^4 experiments (15 effects in the plot) the latter authors recommend three estimators according to a-priori ideas on the likely number of active effects (1-3, 4-6, and 7-8, respectively). The estimators are formed by (1) calculating an initial estimator of the standard error as a quantile of the full set of effects, multiplied with a consistency constant determined from the Normal distribution; (2) stripping

of potential active effects by retaining only effects smaller than a constant times the initial scale estimate; (3) multiplying some function of the remaining effects with a simulated consistency constant. One of the three recommended estimators is based on the median of the full set and the sum of squares of the retained effects; it is called the Adaptive Standard Error (ASE). The other two estimators are based on the median and the 45th percentile, respectively, of the full set; these are Pseudo Standard Errors (PSE). Both use the median of the retained effects. In general, ASE is less robust against contamination with active effects than PSE, because it uses all the effects below the cut-off point. The median-based PSE is obviously less robust than the PSE based on the 45th percentile.

Haaland & O'Connell (1995) suggest judging *t*-values from the effects and the calculated scale estimate against critical values determined by simulation. They present consistency constants for two of the recommended estimators and critical values for one of them, each for 7, 11, 15, 17, 23 and 31 effects, respectively. We have extended their results to the whole range from 7 up to 127 effects and all three estimators.

The treatment effects to be studied should be specified using the TREATMENTSTRUCTURE directive before using A2PLOT. They are grouped according to the error strata as specified by a previous BLOCKSTRUCTURE statement. Normal or half-Normal plots, according to the METHOD option, are made in either lineprinter or high-resolution quality (option GRAPHICS). By default plots are made for each error stratum. Alternatively, option STRATUM can be set to a formula defining the strata from which the Yates effects are to be plotted. The EXCLUDE option specifies the number of largest effects to be exclude from half-Normal plots (the option does not work with Normal plots). The titles of the plots can be provided using option TITLE. Setting METHOD=* suppresses the plots. Options FACTORIAL, PRINT and CHANNEL, are as in ADISPLAY. Note, however, that effects are printed as Yates effects, and that CHANNEL also controls the line-printer graphics.

When the number of effects in the plot is in the range 7 to 127, robust estimators are calculated for the standard error of the effects. The robustness of the estimators against contamination with active effects is specified through option ROBUSTNESS. A vertical line in the plot indicates the least significant Yates effect (LSE). The type I error is controlled by option ALPHALEVEL. Effects larger than the LSE are labelled in the plot.

The data variates are specified using the Y parameter. The EFFECTS parameter can save a pointer holding a variate of effects, sorted from small to large, for each error stratum. Effects are either the usual Yates effects (METHOD=normal) or their absolute values (METHOD=halfnormal). Parameter SIGNIFICANT can save a formula with the joint significant effects of all the strata. Parameter SE holds scalars with the standard errors of the effects in the respective strata.

Options: PRINT, CHANNEL, FACTORIAL, STRATUM, GRAPHICS, TITLE, METHOD, ROBUSTNESS, ALPHALEVEL, EXCLUDE.
Parameters: Y, EFFECTS, SE, SIGNIFICANT.

Method

A2PLOT accesses the current BLOCKSTRUCTURE and TREATMENTSTRUCTURE settings using the GET directive. If the STRATUM option is unset, separate plots for each of the strata are to be produced. A2PLOT checks, therefore, whether all strata are set explicitly. If this is not the case it augments the current BLOCKSTRUCTURE with a bottom stratum using procedure AFUNITS. If no BLOCKSTRUCTURE is set, it generates an explicit Units stratum and sets the BLOCKSTRUCTURE and STRATUM options to this stratum.

Yates effects for each stratum are saved using AKEEP. They are ordered and plotted against either Normal or half-Normal quantiles. Normal quantiles are calculated as

$$q_i = \text{NED}((i-0.375) / (n+0.25)) \quad i=1...n$$

Half-Normal quantiles are calculated as

q_i = NED(0.5 + 0.5 × (i−0.375) / (n+0.25)) i=1...n

For ROBUSTNESS=low, ASE based standard errors are calculated with the initial standard error calculated from the median of all effects, a cut-off of 2.5 times this value, and a final standard error from the sum of squares of the remaining effects. For ROBUSTNESS=medium, PSE based standard errors are calculated with the same cut-off as for ASE and a final standard error is calculated from the median of the remaining effects. For ROBUSTNESS=high, PSE based standard errors are calculated using the 45th percentile instead of the median for the initial estimate, and 1.25 instead of 2.5 as a multiplication factor to establish the cut-off. The final estimate uses the median of the retained effects.

Significant Yates effects are labelled in the half-Normal plots using the factor names from the TREATMENT statement.

Acknowledgements
The authors thank Peter Lane for suggesting and sketching procedure _A2PL_EXPAND.

Action with RESTRICT

AFUNITS (which may be called by A2PLOT if the STRATUM option is unset and no explicit bottom error stratum is specified in the current BLOCKSTRUCTURE setting) requires that none of the blocking factors be restricted.

References

Box, G.E.P., W.G. Hunter & J.S. Hunter (1978), *Statistics for Experimenters*. New York, Wiley.

Daniel, C. (1959), Use of half-normal plots in interpreting factorial two-level experiments. *Technometrics*, **1**, 311-342.

Haaland, P.D. & M.A. O'Connell (1995), Inference for effect-saturated fractional factorials. *Technometrics*, **37**, 82-93.

See also

Genstat Reference Manual 1 Summary section on: Analysis of variance.

A2RDA

Saves results from an analysis of variance in R data frames (R.W. Payne & Z.Zhang).

Options

TERM = *formula*	Treatment term whose means, effects etc. are to be saved; must be set if any of these are to be saved, unless there is only one treatment term
STRATUM = *formula*	Model term of the lowest stratum to be searched for effects and contrasts; default * implies the lowest stratum
SUPPRESSHIGHER = *string token*	Whether to suppress the searching of higher strata if a term is not found in STRATUM (yes, no); default no
LSDLEVEL = *scalar*	Significance level (%) for least significant differences and multiple comparisons; default 5
EQFACTORS = *factors*	Factors whose levels are to be assumed to be equal within the comparisons between means calculated for effective standard errors of treatment means
RMETHOD = *string tokens*	Types of residuals to form (simple, standardized, combined); default simp
MCOMPARISON = *string token*	Method to use to make multiple comparisons between the means (tukey, regwmr, duncan, scheffe, fplsd, fulsd, fpstudentizedlsd, fustudentizedlsd, bonferroni, sidak); default * i.e. none
SAVE = *ANOVA save structure*	Specifies the analysis from which to save the results; default * i.e. most recent one

Parameters

INFORMATION = *string tokens*	What to save (aovtable, covariates, effects, cbeffects, partialeffects, contrasts, means, semeans, sedmeans, lsdmeans, dfmeans, cbmeans, secbmeans, sedcbmeans, replications, fittedvalues, missingvalues, stratumvariances, %cv, fixedcoefficients, randomcoefficients); must be set
OUTFILENAME = *texts*	Name of the R (.rda) file to create for each set of information; must be set
COLUMNNAMES = *texts*	Specifies names for the columns in the file; if this is not set, suitable names are chosen automatically
EXIT = *scalars*	Records the exit status, 0 if the information was saved successfully, 1 otherwise

Description

A2RDA puts results from an analysis of variance into R data frames. By default the results are from the most recent ANOVA, but you use the SAVE option to specify the save structure from some other analysis.

The INFORMATION parameter specifies the information to save, with settings:

aovtable	analysis of variance table;
covariates	estimated covariate regression coefficients and their standard errors (if any covariates in the analysis);
effects	table of treatment effects;
cbeffects	table of effects for a treatment term, combining the

	information from every stratum where it is estimated;
partialeffects	table of partial effects for a treatment term, adjusted for every other treatment term;
contrasts	estimated treatment contrasts;
means	table of predicted means;
semeans	table of effective standard errors of means;
sedmeans	symmetric matrix of standard errors of differences of means;
lsdmeans	symmetric matrix of least significant differences of means;
dfmeans	symmetric matrix of degrees of freedom for differences of means;
cbmeans	table of predicted means for a treatment term, combining information from all the strata in which its effects are estimated;
secbmeans	table of effective standard errors of combined treatment means;
sedcbmeans	symmetric matrix of standard errors of differences of combined treatment means;
replications	replication table;
fittedvalues	y-variate, fitted values and residuals;
missingvalues	estimates for missing values (if any);
stratumvariances	estimated variances of the units in each stratum, stratum variance components and effective numbers of degrees of freedom;
%cv	coefficients of variation, numbers of residual degrees of freedom and standard errors of individual units in each stratum;
fixedcoefficients	estimated treatment effects and covariate regression coefficients;
randomcoefficients	residuals (from all the strata).

The OUTFILENAME parameter specifies the name of the file to save the information. If this does not have a .rda extension, one will be appended to the name automatically.

The COLUMNNAMES parameter allows you to specify a text containing names to use for the columns in the file. Thus may be useful if you want to avoid name clashes when you are saving several sets of output.

The EXIT parameter can save a scalar to record the exit status for each set of information. This contains zero if the information was saved successfully, and one otherwise. (For example, it will not be possible to save missing-value estimates if no responses were missing.)

The TERM option specifies the treatment term whose means, effects etc. are to be saved. This must be set if any of these are to be saved, unless there is only one treatment term. The EQFACTORS option allows you to specify factors within the tables of means whose levels are assumed to be equal for the two means when forming effective standard errors of treatment means (see the AKEEP directive for more details).

In designs where there is partial confounding, and treatment terms are estimated in more than one stratum, options STRATUM and SUPPRESSHIGHER allow you to specify the strata from which effects and contrasts are taken. By default, Genstat searches all the strata, and takes the information from the lowest of the strata where the term is estimated. If you set the STRATUM option, only strata down to the specified stratum are searched. By setting SUPPRESSHIGHER=yes, you can restrict the search to only that stratum.

The LSDLEVEL option specifies the significance level (%) to use in the calculation of least significant differences, or the experiment-wise significance level for multiple comparisons

(default 5%).

The MCOMPARISON option specifies the method to use for multiple comparisons. The settings are Tukey, REGWMR (Ryan/Einot-Gabriel/Welsch multiple range test), Duncan, Scheffe, FPLSD (Fisher's Protected Least Significant Difference), FULSD (Fisher's Unprotected Least Significant Difference), FPSTUDENTIZEDLSD (Fisher's Protected Least Significant Difference, using the Studentized Range statistic rather than Student's t), FUSTUDENTIZEDLSD (Fisher's Unprotected Least Significant Difference, using the Studentized Range statistic), Bonferroni and Sidak. By default, no multiple comparisons are done. The results are saved in a textual column, with the means. This contains identical letters (a, b etc.) alongside the sets of means that do not differ significantly. For more details see procedure AMCOMPARISON, which is used to do the calculations.

The RMETHOD option selects the types of residual to save with the fitted values, with settings:

simple	ordinary "simple" residuals (labelled residual in the data frame),
standardized	residuals standardized according to their variances (labelled stdresidual), and
combined	residuals that incorporate the variation from all the strata, (labelled cbresidual).

By default, only simple residuals are saved. You can save more than one type of residual, by specifying a list of settings (and they are included in the data frame in the order with which they occur in the list).

Options: TERM, STRATUM, SUPPRESSHIGHER, LSDLEVEL, EQFACTORS, RMETHOD, MCOMPARISON, SAVE.
Parameters: INFORMATION, OUTFILENAME, COLUMNNAMES, EXIT.

Action with RESTRICT

If the Y variate in the analysis is restricted, that restriction will carry over into the fitted-values spreadsheet.

See also

Procedure: RXGENSTAT.
Genstat Reference Manual 1 Summary sections on: Analysis of variance and Input and output.

A2RESULTSUMMARY

Provides a summary of results from an analysis by A2WAY (R.W. Payne).

Options

PRINT = *string tokens*	What to print (description, means, significant); default desc, mean, sign
PSE = *string tokens*	Standard errors to be printed with the means (sed, sedsummary, lsd, lsdsummary, dfmeans); default sed, dfme
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
SAVE = *pointer*	Save structure from A2WAY; default uses the save structure from the most recent A2WAY analysis

No parameters

Description

A2RESULTSUMMARY provides information from an A2WAY analysis that would be useful for a report. By default, all the information is printed, but you can control this with the PRINT option, whose settings are:

description	prints the name of the y-variate, any covariates and the block and treatment models,
means	prints relevant tables of means, and
significant	lists the significant treatment terms.

The relevant tables of means are those that contain significant treatment effects. If the interaction is significant in an analysis with two treatment factors, the relevant table is just the two-way table of means. Otherwise the relevant tables consist of the one-way tables of means for any significant main effect.

The PSE option controls the information provided with the tables of means:

sed	standard errors for differences between means,
sedsummary	summary of the standard errors for differences,
dfmeans	degrees of freedom for the standard errors of differences,
lsd	least significant differences between the means, and
lsdsummary	summary of the least significant differences.

The default is to print the standard errors of differences and their degrees of freedom.

The LSDLEVEL option specifies the significance level (%) to use in the calculation of least significant differences (default 5%).

Options: PRINT, PSE, LSDLEVEL, SAVE.
Parameters: none.

See also

Procedure: A2WAY, A2DISPLAY, ARESULTSUMMARY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

A2WAY

Performs analysis of variance of a balanced or unbalanced design with up to two treatment factors (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output from the analysis (aovtable, information, covariates, effects, residuals, means, %cv, missingvalues); default aovt, mean
TREATMENTS = *factors*	Defines either one or two treatment factors
BLOCKS = factor	Can specify a blocking factor e.g. for a randomized block design
COVARIATES = *variates*	Specifies any covariates
FACTORIAL = *scalar*	Can be set to 1 to fit only the main effects of the treatments factors; default 2 also fits their interaction
FPROBABILITY = *string token*	Probabilities for variance ratio (yes, no); default no
PLOT = *string tokens*	Which residual plots to provide (fittedvalues, normal, halfnormal, histogram, absresidual); default fitt, norm, half, hist
GRAPHICS = *string token*	Type of graphs (lineprinter, highresolution); default high
COMBINATIONS = *string token*	Factor combinations for which to form predicted means (present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when predicting means (marginal, equal, observed); default marg
PSE = *string tokens*	Types of standard errors to be printed with the means (differences, lsd, means, alldifferences, alllsd); default diff
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
RMETHOD = *string token*	Type of residuals to save or display (simple, standardized); default simp
MVINCLUDE = *string token*	Whether to include units with missing y-values when using ANOVA (yvariate); default * i.e. not included
EXIT = *scalar*	Saves an exit code indicating the properties of the design

Parameters

Y = *variates*	Each of these contains the data values for an analysis
RESIDUALS = *variates*	Saves the residuals from each analysis
FITTEDVALUES = *variates*	Saves the fitted values from each analysis
SAVE = *pointers*	Save structure for each analysis (to use in A2DISPLAY or A2KEEP)

Description

A2WAY provides specialized facilities for analysis of variance with either one or two treatment factors. There can also be a blocking factor. It automatically determines the type of design and uses the appropriate method: the ANOVA directive if the design is balanced, or the regression directives (FIT, ADD and so on) if it is unbalanced. So, for example, it can analyse randomized complete block designs with one or two treatment factors, or unbalanced two-way treatment structures with or without blocking, or designs with a single treatment factor whose levels are allocated unevenly across the blocks. By default, any units with missing values in the y-variate are excluded from the analysis. Conversely, in ANOVA they need to be included to ensure balance

in the more general situations that it covers, and are estimated as part of the analysis. However, you can reproduce the ANOVA analysis by setting option MVINCLUDE=yvariate.

The output is also customized. For example, if the treatments have unequal replication, a standard error is printed for each mean, rather than the summary for comparisons of means with minimum and maximum replication as given by ANOVA. Similarly, the two-way analyses show the sums of squares for the main effects both omitting and ignoring the other main effect. In addition, A2WAY provides residual plots directly, instead of requiring you to use procedure APLOT after the analysis.

The Y parameter supplies a variate containing the data values to be analysed. The RESIDUALS parameter can save the residuals from the analysis, and the FITTEDVALUES parameter can save the fitted values. The RMETHOD option controls whether simple or standardized residuals are saved or displayed; by default RMETHOD=simple.

The SAVE parameter can save a "save" structure that can be used as input to procedure A2DISPLAY to produce further output, or to procedure A2KEEP to copy output into Genstat data structures.

The treatment factor or factors are specified by the TREATMENTS option, and the BLOCKS option can be used to supply a blocking factor. Covariates can be supplied using the COVARIATES option. As in ANOVA, the FACTORIAL option sets a limit in the number of factors in each treatment term. So you can set FACTORIAL=1 to fit only the main effects when there are two treatment factors; the default FACTORIAL=2 also fits their interaction.

Printed output is controlled by the PRINT option, with settings:

aovtable	analysis-of-variance table (probabilities are given for the variance ratios if option FPROBABILITY=yes);
information	information about the design (non-orthogonality &c);
covariates	covariate regression coefficients);
effects	treatment parameters in the linear model;
means	table of means;
%cv	to print the coefficient of variation;
missingvalues	to print estimates for any missing values.

The PSE option controls the types of standard errors that are produced to accompany the tables of means, with settings:

differences	summary of standard errors for differences between pairs of means;
alldifferences	standard errors for differences between all pairs of means (unbalanced designs only);
lsd	summary of least significant differences between pairs of means;
alllsd	least significant differences between all pairs of means (unbalanced designs only);
means	standard errors of the means – for unbalanced designs, these are approximate effective standard errors formed by procedure SED2ESE with the aim of allowing good approximations to the standard errors for differences to be calculated by the usual formula of $sed_{ij} = \sqrt{(ese_i^2 + ese_j^2)}$

The default is differences. The LSDLEVEL option sets the significance level (as a percentage) for the least significant differences.

For unbalanced designs, analysed using Genstat regression, the means are produced using the PREDICT directive. The first step (A) of the calculation forms the full table of predictions, classified by all the treatment and blocking factors. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in

all the cells other than those that involve parameters that cannot be estimated. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data.

The PLOT option allows up to four of the following residual plots to be requested:

fittedvalues	for a plot of residuals against fitted values;
normal	for a Normal plot;
halfnormal	for a half-Normal plot;
histogram	for a histogram of residuals; and
absresidual	for a plot of the absolute values of the residuals against the fitted values.

By default the first four are produced. The GRAPHICS option determines the type of graphics that is used, with settings highresolution (the default) and lineprinter.

The RMETHOD option controls whether simple or standardized residuals are printed or plotted; by default RMETHOD=simple.

The EXIT option can save an exit code indicating how the analysis was done. For the exact meanings of the values see the ANOVA directive. Essentially, it has the values 0 or 1 if the analysis has been done using ANOVA (0 if design orthogonal and 1 if it is balanced). Other values indicate that it has been done using the regression directives.

Options: PRINT, TREATMENTS, BLOCKS, COVARIATES, FACTORIAL, FPROBABILITY, PLOT, GRAPHICS, COMBINATIONS, ADJUSTMENT, PSE, LSDLEVEL, RMETHOD, MVINCLUDE, EXIT.
Parameters: Y, RESIDUALS, FITTEDVALUES, SAVE.

Method

The EXIT option of the ANOVA directive is used to determine whether or not the design is unbalanced (and thus whether the Genstat regression facilities need to be used rather than the analysis of variance facilities).

Action with RESTRICT

If the Y variate is restricted, only the units not excluded by the restriction will be analysed.

See also

Procedures: A2DISPLAY, A2KEEP, A2RESULTSUMMARY.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

BACKTRANSFORM

Calculates back-transformed means with approximate standard errors and confidence intervals (V.M. Cave).

Options

`PRINT` = *string tokens*	Controls printed output (`description`, `means`, `backmeans`); default `desc`, `back`
`PLOT` = *string tokens*	The confidence intervals of the back-transformed means to plot (`backtransformed`, `approximate`, `both`); default `*` i.e. none
`TRANSFORMATION` = *string tokens*	Transformation (`identity`, `logarithm`, `log10`, `logit`, `squareroot`, `reciprocal`, `power`, `probit`, `complementaryloglog`,`logratio`, `angular`, `arcsinesquareoot`, `calculated`); default `iden` (i.e. no transformation)
`CLOG` = *scalar*	Constant *c* for the `logarithm` and `log10` transformations, in form log(mean+*c*); default 0
`EXPONENT` = *scalar*	Exponent for `power` transformation; default −2
`KLOGRATIO` = *scalar*	Parameter *k* for `logratio` transformation, in form log(mean/(mean+*k*)); default 1
`BACKTRANSFORMATION` = *expression*	Expression, formed using argument `Y`, that defines the inverse of the transformation; must be specified when `TRANSFORMATION = calculated`
`DERIVATIVE` = *expression*	Expression, formed using argument `Y`, that defines the first derivative of the transformation; must be specified when `TRANSFORMATION = calculated`
`CIPROBABILITY` = *scalar*	Probability for the confidence intervals; default 0.95
`DIRECTION` = *string tokens*	Order in which the back-transformed means are plotted (`ordinal`, `ascending`, `descending`); default `ordi`
`USEPENS` = *string tokens*	Whether to use the current pen definitions for plotting; (`yes`, `no`); default `no`
`WINDOW` = *scalar*	Window to use for plot; default 3

Parameters

`MEANS` = *tables*, *variates* or *scalars*	Supplies the transformed mean(s)
`SEMEANS` = *tables*, *variates* or *scalars*	Supplies the standard error(s) of the transformed mean(s)
`DF` = *scalars*	Degrees of freedom to construct the confidence intervals; default `*`
`DECIMALS` = *scalars*	Number of decimal places for printing; default `*`
`BACKTRANSFORMEDMEANS` = *tables*, *variates* or *scalars*	Saves the back-transformed means
`SEBACKTRANSFORMEDMEANS` = *tables*, *variates* or *scalars*	Saves the approximate standard errors for the back-transformed means
`CIAPPROXIMATE` = *pointers*	Saves the approximate confidence intervals for the back-transformed means
`CIBACKTRANSFORMED` = *pointers*	Saves the back-transformed confidence intervals for the back-transformed means

TITLE = *texts*	Title for plot; default * i.e. none
YTITLE = *texts*	Title for y-axis; default * i.e. none
XTITLE = *texts*	Title for x-axis; default * i.e. formed automatically

Description

BACKTRANSFORM calculates back-transformed means, with approximate standard errors and confidence intervals. The means and corresponding standard errors, for back-transforming, are supplied using the MEANS and SEMEANS parameters, respectively, as either tables, variates or scalars. If MEANS supplies a table or variate, SEMEANS can be either of the same type or a scalar, whereas if MEANS supplies a scalar, SEMEANS must be a scalar.

The degrees of freedom, used to construct the confidence intervals, can be set using the DF parameter. If these are not supplied, *z*-scores are used to form the confidence intervals. The probability for the confidence intervals is specified by the CIPROBABILITY option; the default 0.95 gives 95% confidence intervals.

The function that was used to transform the data prior to analysis is specified using the TRANSFORMATION option; the default takes the identity link (i.e. no transformation). The natural logarithm, log10, logit, square root, reciprocal, power, probit, complementary log-log, log-ratio and angular (or arcsine-square root) functions are provided directly by the TRANSFORMATION option. The angular and arcsinesquareroot transformations are synonyms, and the transformed data are assumed to be in radians.

You can also define your own transformation by setting TRANSFORMATION = calculated and providing expressions to calculate the inverse and first derivative of the transformation, using the BACKTRANSFORMATION and DERIVATIVE options, respectively. The calculations are specified in terms of the argument Y. Thus, for example, the logarithm transformation could be specified by setting options

```
BACKTRANSFORMATION=!E(exp(Y))
```

and

```
DERIVATIVE=!E(1/Y).
```

The CLOG option sets the constant *c* used by the logarithm and log10 transformations (default 0), the EXPONENT option sets the exponent used by the power transformation (default -2) and the KLOGRATIO option sets the parameter *k* used by the logratio transformation (default 1).

Back-transformation from the logit, probit, angular or arcsinesquareroot scale returns proportions (rather than percentages).

The PLOT option allows you to request plots of the results, using settings:

backtransformed	for back-transformed means and back-transformed confidence intervals (i.e. confidence intervals that maintain exactly the percent coverage on the transformed scale),
approximate	for back-transformed means with approximate confidence intervals (i.e. derived from the approximate standard errors), and
both	for back-transformed means with both types of confidence interval.

By default no plots are produced.

The TITLE, YTITLE and XTITLE parameters can supply an overall title, a y-axis title and a x-axis title for the plot, respectively. By default, neither an overall title nor a y-axis title is displayed. The default for the x-axis, when MEANS supplies the transformed means as a table, is to use the identifiers of the table's classifying factors to form a title for the x-axis. If MEANS supplies a variate or scalar, the default is not to display an x-axis title. To omit the x-axis title, you can supply a blank string i.e.

```
XTITLE=' '
```

By default, the pen attributes used for plotting are determined automatically within the procedure. However, you can set USEPENS to yes, to request that the current COLOURS, CFILL, SYMBOLS, SMSYMBOL and THICKNESS pen definitions of pens 1 and 2 are used. Pen 1 controls the colour, symbol, symbol size and line thickness used to plot the back-transformed means with back-transformed confidence intervals, whereas Pen 2 controls these attributes when the back-transformed means are plotted with approximate confidence intervals. The WINDOW option specifies the window used for plotting; default 3.

By default, the back-transformed means are plotted in order of their ordinal level, however you can use the DIRECTION option to request that they are plotted in ascending or descending numerical value instead.

The BACKTRANSFORMEDMEANS, SEBACKTRANSFORMEDMEANS parameters can save back-transformed means, approximate standard errors, in data structures of the same type as MEANS. The CIAPPROXIMATE and CIBACKTRANSFORMED parameters can save pointers of approximate confidence intervals and back-transformed confidence intervals, respectively. The pointers contain two data structures elements, of the same type as MEANS, storing the lower and the upper confidence limits, respectively.

Printed output is controlled by the PRINT option, with settings:

description	provides a description of the output,
means	prints the transformed means, with their standard errors and confidence intervals, and
backmeans	prints the back-transformed means, with their approximate standard errors and confidence intervals.

You can set the number of decimals places to appear in the printed output, using the DECIMALS parameter.

Options: PRINT, PLOT, TRANSFORMATION, CLOG, EXPONENT, KLOGRATIO, BACKTRANSFORMATION, DERIVATIVE, CIPROBABILITY, DIRECTION, USEPENS, WINDOW.
Parameters: MEANS, SEMEANS, DF, DECIMALS, BACKTRANSFORMEDMEANS, SEBACKTRANSFORMEDMEANS, CIPAPPROXIMATE, CIBACKTRANSFORMED, TITLE, YTITLE, XTITLE.

Method

BACKTRANSFORM uses a first-order Taylor series expansion, to obtain approximate standard errors for the back-transformed means. The methodology is described in Jørgensen & Pedersen (1998). In brief, let $\hat{u}$ denote the estimated mean on the transformed scale, se_u its standard error, $g()$ the transformation function and $g'()$ the first derivative of the transformation function. On the back-transformed scale the estimated mean ($\bar{y}$) and standard error ($se_{\bar{y}}$) are approximated by $g^{-1}(\hat{u})$ and $se_u / \mathrm{mod}(g'(\bar{y}))$, respectively.

A back-transformed confidence interval is given by

$$(g^{-1}(\hat{u} - t \times se_u), g^{-1}(\hat{u} + t \times se_u))$$

where t is the upper (1 − CIPROBABILITY)/2 critical value for the t distribution. Note that, when the degrees of freedom have not been set using the DF parameter, the z-score (i.e. the Normal distribution) is used to construct the confidence interval instead of the t-value. For the logratio and reciprocal transformations, when the estimated mean and confidence limit on the transformed scale lie on different sides of the singularity in the inverse function of the transformation, the back-transformed confidence limit is set to a missing value.

An approximate confidence interval

$$(\bar{y} - t \times se_{\bar{y}}, \bar{y} + t \times se_{\bar{y}})$$

is also provided. However, this should be used only to evaluate the validity of the approximate standard error. If the back-transformed and approximate confidence intervals differ greatly, the

approximate standard error is inadequate.

Action with `RESTRICT`

`BACKTRANSFORM` will work when `MEANS` and/or `SEMEANS` supplies a restricted variate; however, if both variates are restricted they must be restricted in the same way. Furthermore, their unrestricted length must be the same.

References

Jørgensen, E., & Pedersen, A.R. (1998). How to obtain those nasty standard errors from transformed data – and why they shouldn't be used. Biometry Research Unit, Danish Institute of Agricultural Sciences.

See also

Directives: `AKEEP`, `PREDICT`, `VPREDICT`.
Procedures: `AFMEANS`, `LSIPLOT`.

BAFFYMETRIX

Estimates expression values from an Affymetrix CED and CDF file (D.B. Baird).

Options

METHOD = *string token*	Method for calculating probe expression values (mas4, mas5, rma, rma2); default rma
TRANSFORMATION = *string*	How to transform the data (log2, none); default none when METHOD=mas4, otherwise log2

Parameters

CELFILES = *texts*	Affymetrix CEL files
CDFFILE = *texts*	Associated CDF file
GSHFILE = *texts*	Genstat spreadsheet file containing the estimated expression values, together with the associated slide and probe information

Description

BAFFYMETRIX estimates expression values for Affymetrix data. It operates in a "batch" mode, in which each set of CEL files and associated CDF file are loaded into the server, and processed automatically to generate a summary spreadsheet containing the estimates together with the associated slide and probe information.

The METHOD option selects the method to use to summarize over the PM and MM pairs, with settings:

rma	Robust Means Analysis model – the probe level model introduced by Irizarry *et al*. (2003) which only uses PM information and transforms the values based on a kernel density estimate of the PM distribution;
rma2	Robust Means Analysis 2 – an adaptation of RMA algorithm which fits the kernel density to a truncated distribution of the PM values, with the truncation point based on an initial kernel density estimate;
mas4	Affymetrix Version 4 – the AvDiff algorithm introduced in the Affymetrix version 4 software; and
mas5	Affymetrix Version 5 – the Tukey biweight algorithm introduced in the Affymetrix version 5 software.

In the Affymetrix MAS 4 and 5 methods, the difference between the signals (PM – MM) is averaged using a robust averaging method. The MAS 4 algorithm uses the AvDiff algorithm which discards the minimum and maximum difference, and any differences greater than 3 standard deviations from the mean. The MAS 5 algorithm uses the Tukey biweight algorithm which reweights the values depending on how far they are from the median, and discards any that are more than 5 times the median absolute distance away. The MAS 5 algorithm also replaces the MM value with a value known as an Ideal Mismatch (IM), which is always less than the PM value.

The TRANSFORMATION option controls whether the PM and MM values are transformed to logarithms base 2. The default does the transformation only for METHOD = mas5, rma or rma2.

Options: METHOD, TRANSFORMATION.
Parameters: CELFILES, CDFFILE, GSHFILE.

Reference

Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U. & Speed,

T.P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, Number 2, 249-264.

See also
Procedure: AFFYMETRIX.
Genstat Reference Manual 1 Summary section on: Microarray data.

BANK

Calculates the optimum aspect ratio for a graph (J. Ollerton & S.A. Harding).

Option

WINDOW = *scalar*	Window number; default 1

Parameters

Y = *variates*	Vertical coordinates
X = *variates*	Horizontal coordinates
ASPECTRATIO = *scalars*	Store the calculated aspect ratios

Description

BANK calculates the aspect ratio for a graph from the data to be plotted, specified as *y* and *x* variates. A window is set up using the FRAME directive so that the *y* and *x* dimensions are in the correct ratio; by default the window used is window 1, but this can be changed using the WINDOW option. The new window bounds are guaranteed to be within the existing definition; if the new aspect ratio is less than the current value the *y* upper bound of the window is reduced by an appropriate amount, otherwise the *x* upper bound is reduced. The aspect ratio can be saved using the parameter ASPECTRATIO.

Option: WINDOW. Parameters: Y, X, ASPECTRATIO.

Method

The aspect ratio is calculated from the X and Y variates of data using the Median-absolute-slope algorithm which centres the orientations of the individual line segments on 45 degrees. The specified window is resized to obtain the optimum aspect ratio within the plotting area; the existing axis margin sizes are preserved.

Action with RESTRICT

The Y and X variates can be restricted, however this restriction must be identical for both variates. Existing restrictions will not be altered.

References

Cleveland, W.S. (1987). Graphical Perception: The visual decoding of Quantitative Information on graphical displays of data. *Journal of the Royal Statistical Society, Series A*, **150**, 192-229.

Cleveland, W.S. (1993). *Visualizing Data*. Hobart Press, Summit, New Jersey.

See also

Directive: FRAME.

Genstat Reference Manual 1 Summary section on: Graphics.

BASELINE

Estimates a baseline for a series of numbers whose minimum value is drifting. (D.B. Baird).

Options

PLOT = *string token*	Whether to plot the series and the fitted baseline (baseline); default * i.e. no plot
BANDWIDTH = *scalar*	Bandwidth for the moving minimum; default 50
WINDOW = *scalar*	Window number for the plot; default 1
KEYWINDOW = *scalar*	Window for the key (zero for no key); default 2

Parameters

Y = *variates*	Series whose baseline is to be estimated
NEWY = *variates*	Saves the y-values corrected to a zero baseline
BASELINE = *variates*	Saves the estimated baseline
TITLE = *text*	Title for the plot

Description

BASELINE is useful in the situation where you have a series of observations that are assumed to be fluctuating above and then back down to a baseline. Often the baseline may be drifting, and this will need to be corrected if, for example, you want to identify peaks and their heights.

The series is supplied, in a variate, by the Y parameter. The corrected values can be saved using the NEWY parameter, and the estimated baseline can be saved using the BASELINE parameter (both in variates). The baseline is estimated by taking the maximum of a moving minimum over a bandwidth specified by the BANDWIDTH option (default 50). If you want to detect peaks, the bandwidth should be greater than their anticipated width.

You can set option PLOT=baseline to plot the series and the estimated baseline. The WINDOW option specifies the window to use for the plot (default 1), and the KEYWINDOW option specifies the window for the key (default 2). You can supply a title for the plot using the TITLE parameter; the default title is "Moving minimum (*b*) baseline fitted to *y*" where *b* is the bandwidth, and *y* is the identifier of the Y variate.

Options: PLOT, BANDWIDTH, TITLE, WINDOW, KEYWINDOW.
Parameters: Y, NEWY, BASELINE.

Action with RESTRICT

Any restrictions on the Y variate are ignored.

See also

Procedures: ALIGNCURVE, PEAKFINDER.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

BBINOMIAL

Estimates the parameters of the beta binomial distribution (D.M. Smith).

Options

PRINT = *string tokens*	Controls printed output (estimates, loglikelihood); default esti
MAXCYCLE = *scalar*	Maximum number of iterations; default 50
TOLERANCE = *scalar*	Convergence criterion; default 10^{-5}

Parameters

RBINOMIAL = *variates*	Numerator of binomial data
NBINOMIAL = *variates*	Denominator of binomial data or scalars
MU = *scalars*	Mean, expectation of underlying beta distribution
THETA = *scalars*	Shape-determining parameter of underlying beta distribution
SEMU = *scalars*	Standard error of mu
SETHETA = *scalars*	Standard error of theta
LOGLIKELIHOOD = *scalars*	Log likelihood
NCYCLES = *scalars*	Number of iterations
EXIT = *scalars*	Indicator of faults

Description

BBINOMIAL estimates the two parameters of the beta binomial distribution by maximum likelihood, using the methods of Smith (1983) and Smith & Ridout (1995). The parameters mu and theta are estimated instead of the standard alpha and beta, as they are more "stable" i.e. reliable to estimate. (See Williams 1975.) The relationship is that

mu = alpha / (alpha + beta)
theta = 1 / (alpha + beta)

or

alpha = mu / theta
beta = (1 – mu) / theta

(Note: in the descriptions of the probability functions, PRBETA etc, alpha and beta are represented as a and b.)

The numbers responding are specified in a variate using the RBINOMIAL parameter, and the corresponding total numbers are specified by the NBINOMIAL parameter in either a variate or a scalar.

Printed output is controlled by the PRINT option, with settings:

estimates	to print the estimated values of mu and theta, together with the corresponding values of alpha and beta, and
loglikelihood	to print the log-likelihood.

The estimates of the two parameters of the distribution can be saved by the parameters MU and THETA, and their standard errors can be saved by parameters SEMU and SETHETA. The LOGLIKELIHOOD parameter can save the value of the log-likelihood.

The NCYCLES parameter can save the number of iterations that were needed. The MAXCYCLE option sets a limit on the total number of iterations (default 50), and the TOLERANCE option sets the convergence criterion (default 10^{-5}). The EXIT parameter can save a scalar to indicate the success or failure of the estimation, as follows.

0 success.
1 a value of NBINOMIAL is less than or equal to 1.
2 all values of RBINOMIAL are zero.
3 all values of RBINOMIAL are equal to NBINOMIAL.

4 a value of RBINOMIAL is greater than NBINOMIAL.
5 some values of either RBINOMIAL or NBINOMIAL are less than zero.
6 if either MU went outside range 0 to 1 or THETA went outside range 0 to *infinity*, where *infinity* is the value (10^6) set inside BBINOMIAL to represent infinity.
7 if the maximum number of iterations (MAXCYCLE) was exceeded.
8 if the damped Newton-Raphson procedure failed.
9 if the minimum value for THETA has been reached and the maximum likelihood estimate of MU found, but moving THETA away from the minimum value slightly increases the log likelihood. The estimate of MU returned is the estimate on the minimum value of THETA. The estimates are not then the overall maximum likelihood estimates.

When EXIT = 1, 2, 3, 4 or 5 BBINOMIAL gives a fault, and MU, THETA, SEMU, SETHETA and LOGLIKELIHOOD are undefined. When EXIT = 6, 7, 8 or 9 BBINOMIAL gives a warning, and MU, THETA and LOGLIKELIHOOD are returned with their current values, while SEMU and SETHETA contain missing values. When EXIT = 6 the out-of-range parameter is set to the appropriate limiting value.

Options: PRINT, MAXCYCLE, TOLERANCE.
Parameters: RBINOMIAL, NBINOMIAL, MU, THETA, SEMU, SETHETA, LOGLIKELIHOOD, NCYCLES, EXIT.

Method
For full details of the methods implemented in these procedures see Smith (1983) and Smith & Ridout (1995). BBINOMIAL has four associated procedures _BBSET, _BBME, _BBL and _BBGDER that are Genstat implementations of various Fortran subroutines of Smith (1983) and Smith & Ridout (1995). They can also be run independently if desired. _BBSET calculates the integer arrays of counts required by _BBL and _BBGDER. _BBME calculates moment estimates of mu and theta. _BBL calculates the log likelihood given mu and theta.

Action with RESTRICT
If either RBINOMIAL or NBINOMIAL are restricted, the analysis will exclude the restricted units.

References
Kupper, L.L. & Haseman, J.K. (1978). The use of a correlated binomial model for the analysis of toxicological experiments. *Biometrics*, **34**, 69-76.
Smith, D.M. (1983). AS 189. Maximum Likelihood Estimation of the Parameters of the Beta Binomial Distribution. *Applied Statistics*, **32**, 196-204.
Smith, D.M. & Ridout, M.S. (1995). AS R93. A remark on AS 189. Maximum Likelihood Estimation of the Parameters of the Beta Binomial Distribution. *Applied Statistics*, **44**, 545-547.
Williams, D.A. (1975). The analysis of binary responses from toxicological experiments involving reproduction and teratogenicity. *Biometrics*, **31**, 949-952.

See also
Directive: DISTRIBUTION.
Procedure: DPROBABILITY.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

BCDISPLAY

Displays a classification tree (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (summary, details, indented, bracketed, labelleddiagram, numbereddiagram, graph); default * i.e. none

Parameter

TREE = *tree*	Tree to be displayed

Description

BCDISPLAY displays a classification tree, as constructed by the BCLASSIFICATION procedure. The tree can be saved from BCLASSIFICATION (using the TREE option of BCLASSIFICATION), and is specified using the TREE parameter of BCDISPLAY. The type of output is specified by the PRINT option, with settings:

summary	prints a summary of the properties of the tree;
details	gives detailed information about the nodes of the tree;
bracketed	display as used to represent an identification key in "bracketed" form (printed node by node).
indented	display as used to represent an identification key in "indented" form (printed branch by branch);
labelleddiagram	diagrammatic display including the node labels;
numbereddiagram	diagrammatic display with the nodes labelled by their numbers;
graph	plots the tree using high-resolution graphics.

Option: PRINT.
Parameter: TREE.

Method

BCDISPLAY displays the tree using procedures BPRINT and BGRAPH.

See also

Procedures: BCLASSIFICATION, BCIDENTIFY, BCKEEP.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BCFDISPLAY

Displays information about a random classification forest (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (outofbagerror, confusion, importance, orderedimportance); default * i.e. none

Parameter

SAVE = *pointers*	Save structure from BCFOREST providing information about the random forest

Description

BCFDISPLAY displays information about a random classification forest, constructed by the BCFOREST procedure. The SAVE parameter can be set to a pointer, saved using the SAVE option of BCFOREST, containing the necessary information about the forest. Alternatively, if you do not set SAVE, information will be printed about the forest most recently constructed by BCFOREST.

The output is controlled by the PRINT option, with settings:

outofbagerror	out-of-bag error rate,
confusion	confusion matrix,
importance	importance ratings of the X variates and factors, and
orderedimportance	importance ratings of the X variates and factors in decreasing order.

The default is PRINT=* i.e. no printing

Option: PRINT.
Parameter: SAVE.

See also

Procedures: BCFOREST, BCFIDENTIFY.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BCFIDENTIFY

Identifies specimens using a random classification forest (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (identification); default * i.e. none
IDENTIFICATION = *scalar* or *variate*	Saves the identification of each specimen
VOTES = *matrix*	Saves numbers of the terminal nodes reached by the specimens
SAVE = *pointers*	Save structure from BCFOREST providing information about the random forest

Parameters

X = *variates* or *factors*	Explanatory variables
VALUES = *scalars*, *variates* or *texts*	Values to use for the explanatory variables; if these are unset for any variable, its existing values are used

Description

BCFIDENTIFY identifies specimens using a random classification forest, constructed by the BCFOREST procedure. The SAVE parameter can be set to a pointer, saved using the SAVE option of BCFOREST, containing the necessary information about the forest. Alternatively, if you do not set SAVE, the identification will be made using the forest most recently constructed by BCFOREST.

The characteristics of the specimens can be specified in the variates or factors listed by the X parameter. These must have identical names (and levels) to those used originally to construct the tree. You can use the VALUES parameter to supply new values, if those stored in any of the variates or factors are unsuitable.

The PRINT option controls printed output, with settings:

identification	to print the identifications obtained using the tree.

By default nothing is printed.

The IDENTIFICATION option allows you to save the identifications (in a scalar or variate according to whether there is one or several specimens); a missing value is given if there is no clear result (i.e. more than one group possible) for the specimen concerned. The VOTES option can save a specimens-by-groups matrix with the votes given by the forest for each of the groups with each specimen.

Options: PRINT, IDENTIFICATION, VOTES, SAVE.
Parameters: X, VALUES.

Action with RESTRICT

Restrictions are ignored.

See also

Procedures: BCFOREST, BCFDISPLAY.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BCFOREST

Constructs a random classification forest (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (outofbagerror, confusion, importance, orderedimportance, monitoring); default outo, conf, impo
NTREES = *scalar*	Number of trees in the forest; no default – must be specified
NXTRY = *scalar*	Number of X variables to select at random at each node from which to choose the X variable to use there; default is the square root of number of X variables
NUNITSTRY = *scalar*	Number of units of the X variables to select at random to use in the construction of each tree; default is two thirds of the number of units
METHOD = *string token*	Selection criterion to use when constructing the trees (Gini, MPI); default Gini
GROUPS = *factor*	Groupings of the individuals to identify in the trees
NSTOP = *scalar*	Number of individuals in a group at which to stop selecting tests; default 5
ANTIENDCUTFACTOR = *string token*	Adaptive anti-end-cut factor to use (classnumber, reciprocalentropy); default * i.e. none
SEED = *scalar*	Seed for random numbers to select the NXTRY X-variables and NUMITSTRY units; default 0
OWNBSELECT = *string token*	Indicates whether or not your own version of the BSELECT procedure is to be used, as explained in the Method section (yes, no); default no
OUTOFBAGERROR = *scalar*	Saves the "out-of-bag" error rate
CONFUSION = *matrix*	Saves the confusion matrix
SAVE = *pointer*	Saves details of the forest that has been constructed

Parameters

X = *factors* or *variates*	X-variables available for constructing the tree
ORDERED = *string tokens*	Whether factor levels are ordered (yes, no); default no
IMPORTANCE = *scalars*	Saves the importance of each x-variable

Description

The data to construct a random classification forest is a sample of individuals from several groups. The characteristics of the individuals are described in Genstat by a set of factors or variates which are specified by the X parameter of BCFOREST. The GROUPS option of BCFOREST defines the group to which each individual in the sample belongs, and the aim is to be able to identify the groups to which new individuals belong.

A random classification forest is a set of classification trees that are used collectively to identify the group to which an individual specimen belongs (see e.g. Breiman 2001). The identification is obtained by running a new individual through each tree to obtain that tree's "vote" for the group of the individual. The identification is then taken as the group with most votes.

Each classification tree is formed using a random sample of the X variables in the data set, and a bootstrap random sample of their units (i.e. sampled with replacement). The NXTRY option defines how many X variables to select, and the NUNITSTRY option defines how many units to

take. The default for NXTRY is the square root of the number of variables, and the default for NUNITSTRY is two thirds of the number of units. The SEED option specifies a seed for the random numbers that are used to select the variables and to select the units. The default of zero continues an existing sequence of random numbers, if any of the random functions (GRSELECT etc) has already been used in the current Genstat run. Otherwise a seed is chosen at random.

A classification tree progressively splits the individuals into subsets based on their values for the factors or variates. Construction starts at a node known as the *root*, which contains all of the individuals. A factor or variate is chosen to use there that "best" divides the individuals into two subsets. Suppose the available X vectors are all factors with two levels: the first subset will then contain the individuals with level 1 of the factor, and the second will contain those with level 2. Also any individual with a missing value for the factor is put into both groups; so you can use a missing value to denote either variable or unknown observations. Factors may have either ordered or unordered levels, according to whether the corresponding value ORDERED parameter is set to yes or no. For example, a factor called Dose with levels 1, 1.5, 2 and 2.5 would usually be treated as having ordered levels, whereas levels labelled 'Morphine', 'Amidone', 'Phenadoxone' and 'Pethidine' of a factor called Drug would be regarded as unordered. For unordered factors, all possible ways of dividing the levels into two sets are tried. With variates or ordered factors with more than 2 levels, a suitable value *p* is found to partition the individuals into those with values less than or greater than *p*. The tree is then extended to contain two new nodes, one for each of the subsets, and factors or variates are selected for use at each of these nodes to subdivide the subsets further.

The effectiveness of the factor or variate to be chosen for each node depends on how the groups are split between the resulting subsets - the aim is to form subsets that are each composed of individuals from the same group. By default, this is assessed using Gini information (see Breiman *et al.* 1984, Chapter 4) but you can set option METHOD=mpi to use the mean posterior improvement criterion devised by Taylor & Silverman (1993). The ANTIENDCUTFACTOR option allows you to request Taylor & Silverman's adaptive anti-end-cut factors (by default these are not used). The process stops when either no factor or variate provides any additional information, or the subset contains individuals all from the same group, or the subset contains fewer individuals than a limit specified by the NSTOP option (default 1). These nodes where the construction ends are known as *terminal nodes*.

The resulting forest (and its associated information) can be saved using the SAVE option. This can then be used in the BCFDISPLAY procedure to produce further output, or in the BCFIDENTIFY procedure to identify the groups for new values of the x-variables..

The OUTOFBAGERROR option can save the "out-of-bag" error rate. This is calculated using the individuals that were not involved in the construction of each tree. So, it gives an independent measure of the reliability of the forest. The idea is to put each individual through all of the trees where it was not used, and accumulate its votes for each of the groups. The individual is then identified by taking the group where it had most votes, and the error rate is calculated by comparing the identifications of the individuals with their true group (as defined by the GROUPS factor).

The CONFUSION option can save the confusion matrix. This is a groups-by-groups matrix that can be calculated at the same time as the out-of-bag error. The rows represent the true groups, and the columns represent the out-of-bag identifications obtained using the forest. The diagonal of the matrix records the number of individuals correctly identified in each group, while the off-diagonal elements show the numbers that have been identified incorrectly (i.e. that have been "confused" with other groups).

The IMPORTANCE parameter can save a variate giving the "importance" of each X variate or factor in the forest. This is calculated by accumulating the sum of the values of the selection function (see METHOD) over the times when the X variable is used in the forest.

Printed output is controlled by the PRINT option, with settings:

`outofbagerror`	out-of-bag error rate,
`confusion`	confusion matrix,
`importance`	importance ratings of the `X` variates and factors,
`orderedimportance`	importance ratings of the `X` variates and factors in decreasing order, and
`monitoring`	monitoring information during the construction process.

The default is `PRINT=outofbagerror,confusion,importance`.

Options: `PRINT`, `NTREES`, `NXTRY`, `NUNITSTRY`, `METHOD`, `GROUPS`, `NSTOP`, `ANTIENDCUTFACTOR`, `SEED`, `OWNBSELECT`, `OUTOFBAGERROR`, `CONFUSION`, `SAVE`.
Parameters: `X`, `ORDERED`, `IMPORTANCE`.

Method

`BCFOREST` calls procedure `BCONSTRUCT` to form the tree. This uses a special-purpose procedure `BSELECT`, which is customized specifically to select splits for use in classification trees. You can use your own method of selection by providing your own `BSELECT` and setting option `OWNBSELECT=yes`. In the standard version of `BSELECT`, the `BASSESS` directive is used to assess the potential splits.

Action with `RESTRICT`

Restrictions on the `X` vectors or `GROUPS` factor are ignored.

References

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth, Monterey.

Breiman, L. (2001) Random forests. *Machine Learning*, **45**m, 5-32.

Taylor, P.C. & Silverman, B.W. (1993). Block diagrams and splitting criteria for classification trees. *Statistics & Computing*, **3**, 147-161.

See also

Procedures: `BCFDISPLAY`, `BCFIDENTIFY`, `BCLASSIFICATION`, `BKEY`, `BREGRESSION`, `KNEARESTNEIGHBOURS`.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BCIDENTIFY

Identifies specimens using a classification tree (R.W. Payne).

Options

`PRINT` = *string tokens*	Controls printed output (`identification`, `transcript`); if `PRINT` is unset in an interactive run `BCIDENTIFY` will ask what you want to print, in a batch run the default is `iden`
`TREE` = *tree*	Specifies the tree
`IDENTIFICATION` = *text*	Saves the identification of each specimen
`TERMINALNODES` = *pointer*	Saves the numbers of the terminal nodes reached by each specimen
`PROBABILITIES` = *matrix*	Specimen × group matrix giving the probability that the specimens belong to each group
`MVINCLUDE` = *string token*	Whether to provide identifications for specimens with missing or unavailable values of the x-variables (`explanatory`); default `expl`

Parameters

`X` = *variates* or *factors*	Explanatory variables
`VALUES` = *scalars*, *variates* or *texts*	Values to use for the explanatory variables; if these are unset for any variable, its existing values are used

Description

`BCIDENTIFY` identifies specimens using a classification tree, as constructed by the `BCLASSIFICATION` procedure. The tree can be saved from `BCLASSIFICATION` (using the `TREE` option of `BCLASSIFICATION`), and specified for `BCIDENTIFY` using its own `TREE` option. Alternatively, `BCIDENTIFY` will ask you for the identifier of the tree if you do not specify `TREE` when running interactively.

The characteristics of the specimens can be specified in the variates or factors listed by the `X` parameter. These must have identical names (and levels) to those used originally to construct the tree. You can use the `VALUES` parameter to supply new values, if those stored in any of the variates or factors are unsuitable.

If you do not set `X` when running interactively, `BCIDENTIFY` will ask you to supply the relevant characteristics in turn, as required by the tree. Otherwise, if an x-variable in the tree is not specified in the `X` parameter list, its values are assumed to be unavailable (i.e. missing).

By default, when the x-variable required at a node in the tree is unavailable or contains a missing value, `BCIDENTIFY` will follow all the branches from that node, and form a combined conclusion. You can set option `MVINCLUDE=*`, if you would prefer the identification to be missing.

The `PRINT` option controls printed output, with settings:

`identification`	prints the identifications obtained using the tree;
`transcript`	prints the observed characteristics when supplied in response to questions in an interactive run.

If you do not set `PRINT` in an interactive run, `BCIDENTIFY` will ask what you would like to print. In batch, the default is to print the identifications.

The `IDENTIFICATION` option allows you to save the identifications (in a text). The `TERMINALNODES` option allows you to save a pointer, with an element for each specimen, containing the numbers of the terminal nodes reached in the tree to provide its identification. This will be a scalar if the identification was derived from a single node, or a variate if it

involved more than one (because several branches have been taken, as the result of a missing x-value). Finally, the PROBABILITIES option can save a specimen-by-group matrix giving the probability that the specimens belong to each group.

Options: PRINT, TREE, IDENTIFICATION, TERMINALNODES, PROBABILITIES, MVINCLUDE.
Parameters: X, VALUES.

Method
BCIDENTIFY uses BIDENTIFY to find the terminal nodes of the tree that correspond to the values of the explanatory variables.

Action with RESTRICT
Restrictions are ignored.

See also
Procedures: BCLASSIFICATION, BCDISPLAY, BCKEEP.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BCKEEP

Saves information from a classification tree (R.W. Payne).

No options

Parameters

TREE = *trees*	Tree from which the information is to be saved
SUMMARY = *variates*	Saves summary information about each tree
XVARIABLES = *pointers*	Saves the identifiers of the x-variables in each tree

Description

BCKEEP saves information from a classification tree, constructed by the BCLASSIFICATION procedure. The tree can be saved using the TREE option of BCLASSIFICATION, and is specified for BCKEEP using its TREE parameter.

The SUMMARY parameter saves a variate containing summary information. The first element contains the number of nodes, the second contains the number of terminal nodes, and the third contains the misclassification rate.

The XVARIABLES parameter saves a pointer containing the identifiers of the x-variables in the tree.

Options: none.
Parameters: TREE, SUMMARY, XVARIABLES.

See also

Procedures: BCLASSIFICATION, BCDISPLAY, BCIDENTIFY.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BCLASSIFICATION

Constructs a classification tree (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (summary, details, indented, bracketed, labelleddiagram, numbereddiagram, graph, monitoring); default * i.e. none
METHOD = *string token*	Selection criterion to use when constructing the tree (Gini, MPI); default Gini
GROUPS = *factor*	Groupings of the individuals in the tree
TREE = *tree*	Saves the tree that has been constructed
NSTOP = *scalar*	Number of individuals in a group at which to stop selecting tests; default 5
ANTIENDCUTFACTOR = *string token*	Adaptive anti-end-cut factor to use (classnumber, reciprocalentropy); default * i.e. none
OWNBSELECT = *string token*	Indicates whether or not your own version of the BSELECT procedure is to be used, as explained in the Method section (yes, no); default no

Parameters

X = *factors* or *variates*	X-variables available for constructing the tree
ORDERED = *string tokens*	Whether factor levels are ordered (yes, no); default no

Description

The starting point for a classification tree is a sample of individuals from several groups. The characteristics of the individuals are described in Genstat by a set of factors or variates which are specified by the X parameter of BCLASSIFICATION. The GROUPS option of BCLASSIFICATION defines the group to which each individual in the sample belongs, and the aim is to be able to identify the groups to which new individuals belong.

The tree progressively splits the individuals into subsets based on their values for the factors or variates. Construction starts at a node known as the *root*, which contains all of the individuals. A factor or variate is chosen to use there that "best" divides the individuals into two subsets. Suppose the X vectors are all factors with two levels: the first subset will then contain the individuals with level 1 of the factor, and the second will contain those with level 2. Also any individual with a missing value for the factor is put into both groups; so you can use a missing value to denote either variable or unknown observations. Factors may have either ordered or unordered levels, according to whether the corresponding value ORDERED parameter is set to yes or no. For example, a factor called Dose with levels 1, 1.5, 2 and 2.5 would usually be treated as having ordered levels, whereas levels labelled 'Morphine', 'Amidone', 'Phenadoxone' and 'Pethidine' of a factor called Drug would be regarded as unordered. For unordered factors, all possible ways of dividing the levels into two sets are tried. With variates or ordered factors with more than 2 levels, a suitable value *p* is found to partition the individuals into those with values less than or greater than *p*. The tree is then extended to contain two new nodes, one for each of the subsets, and factors or variates are selected for use at each of these nodes to subdivide the subsets further.

The effectiveness of the factor or variate to be chosen for each node depends on how the groups are split between the resulting subsets - the aim is to form subsets that are each composed of individuals from the same group. By default, this is assessed using Gini information (see Breiman *et al.* 1984, Chapter 4) but you can set option METHOD=mpi to use the mean posterior

improvement criterion devised by Taylor & Silverman (1993). The ANTIENDCUTFACTOR option allows you to request Taylor & Silverman's adaptive anti-end-cut factors (by default these are not used). The process stops when either no factor or variate provides any additional information, or the subset contains individuals all from the same group, or the subset contains fewer individuals than a limit specified by the NSTOP option (default 5). These nodes where the construction ends are known as *terminal nodes*.

The resulting tree can be saved using the TREE option. Details of the tree can be printed as selected by the PRINT option, with settings:

summary	prints a summary of the properties of the tree;
details	gives detailed information about the nodes of the tree;
bracketed	display as used to represent an identification key in "bracketed" form (printed node by node).
indented	display as used to represent an identification key in "indented" form (printed branch by branch);
labelleddiagram	diagrammatic display including the node labels;
numbereddiagram	diagrammatic display with the nodes labelled by their numbers;
graph	plots the tree using high-resolution graphics.
monitoring	prints information monitoring the construction process.

BCLASSIFICATION stores the information required for printing as part of the tree. If the X vectors are all factors with 2 levels, the labels for the labelled diagram are formed as "*identifier*==n_1", where n_1 is the first level of the factor. The lines of the indented and bracketed forms are formed similarly if the factor has no extra test and no labels. Otherwise, the form is "*xname lname*", where *xname* is the extra text if this has been defined (by the EXTRA parameter of the FACTOR command) or else the identifier of the factor, and *lname* is the label if available or the level if not. If the X vectors include variates or ordered factors with more than two levels and there is no extra text, the labels are formed as "*identifier*<*p*" and "*identifier*>*p*", where *p* is the value chosen to partition the data for the variate concerned. If there is an extra text for a particular factor or variate, the labels are "*xname* < *p*" and "*xname* > *p*". The style is similar for unordered factors, but here the labels involve the operators .IN. and .NI. instead of < and >.

Generally the construction will result in *over-fitting*, that is it will form a tree that keeps selecting factors or variates to subdivide the individuals beyond the point that can be justified statistically. The solution is to prune the tree to remove the uninformative sub-branches, and this can be performed using the BPRUNE procedure. It is best, if possible, to base the pruning on an independent set of data. The pruning uses *accuracy* figures, which are stored for each node of the tree. The tree also stores a *prediction* for each node, which corresponds to the group with most individuals at the node. For each node of a classification tree, the accuracy is the number of misclassified individuals at the node, divided by the total number of individuals in the data set. It thus measures the impurity of the subset at that node (how far it is from it from being homogeneous i.e. having individuals all from a single group). The BCVALUES procedure can be used to calculate new accuracy and prediction values, from another data set.

Finally, once the tree has been pruned, the group of a new individual can be identified by supplying their values for the X factors or variates to the BCIDENTIFY procedure. This runs the individual through the tree to see which terminal node it would reach. The group can then be identified using the prediction value stored for that node.

Options: PRINT, METHOD, GROUPS, TREE, NSTOP, ANTIENDCUTFACTOR, OWNBSELECT.
Parameters: X, ORDERED.

Method

BCLASSIFICATION calls procedure BCONSTRUCT to form the tree. This uses a special-purpose procedure BSELECT, which is customized specifically to select splits for use in classification trees. You can use your own method of selection by providing your own BSELECT and setting option OWNBSELECT=yes. In the standard version of BSELECT, the BASSESS directive is used to assess the potential splits.

Action with RESTRICT

Restrictions on the X vectors or GROUPS factor are ignored.

References

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth, Monterey.

Taylor, P.C. & Silverman, B.W. (1993). Block diagrams and splitting criteria for classification trees. *Statistics & Computing*, **3**, 147-161.

See also

Procedures: BCDISPLAY, BCIDENTIFY, BCKEEP, BCVALUES, BGRAPH, BPRUNE, BKEY, BCFOREST, BREGRESSION, KNEARESTNEIGHBOURS.

Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BCONSTRUCT

Constructs a tree (R.W. Payne).

Option

PRINT = *string token*	Whether to print monitoring information (monitoring); default * i.e. none

Parameters

TREE = *trees*	Saves the trees that have been constructed
DATA = *identifiers*	Data available for constructing the trees

Description

BCONSTRUCT is a utility procedure that is used by the tree procedures like BCLASSIFICATION, BKEY and BREGRESSION to construct trees. It calls a procedure BSELECT to determine the test to be performed at each node of the tree. Customized versions of this procedure are available for each type of tree, and are accessed automatically along with the top-level procedure for the type of tree concerned. BCONSTRUCT is thus completely general – and can be used for other types of tree simply by providing an appropriate version of BSELECT.

The DATA parameter of BCONSTRUCT supplies a pointer containing the information required to construct the tree. (This is then passed through to BSELECT, together with information about the node for which a test is to be selected.) The TREE parameter saves the tree that has been constructed, and the PRINT option can be set to monitoring to produce monitoring information during construction.

Option: PRINT.
Parameters: TREE, DATA.

Method

BCONSTRUCT calls a procedure BSELECT to decide which test to use at each node of the tree. This must be customized according to the type of tree that is required. BSELECT has no options. Its parameters are as follows.

DATA = *pointer*	Data for constructing the tree (as provided by the DATA parameter of BCONSTRUCT)
TESTS = *pointer*	Tests already made between the root and the current node
BRANCHES = *variate*	Branches taken at each previous node
LABEL = *text*	Returns a label to put onto the node
NEWTEST = *scalar*	or expression New test to be done at the node (expression), or identification made at the node (scalar) if no new test selected
NBRANCH = *scalar*	Returns the number of branches to insert below the node
ADDITIONAL = *pointer*	Other information to store at the node
LADDITIONAL = *text*	Labels for the other information

After BSELECT has selected a test, the tree is extended by the BGROW directive, function BTERMINAL is used to find the next terminal node, and functions BPATH and BBRANCHES are used to ascertain the nodes and branches between that node and the root.

Action with RESTRICT

The use of any restrictions will depend on the BSELECT procedure, called by BCONSTRUCT.

See also
Directives: BASSESS, TREE.
Procedures: BCLASSIFICATION, BKEY, BREGRESSION.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

BCVALUES

Forms values for nodes of a classification tree (R.W. Payne).

Options

GROUPS = *factor*	Groupings of the observations in the data set
TREE = *tree*	Tree for which predictions and accuracy values are to be formed
REPLACE = *string token*	Whether to replace the values stored in the tree (yes, no); default no
PREDICTION = *pointer*	New predictions for the nodes of the tree
ACCURACY = *pointer*	New accuracy values for the nodes of the tree
REPLICATION = *pointer*	New replication tables for the nodes of the tree

Parameter

X = *factors* or *variates*	Values of the factors or variates used in the tree for the new data set

Description

When pruning a classification tree, it is best to use "accuracy" figures that are derived from a different set or sets of data from that which was used to construct the tree. BCVALUES allows these to be calculated, together with new predictions for the nodes of the tree.

The TREE option specifies the tree for which the values are to be formed. The GROUPS option specifies a factor defining the groupings of the observations in the new data set, and the X parameter defines their levels for the factors or variates as used to construct the tree. You can set option REPLACE=yes to use the new values to replace those already stored in the tree. Alternatively, you can use the PREDICTION parameter to save the predictions, in a pointer. This has an element for each node of the tree (and with the same suffix as that node) pointing to a scalar storing the prediction for the node. Similarly, the ACCURACY parameter saves the accuracies, in a pointer to a set of scalars, and the REPLICATION parameter saves the replications of the groups at each node, in a pointer to a set of tables classified by the GROUPS factor. You can use these later to replace the prediction and accuracy values in the original tree by

```
CALCULATE Tree[]['accuracy'] = ACCURACY[]
&         Tree[]['prediction'] = PREDICTION[]
&         Tree[]['replication'] = REPLICATION[]
```

Alternatively, you may want to combine them first with other estimates, for example to form bootstrapped estimates.

Options: GROUPS, TREE, REPLACE, PREDICTION, ACCURACY, REPLICATION.
Parameter: X.

Method

BCVALUES uses the standard Genstat tree functions to obtain the necessary information about the tree.

Action with RESTRICT

BCVALUES takes account of any restrictions on the X vectors or on GROUPS.

See also
Procedures: BCLASSIFICATION, BPRUNE.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BGIMPORT

Imports MCMC output in CODA format produced by WinBUGS or OpenBUGS (D.A. Murray).

Options

INDEXFILE = *text*	Name of file containing index for output files
OUTPREFIX = *text*	Prefix name for the output files
WORKDIRECTORY = *text*	Working directory to use; default current Genstat working directory
PNAMES = *text*	Saves the names of the simulated nodes
NOUTFILES = *scalar*	Number of output files or chains to read; default 1

Parameter

SIMULATIONS = *pointers*	Saves the simulations in a list of pointers, one for each Markov chain

Description

BGIMPORT imports Markov Chain Monte Carlo (MCMC) output in CODA format produced by WinBUGS or OpenBUGS. Data imported using BGIMPORT can be used in the procedure BGPLOT. The monitored values from a MCMC can be saved in text format (CODA files) from WinBUGS or by using BGXGENSTAT. The CODA files consist of an output file for each chain showing the iteration number and value. In addition there is an index file containing a description of which lines of the output files correspond to which variable.

The name of the file containing the index, showing which rows of the output files correspond to which variables, is supplied using the INDEXFILE option. The string for the prefix for the output files is supplied using the OUTPREFIX option. With WinBUGS, the file names have the form <prefix>index.txt, <prefix>1.txt, <prefix>2.txt, etc; here you need to set OUTPREFIX='<prefix>'. Similarly, OpenBUGS produces files with names using the form <prefix>CODAindex.txt, <prefix>CODAchain1.txt, and so on; here you need to set OUTPREFIX='<prefix>CODAchain'. By default, the working directory will be the current directory, but you can supply an alternative directory using the WORKDIRECTORY parameter. The names of the monitored nodes can be saved using the PNAMES option.

The data are saved using the SIMULATIONS parameter as a list of pointers where each pointer contains the simulations and associated information for each Markov chain. These can be input to BGPLOT, for plotting, using its own SIMULATIONS parameter.

Options: INDEXFILE, OUTPREFIX, WORKDIRECTORY, PNAMES, NOUTFILES.
Parameter: SIMULATIONS.

See also

Procedures: BGPLOT, BGXGENSTAT.
Genstat Reference Manual 1 Summary section on: Bayesian methods.

BGPLOT

Produces plots for output and diagnostics from MCMC simulations (D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (summary); default *
PLOT = *string tokens*	Controls the type of plot (trace, density, autocorrelation, gelmanrubin); default trac
ARRANGEMENT = *string tokens*	Specifies whether to draw the plots individually or 4 to a page (single, multiple); default sing
START = *scalar*	Start iteration number for plots
END = *scalar*	End iteration number for plots
MAXLAG = *scalar*	Maximum lag for autocorrelation plots; default 50
BANDWIDTH = *scalar*	The bandwidth value to be used for the density plots.
GRMETHOD = *scalar*	Controls the method of the Gelman-Rubin diagnostic plot (gr, bgr); default bgr
BINWIDTH = *scalar*	Number of values in each bin in the Gelman-Rubin plot; default 50
USEALLSAMPLES = *text*	Whether to use all the samples for Gelman-Rubin plot, or to discard the first half of the observations (yes, no); default no

Parameter

SIMULATIONS = *pointers*	List of pointers containing simulations, one for each Markov chain

Description

BGPLOT produces plots for output and diagnostics from Markov Chain Monte Carlo (MCMC) simulations. The procedure can be used after running a Bayesian MCMC analysis using BGXGENSTAT. Alternatively, data from CODA files produced by WinBUGS that are imported using BGIMPORT can be displayed using BGPLOT.

The data are supplied using the SIMULATIONS parameter as list of pointers, where each pointer contains the simulations and associated information for a Markov chain. The data for BGPLOT can be taken directly from the BGXGENSTAT or BGIMPORT procedures. The SIMULATIONS parameter corresponds to the SIMULATIONS parameters of BGXGENSTAT and BGIMPORT.

BGPLOT produces four types of plot which can be selected using the PLOT option. The ARRANGEMENT option controls whether the plots are each drawn on separate pages or four to a page in a two-by-two arrangement.

The trace setting of PLOT produces a trace plot for each monitored node, where every chain for the monitored nodes is superimposed on the same plot.

The density setting produces a kernel density smooth for each monitored node. The bandwidth for the density plot can be supplied using the BANDWIDTH option.

The autocorrelation setting generates an autocorrelation plot for each monitored node, where every chain for the monitored nodes is superimposed on the same plot. You can specify the maximum lag for which the autocorrelation is calculated using the MAXLAG option (default 50).

The gelmanrubin setting produces the Gelman-Rubin "Potential Scale Reduction Factor" (PSRF) diagnostic. The PSRF compares the between and within variances of multiple chains. BGPLOT produces a plot of the PSRF, and a second plot of the between and within variances. Convergence is assumed to occur when the PSRF is close to one, and the between and within variances stabilise around the same value. You can display the convergence diagnostic by setting

option PRINT=summary. The method used for the diagnostic can be controlled using the GRMETHOD option. The gr setting uses the original Gelman-Rubin PSRF diagnostic, while the bgr setting uses the Gelman-Rubin-Brooks version, which interprets the diagnostic as a ratio of interval lengths rather than a variance ratio. The BINWIDTH option specifies the number of values in each bin of the Gelman-Rubin plot (default 50). Usually the first half of the observations is discarded in the Gelman-Rubin plot, but you can set option USEALLSAMPLES=yes to use them all.

Options: PRINT, PLOT, ARRANGEMENT, START, END, MAXLAG, BANDWIDTH, GRMETHOD, BINWIDTH, USEALLSAMPLES.
Parameter: SIMULATIONS.

Method

The autocorrelations are calculated using the CORRELATE directive and the densities are evaluated using the KERNELDENSITY procedure. Gelman & Rubin's diagnostic (1992) monitors convergence of MCMC output from parallel chains which each have different starting points that are overdispersed with respect to the target distribution. The details for the calculation of the potential scalar reduction factor (R) is described in Gelman & Rubin (1992). The Gelman-Rubin-Brooks diagnostic is an alternative method to the R statistic devised by Gelman & Rubin where R is calculated by the ratio of the central 80% width of the pooled intervals by the average central 80% width of the within intervals, see Brooks & Gelman (1998).

Action with RESTRICT

Any data restrictions will be ignored.

References

Brooks, S.P. & Gelman, A. (1998). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, **7**, 434-455.

Gelman, A. & Rubin, D. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, **7**, 457-511.

See also

Procedures: BGIMPORT, BGXGENSTAT.
Genstat Reference Manual 1 Summary section on: Bayesian methods.

BGRAPH

Plots a tree (R.W. Payne).

Option

SIZE = *scalar*	Provides a multiplier by which to scale the node labels

Parameters

TREE = *trees*	Trees to be plotted
XTERMINAL = *scalars* or *variates*	X-spacing (scalar) or x-values (variate) for the terminal nodes of each tree; default 2

Description

BGRAPH plots a tree. The tree to be plotted is specified by the TREE parameter. BGRAPH arranges the nodes with the root at the top and the terminal nodes at the bottom of the plot. The terminal nodes are arranged automatically across the screen, but the x-coordinates can be specified explicitly using the XTERMINAL parameter. The SIZE option allows the size of the node labels to be adjusted by a scaling factor (default 1).

Option: SIZE.
Parameters: TREE, XTERMINAL.

Method

BGRAPH uses the standard tree functions to parse the tree and thus decide where to plot each node.

See also

Directive: TREE.
Procedures: BPRINT, BRDISPLAY, BKDISPLAY, BRDISPLAY.
Genstat Reference Manual 1 Summary section on: Graphics.

BGXGENSTAT

Runs WinBUGS or OpenBUGS from Genstat in batch mode using scripts (D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (bugslog, nodestatistics, dic); default node
WPATH = *text*	Path specifying the location of the WinBUGS executable
WEXE = *text*	Name of the WinBUGS or OpenBUGS executable to run; default 'WinBUGS14.exe'
MODELFILE = *text*	Name of a file containing the model in WinBUGS code; the file should have an extension of .txt
DATA = *pointer*	A pointer to the data used by the WinBUGS model
IDATANAMES = *text*	A text containing the names for the data
MONITOR = *text*	The names of the variables that are to monitored
NCHAINS = *scalar*	Number of Markov chains; default 3
NBURNIN = *scalar*	Length of burn-in per chain; default 1000
NSAMPLES = *scalar*	Number of samples to run after burn-in; default 5000
THIN = *scalar*	Thinning rate where the samples from every kth iteration are stored; default 1
INAMES = *text*	The names for the initial parameters
DIC = *string token*	Whether to calculate the deviance information criterion (yes, no); default no
SEED = *scalar*	Specifies a seed to use for the random number generator in BUGS; default uses a pseudo-random number generated from the uniform distribution
WORKDIRECTORY = *texts*	Working directory to use; default current Genstat working directory
BUGS = *string token*	Whether to use WinBUGS or OpenBUGS (winbugs, openbugs); default winb
VIEWBUGS = *string token*	Whether to leave WinBUGS open after the run (yes, no); default no
CONTINUE = *string token*	Whether to continue Genstat server without waiting for WinBUGS to complete; (yes, no); default no
CODA = *string token*	Whether to save CODA files (yes, no); default no
WLOG = *text*	Name of file to save log from WinBUGS or OpenBUGS

Parameters

INITIAL = *pointers*	List of pointers, one for each set of initial values for each Markov chain
SIMULATIONS = *pointers*	List of pointers to save simulations, one for each Markov chain

Description

WinBUGS (Bayesian inference Using Gibbs Sampling, Spiegelhalter, Thomas, Best & Lund 2003) is an application that can be used for the Bayesian analysis of complex models using Markov Chain Monte Carlo (MCMC) methods. WinBUGS is available free at

```
http://www.mrc-bsu.cam.ac.uk/bugs/
```

and an open-source version of the core BUGS code (OpenBUGS) is also available at

```
http://mathstat.helsinki.fi/openbugs/
```

BGXGENSTAT can be used to run WinBUGS or OpenBUGS from Genstat in batch mode using scripts. To execute commands within WinBUGS, BGXGENSTAT automatically creates a data and

script file containing the necessary commands for the MCMC. These files are passed to WinBUGS and, once it has completed execution, the results can be imported into Genstat.

To use BGXGENSTAT either WinBUGS or OpenBUGS must be installed on the current system. The control of whether WinBUGS or OpenBUGS is to be run is set using the BUGS option. The executable used when submitting the script is specified using the WEXE option, by default this uses WinBUGS14.exe for WinBUGS and Winbugs.exe for OpenBUGS. The location of the BUGS executable used to run the script should be specified using the WPATH option. The directory for the path should be specified as a text containing the absolute pathname, for example in Windows the default directory for the executable for WinBUGS 1.4 would be

```
C:/Program Files/WinBUGS14
```

WinBUGS requires the script file to be written to the directory containing the executable. Therefore, you must have write permission to this directory to be able to run BGXGENSTAT.

The model to run in WinBUGS should be supplied using WinBUGS code, within a file supplied using the MODELFILE option. The data are supplied in a pointer using the DATA option. Each element of the pointer should represent a different data structure used in the model, and can be a scalar or a variate or, for 2-D data, a matrix or pointer to variates. The names for the data structures are supplied using the IDATANAMES option, and should be in the same order as that in which the data occur in the DATA pointer. The names of the variables of interest that are to be monitored are supplied within a text using the MONITOR option. By default three Markov chains are run, but this can be changed by the NCHAINS option. The number of burn-in iterations and the number of samples to run after the burn-in are specified using the NBURNIN and NSAMPLES options respectively. A thinning rate can be specified using the THIN option, where the samples from every *k*th iteration are stored. The initial values for the chains are supplied in a list pointers using the INITIAL parameter. The elements of each pointer can be either a scalar or variate, and must be in the same order in all the pointers. The names of the variables for the initial values are supplied using the INAMES option where the names are in the same order as the data within the pointers for the INITIAL parameter. Each run of WinBUGS produces CODA output files containing the Markov Chain Monte Carlo output in CODA format. The data from the CODA files can be imported into Genstat using the SIMULATIONS parameter. By default the CODA files are deleted, but you can set CODA=yes to save these in the working directory. The CODA files consist of an output file for each chain, showing the iteration number and value. In addition there is an index file containing a description of which lines of the output files correspond to which variable. BGXGENSTAT saves the index file using the name WBGCODAIndex.txt and the output files are saved as WBGCODA1.txt, WBGCODA2.txt, etc.

When WinBUGS is run, the Genstat server is suspended until the WinBUGS script has run. If you do not want to wait until the WinBUGS script has run before continuing with Genstat, you can set the option CONTINUE=yes. However, when this option is set to yes, the CODA and PRINT options, and SIMULATION parameter will be ignored. If errors occur within the WinBUGS script, the VIEWBUGS option can be used to keep WinBUGS open after the run.

Running WinBUGS produces a log file which can be saved using the WLOG option. By default, the working directory will be the current directory. However, an alternative directory can be supplied using the WORKDIRECTORY parameter.

The PRINT option controls printed output, with the settings:

bugslog	to print the contents of the log file,
nodestatistics	to display summary statistics for the nodes, and
dic	to display the deviance information criterion (the DIC option must then be set).

Options: PRINT, WPATH, WEXE, MODELFILE, DATA, IDATANAMES, MONITOR, NCHAINS, NBURNIN, NSAMPLES, THIN, INAMES, DIC, SEED, WORKDIRECTORY, BUGS, VIEWBUGS, CONTINUE, CODA, WLOG.

Parameters: INITITAL, SIMULATIONS.

Method

To execute commands within WinBUGS, BGXGENSTAT automatically creates a data and script file containing the necessary commands. In Windows the commands are submitted to WinBUGS by creating a bat file containing a command line and then executes this within a windows command processor. Once WinBUGS has completed Genstat uses the log file and CODA files generated by WinBUGS to retrieve the results. The node statistics are read directly from the log file, and the SIMULATIONS are imported from the coda files.

Action with RESTRICT

Any data restrictions will be ignored.

References

Gelman, A., Carlin J., Stern H. & Rubin, D. (2003). *Bayesian Data Analysis*. CRC Press, London.

Spiegelhalter, D.J., Thomas, A., Best, N.G. & Lund, D. (2003). *WinBUGS Version 1.4 Users Manual*. MRC Biostatistics Unit, Cambridge.
URL http://www.mrc-bsu.cam.ac.uk/bugs/

Spiegelhalter, D.J., Thomas, A., Best, N.G. & Lund, D. (2004). *WinBUGS Version 2.0 Users Manual*. MRC Biostatistics Unit, Cambridge.
URL http://www.mrc-bsu.cam.ac.uk/openbugs/

See also

Procedures: BGIMPORT, BGPLOT, DEMC.

Genstat Reference Manual 1 Summary section on: Bayesian methods.

BIPLOT

Produces a biplot from a set of variates (S.A. Harding).

Options

PRINT = *string tokens*	Printed output from the analysis (singular, scores); default * i.e. no output
GRAPHICS = *string token*	What sort of graphics to use (lineprinter, highresolution); default high
WINDOW = *scalar*	Window number for the graph; default 3
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
METHOD = *string token*	Type of analysis required (principalcomponent, variate, diagnostic); default prin
STANDARDIZE = *string tokens*	Whether to centre the configurations (at the origin), and/or to normalize them (to unit sum of squares) prior to analysis (centre, normalize); default cent, norm
LABELS = *factor* or *text*	Labels to identify the points for the individuals
VLABELS = *factor* or *text*	Labels to identify the points for the variates
NDIMENSIONS = *scalar*	Number of dimensions to save with COORDINATES and VCOORDINATES; default 2

Parameters

DATA = *pointers*	Each pointer contains a set of variates to be analysed
COORDINATES = *matrices*	Used to store the scores for the individuals
VCOORDINATES = *matrices*	Used to store the scores for the variates

Description

BIPLOT produces a graphical representation of the relationships between data units and variates, as described by Gabriel (1971).

The data for the procedure consist of a set of variates, contained in a pointer specified by the DATA parameter. The data may be centred at the origin and/or normalized before plotting, by setting the STANDARDIZE option. The variates must not contain any missing values, nor should they be restricted. The values of the variates remain unaltered on exit from the procedure. The METHOD option allows the user to select which form of the biplot is to be plotted: principal component, variate, or diagnostic biplot.

Printed output is controlled by the option PRINT with settings: singular to print the singular values, and scores to print the scores. By default, nothing is printed.

The GRAPHICS option controls whether the biplot is plotted in high-resolution or line-printer styles; or setting GRAPHICS=* suppresses the plot. The WINDOW option specifies the window to use for a high-resolution plot (default 3), and the SCREEN option controls whether or not to clear the screen first (default clear).

Results from the analysis can be saved using the parameters COORDINATES and VCOORDINATES. The structures specified for these parameters need not be declared in advance. The number of dimensions that are saved is specified by the NDIMENSIONS option; default 2.

Options: PRINT, GRAPHICS, WINDOW, SCREEN, METHOD, STANDARDIZE, LABELS, VLABELS, NDIMENSIONS.

Parameters: DATA, COORDINATES, VCOORDINATES.

Method

The biplot (Gabriel, 1971) is a graphical representation of the relationships between n individuals and between p variates. If these variates are arranged as a matrix X ($n \times p$), the singular value decomposition of X ($X = U S V'$) is used to express the least-squares approximation to X in two dimensions in the form $X2 = A B'$, where $X2$ is ($n \times 2$); A ($n \times 2$) and B ($p \times 2$) are given by the first two columns of ($U S^r$) and ($V S^{(1-r)}$) respectively. The matrices A and B give the coordinates of the row and column markers, and the constant r can be set to either 0, 0.5, or 1 to obtain the form of the biplot requested by the METHOD option.

Action with RESTRICT

The variates must not be restricted.

References

Gabriel, K.R. (1971). The biplot graphic display of matrices with application to pricipal component analysis. *Biometrika*, **58**, 453.

Gower, J.C. and Digby, P.G.N. (1981). Expressing Complex Relationships in Two Dimensions. In: *Interpreting Multivariate Data* (ed. V. Barnett). Wiley, New York.

See also

Procedures: DBIPLOT, CABIPLOT, CRBIPLOT, CRTRIPLOT.

Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

BJESTIMATE

Fits an ARIMA model, with forecast and residual checks (G. Tunnicliffe Wilson & S.J. Welham).

Options

PRINT = *string tokens*	Controls printed output (description, monitoring, model); default desc, moni, mode
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
WINDOWS = *scalar* or *variate*	Windows to be used for residual plots: a scalar N indicates that plots are to be produced on separate pages in window N (as currently defined), whereas a variate specifies four separate windows to be redefined (within the procedure) for plotting four graphs on one page; default 1
PENS = *variate*	The three pens to be used (after being defined appropriately) for drawing the plots; default !(1,2,3)

Parameters

SERIES = *variates*	Holds the time series to which the model is to be fitted
LENGTH = *scalars* or *variates*	Specifies the units to be used from each series: a scalar N indicates that the first N units of the series are to be used, a variate of length 2 gives the index of the first and last units of the subseries to be used; by default the whole series is used
ORDERS = *variates*	Variate holding the orders for the ARIMA model to be fitted to each series
PARAMETERS = *variates*	Variate specifying the initial values for the parameters (to be used by the TFIT directive)
TSM = *TSMs*	TSM to store each fitted model, also to supply values for orders and parameters if ORDERS and PARAMETERS are unset
RESIDUALS = *variates*	Variate to save the residuals from fitting the model to each series

Description

BJESTIMATE fits an ARIMA model of specified orders to a time series given by the SERIES parameter. If only part of the series is to be used, this should be specified by the parameter LENGTH, using either a scalar N to indicate that the first N values should be used, or a variate of length 2 holding the positions of the first and last units of the subseries to be included. If only a subseries is used in the estimation, forecasts of any later series values are plotted to act as a check on the fitted model. The fit of the model is examined using the procedure BJIDENTIFY on the residual series; this residual series is plotted, together with its sample autocorrelations, partial autocorrelations and periodogram. The residuals from the fitted model can be saved using the RESIDUALS parameter.

The orders of the ARIMA model can be specified by the ORDERS parameter; alternatively, if parameter TSM has been set to the identifier of a TSM structure to save the results, ORDERS can be omitted and the orders will be taken from those of the TSM. Likewise, the PARAMETERS parameter can be set to a variate of initial values for the TFIT directive, used by the procedure to fit the model; if PARAMETERS is unset these will again be taken from the setting of the TSM parameter, if available. Any unset initial values are determined automatically by TFIT.

Printed output is controlled by the option PRINT; by default, a description of the series, monitoring of the estimation process and the fitted model are printed.

Graphical output is controlled by the options GRAPHICS, WINDOWS and PENS. Option GRAPHICS controls whether plots are produced for line-printer output or on the current high-resolution graphics device; by default high-resolution plots are given. Option WINDOWS controls the way in which the high-resolution plots are arranged. First of all there may be a graph of forecasts; this is plotted on a new page (i.e. a cleared screen), using the first window specified. Then procedure BJIDENTIFY is called to produce four different plots of residuals. If WINDOWS is set to a scalar N, the graphs are all produced in window N on separate pages; the FRAME directive can be used to set the attributes of window N before calling the procedure. Alternatively, WINDOWS can be set to a variate of length four; the attributes of the four windows specified are then redefined within the procedure so that four graphs are produced on the same page. By default WINDOWS=1. The PENS option controls which pens are used for the plots; the attributes of these pens are modified appropriately within the procedure. By default pens 1-3 are used, but these can be changed by setting option PENS to a variate of length 3 containing the numbers of the three different pens required.

Options: PRINT, GRAPHICS, WINDOWS, PENS.
Parameters: SERIES, LENGTH, ORDERS, PARAMETERS, TSM, RESIDUALS.

Method
The model is fitted using the default settings of directive TFIT, and forecasts are constructed for increasing leadtimes using the directive TFORECAST. BJIDENTIFY is called to display time series statistics for the residual series after fitting the required ARIMA model.

Action with RESTRICT
Input structures must not be restricted. Restriction of the input SERIES to a contiguous set of units can be achieved using the LENGTH parameter.

See also
Directive: TFIT.
Procedures: BJFORECAST, BJIDENTIFY.
Genstat Reference Manual 1 Summary section on: Time series.

BJFORECAST

Plots forecasts of a time series using a previously fitted ARIMA (G. Tunnicliffe Wilson & S.J. Welham).

Options

PROBABILITY = *scalar*	Probability value used for forecast limits; default 0.9
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
WINDOW = *scalar*	Window to be used for plotting; default 1
PENS = *variate*	The three pens to be used (after being defined appropriately) for drawing the plots; default !(1,2,3)

Parameters

SERIES = *variates*	Variates holding the time series to be used for producing forecasts
LENGTH = *scalars* or *variates*	Specifies the units to be used from each series: a scalar N specifies that the first N units of the series are to be used, a variate of length 2 gives the time index of the first and last units of the subseries to be used; by default the whole series is used
TSM = *TSMs*	ARIMA model to be used for forecasting
TIMERANGE = *variates*	The first and second elements of each variate specify respectively the first and last time index, relative to the whole series, of the range to be forecast
ORIGIN = *scalars*	The time of the latest observation to be used to construct forecasts with increasing leadtimes for each series; if ORIGIN is unset, the default is to take the latest time point in the series prior to the range given by TIMERANGE, unless parameter LEADTIME is set, in which case fixed leadtime forecasts are constructed
LEADTIME = *scalars*	The fixed leadtime to be used to construct forecasts if ORIGIN is unset
FORECAST = *variates*	Save the values of the constructed forecasts
LOWER = *variates*	Save the lower limits of the forecasts
UPPER = *variates*	Save the upper limits of the forecasts
SFE = *variates*	Save the standardized forecast errors, available only for LEADTIME=1

Description

For a time series variate, given by the SERIES parameter, BJFORECAST plots forecasts calculated from a previously fitted ARIMA model, specified by the TSM parameter. The set of time points for which forecasts are produced is defined by setting the TIMERANGE parameter to a variate of length 2 holding the first and last time index. If only part of the series is to be used to initialize for forecasting, this is specified by setting parameter LENGTH, either to a scalar N to indicate that the first N values are to be used, or to a variate of length 2 holding the positions of the first and last units to be included. The procedure also prints a description of the series, and details of the model involved in the initialization for forecasting.

There are two options to control the type of forecasting. Setting the ORIGIN parameter to a scalar indicates that forecasts are calculated from this time point (at increasing leadtimes) for the range of future times specified by the TIMERANGE parameter. Alternatively, if ORIGIN is unset, it is possible to produce forecasts with a fixed leadtime, by setting the parameter LEADTIME to

the required value. If neither ORIGIN nor LEADTIME are set, a default origin is taken, namely the last element before the time range to be forecast. Where possible, the values of the supplied series are also plotted for comparison. If one-step-ahead forecasts are requested (fixed leadtime set to 1), the standardized forecast errors are plotted as a tracking signal for use in checking the continuing adequacy of the model.

The FORECAST parameter can be used to save the calculated forecasts in a variate and parameters LOWER and UPPER can save the lower and upper confidence limits for these forecasts. If the forecasts are from a fixed leadtime of 1, the standardized forecast errors can be saved in a variate given by parameter SFE; because of the way in which the standard errors are calculated, the last value of this variate is always missing. The PROBABILITY option indicates the probability value to be used for the confidence limits, with 0.9 as the default value.

Option GRAPHICS controls whether plots are produced for line printer or for the current high-resolution graphics device; by default high-resolution plots are produced. The window to be used for high-resolution plots is specified by the WINDOW option; by default WINDOW=1. The FRAME directive can be used to set the attributes of this window before calling the procedure, and these will be unchanged on leaving the procedure. The PENS option controls which pens are to be used for the plots; the attributes of these pens are modified within the procedure. By default pens 1-3 are used, but these can be changed by setting option PENS to a variate of length 3 containing the numbers of the three different pens required.

Options: PROBABILITY, GRAPHICS, WINDOW, PENS.
Parameters: SERIES, LENGTH, TSM, TIMERANGE, ORIGIN, LEADTIME, FORECAST, LOWER, UPPER, SFE.

Method

The values of the supplied series values, up to the origin, are used to initialize for forecasting (by residual regeneration), and the forecasts are then constructed over the requested time range. If fixed leadtime forecasts are required, the origin is successively updated for each forecast.

Action with RESTRICT

The input and output structures must not be restricted. Restriction of the input series to a contiguous set of units can be achieved using the LENGTH parameter.

See also

Directive: TFORECAST.
Procedures: BJESTIMATE, BJIDENTIFY.
Genstat Reference Manual 1 Summary section on: Time series.

BJIDENTIFY

Displays time series statistics useful for ARIMA model selection (G. Tunnicliffe Wilson & S.J. Welham).

Options

PRINT = *string token*	Controls printed output (description); default desc
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
WINDOWS = *scalar* or *variate*	Windows to be used for the plots: a scalar N indicates that plots are to be produced on separate pages in window N (as currently defined), whereas a variate specifies four separate windows to be redefined (within the procedure) for plotting four graphs on one page; default 1
PENS = *variate*	The three pens to be used (after being defined appropriately) for drawing the plots; default !(1,2,3)

Parameters

SERIES = *variates*	Variates holding the time series for which the statistics are to be produced
LENGTH = *scalars* or *variates*	Specifies the units to be used from each series: a scalar N indicates that the first N units of the series are to be used, a variate of length 2 gives the index of the first and last units of the subseries to be used; by default the whole series is used

Description

BJIDENTIFY displays time series statistics useful for ARIMA model selection. For a time series, specified (in a variate) using the SERIES parameter, four graphs are produced. These are of the series itself, its sample autocorrelation function and partial autocorrelation function, and its sample spectrum (or periodogram). The LENGTH parameter can specify that only part of the series is to be used: setting LENGTH to a scalar N indicates that the first N values are to be used; alternatively, a variate of length 2 can be specified holding the positions of the first and last units of the subseries. The maximum lag of the autocorrelations and the frequency grid for the periodogram are determined automatically by the procedure.

Printed output can be suppressed by setting the option PRINT=*; by default, PRINT=description, which gives a description of the series.

Graphical output is controlled by the options GRAPHICS, WINDOWS and PENS. Option GRAPHICS controls whether plots are produced for line-printer output or on the current high-resolution graphics device; by default high-resolution plots are given. Option WINDOWS controls the way in which the high-resolution plots are arranged. If WINDOWS is set to a scalar N, all the graphs are produced in window N on separate pages; the FRAME directive can then be used to set the attributes of window N before calling the procedure. Alternatively, WINDOWS can be set to a variate of length four; the attributes of the four windows specified are then redefined within the procedure so that four graphs are produced on the same page. By default WINDOWS=1. The PENS option controls which pens are to be used for the plots; the attributes of these pens are modified within the procedure. By default pens 1-3 are used, but these can be changed by setting option PENS to a variate of length 3 containing the numbers of the three different pens required.

Options: PRINT, GRAPHICS, WINDOWS, PENS.
Parameters: SERIES, LENGTH.

Method
The autocorrelation and partial autocorrelation functions are calculated using the CORRELATE directive. The maximum lag is chosen to be half the length of the series, but adjusted for very short or very long series. The number of periodogram ordinates is chosen to be approximately four times the length of the series. Before calculation of the periodogram, using the FOURIER directive, the series is mean corrected and missing values are replaced by zero.

Action with RESTRICT
Input structures must not be restricted. Restriction of the input SERIES to a contiguous set of units can be achieved using the LENGTH parameter.

See also
Procedures: BJESTIMATE, BJFORECAST.
Genstat Reference Manual 1 Summary section on: Time series.

BKDISPLAY

Displays an identification key (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (indented, bracketed, diagram, graph); default * i.e. none

Parameter

KEY = *tree*	Key to be displayed

Description

BKDISPLAY displays an identification key, as constructed by the BKEY procedure. The key can be saved from BKEY as a Genstat tree structure (using the KEY option of BKEY), and is supplied to BKDISPLAY using the KEY parameter. The type of output is specified by the PRINT option, with settings:

indented	indented key – prints the key branch by branch;
bracketed	bracketed key – prints the key test by test;
diagram	diagrammatic representation of the key;
graph	plots the key using high resolution graphics.

Option: PRINT.
Parameter: KEY.

Method

BKDISPLAY displays the key using procedures BPRINT and BGRAPH.

See also

Procedures: BKEY, BKIDENTIFY, BKKEEP, IDENTIFY.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BKEY

Constructs an identification key (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (indented, bracketed, diagram, graph); default * i.e. none
TAXONNAMES = *text*	Names of the taxa in the key; default * uses textual versions of the numbers 1, 2 onwards
GROUPS = *factor*	Groupings of the taxa, if the key is to identify the group of a specimen rather than its taxon
CRITERION = *string token*	Criterion to use to select the character to use at each node of the key (CME, CMV, GME); default GME when GROUPS is set, otherwise CME
PARTIAL = *string token*	Controls whether or not to use partial separation; (yes, no) default no
KEY = *tree*	Saves the key

Parameters

CHARACTER = *factors*	Characters available to construct the key
COST = *scalars*	Cost of each character; default 1

Description

Identification keys provide efficient ways of identifying objects, or *taxa*, whose properties can be described by a set of discrete-valued tests. Many applications are biological. For example, in botanical work, the taxa may be species of plant and the tests may require the observation of characters like the colours of petals or numbers of leaves. Similarly, in microbiology, the tests may involve the ability of an organism to grow in various media. Using a key involves doing a sequence of tests which continues until the unknown specimen can be identified.

The characters that are available for constructing the key are specified, as a list of factors, using the CHARACTER parameter. Each factor has a level for each possible value of the character concerned, and you can insert a missing value for a particular taxon to indicate that its value for the character is either variable or unknown. If an "extra" text has been defined for the factor (using the EXTRA parameter of the FACTOR directive), BKEY will use this when printing the textual forms of the key instead of the identifier of the factor. (So the characters can be described in the key using any printable symbol, not just those that may be used in identifiers.) The COST parameter allows you to specify a cost for each character. This may be how much it costs to observe or may simply record your own personal preferences between the parameters. By default all the costs are 1. The names of the taxa can be specified in a text using the TAXONNAMES option. If this is omitted, they are simply numbered 1, 2 and so on. If the taxa are classified into groups, BKEY can construct a key to identify the group of a specimen rather than the taxon itself. These groupings can be supplied using the GROUPS factor.

The efficiency of a key is usually measured by its expected cost of identification. To find the optimal key using a particular set of data essentially requires the construction and comparison of all possible keys for the taxa that could be formed with the available tests. This is impracticable even for moderate numbers of tests and taxa. Thus, heuristic algorithms are used which construct the key sequentially, selecting first the test that "best" divides the taxa into sets (where set *k* for test *i* contains all the taxa that can give result *k* to test *i*), then selecting the best test to use with each set, continuing until the sets each contain only one taxon – or until no further separation is possible. The "best" test can be defined using a *selection criterion function* (Gower & Payne 1975). BKEY provides three criteria, which can be selected using the CRITERION option, with settings:

CME	is an estimate of the expected cost of completing the identification from the current point of the key, assuming that test *i* is used and that, below this point, the key is completed optimally (this is the function CMe devised by Payne 1981);
CMV	is a less optimistic estimates, which assumes that the key is completed by simple binary tests (i.e. tests for each of which one particular taxon always gives a positive response and other taxa give negative responses) which corresponds to the function CMv′ of Payne (1981);
GME	is an equivalent version of CMv for the identification of groups of taxa (see Payne, Yarrow & Barnett 1982).

CMe and CMv′ (and two other criteria) were studied by Payne & Thompson (1989), who found that each of them produced the best key for some sets of data. They thus concluded that programs for key construction should allow their users to try several so that they can choose the one that behaves best with any particular set of data.

Usually construction of the key stops when the possible taxa at that point share identical values or have missing values for all the characters. However, if the missing values represent variable rather than unknown values, it may still be worth using these tests in case a specimen of the taxon concerned is obtained that happens to give a level different from the shared level. This *partial* separation can be requested by setting option PARTIAL=yes.

The key can be printed in various formats, as requested by the PRINT option, or it can be saved using the KEY option. The settings of PRINT are:

indented	indented form – prints the key branch by branch;
bracketed	bracketed form – prints the key test by test;
diagram	diagrammatic representation;
graph	plots the key using high resolution graphics.

BKEY stores the information required for printing as part of the tree. The labels for the diagram are formed as "identifier==n_1", where n_1 is the first level of the factor. The lines of the indented and bracketed keys are formed similarly if the factor has no extra test and no labels. Otherwise, the form is "*fname lname*", where *fname* is the extra text if this has been defined (by the EXTRA parameter of the FACTOR command) or else the identifier of the factor, and *lname* is the label if available or the level if not.

Options: PRINT, TAXONNAMES, GROUPS, CRITERION, PARTIAL, KEY.
Parameters: CHARACTER, COST.

Method

BKEY calls procedure BCONSTRUCT to form the key. This uses a special-purpose procedure BSELECT, which is customized specifically for keys, and stored with BKEY. The methodology involved in the construction of keys is reviewed by Payne & Preece (1980). Statistical applications of keys are described by Payne (1992).

Action with RESTRICT

Any restrictions on the CHARACTER factors or on TAXONNAMES or GROUPS are removed.

References

Gower, J.C. & Payne, R.W. (1975). A comparison of different criteria for selecting binary tests in diagnostic keys. *Biometrika*, **62**, 665-671.

Payne, R.W. & Preece, D.A. (1980). Identification keys and diagnostic tables: a review (with discussion). *Journal of the Royal Statistical Society, Series A*, **143**, 253-292.

Payne, R.W. (1981). Selection criteria for the construction of efficient diagnostic keys. *Journal of Statistical Planning and Inference*, **5**, 27-36.

Payne, R.W., Yarrow, D. & Barnett, J.A. (1982). The construction by computer of a diagnostic key to the genera of yeasts and other such groups of taxa. *Journal of General Microbiology*, **128**, 1265-1277.

Payne, R.W. & Thompson, C.J. (1989). A study of selection criteria for constructing identification keys containing tests with different costs. *Computational Statistics Quarterly*, **5**, 43-52.

Payne, R.W. (1992). The use of identification keys and diagnostic tables in statistical work. In: *COMPSTAT 92 Proceedings in Computational Statistics* (Ed. Y. Dodge & J. Whittaker), Volume 2, 239-244. Heidelberg: Physica-Verlag.

See also

Directive: `IRREDUNDANT`.

Procedures: `BKDISPLAY`, `BKIDENTIFY`, `BKKEEP`, `BCLASSIFICATION`, `BCFOREST`, `BREGRESSION`, `IDENTIFY`.

Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BKIDENTIFY

Identifies specimens using a key (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (identification, transcript); if PRINT is unset in an interactive BKIDENTIFY will ask what you want to print, in a batch run the default is iden
KEY = *tree*	Specifies the key
IDENTIFICATION = *variate*	Saves the identification of each specimen
TERMINALNODE = *variate*	Saves numbers of the terminal nodes reached by the specimens

Parameter

CHARACTER = *factors*	Character values of the specimens

Description

BKIDENTIFY identifies specimens using an identification key, as constructed by the BKEY procedure. The key can be saved from BKEY as a Genstat tree structure (using the KEY option of BKEY), and supplied to BKIDENTIFY using its own KEY option. Alternatively, BKIDENTIFY will ask you for the identifier of the key if you do not specify KEY when running interactively.

The characteristics of the specimens can be specified by using the CHARACTER parameter. This must be set to a list of factors with names (and levels) identical to those used originally to construct the key. If you do not set CHARACTER when running interactively, BKIDENTIFY will ask you to examine the characters in turn, as required by the key.

The PRINT option controls printed output, with settings:

identification	prints the identifications obtained using the key;
transcript	prints the observed characteristics when supplied in response to questions in an interactive run.

If you do not set PRINT in an interactive run, BKIDENTIFY will ask what you would like to print. In batch, the default is to print the identifications.

The IDENTIFICATION option allows you to save the identifications (in a text), and the TERMINALNODE option allows you to save a variate containing the numbers of the terminal nodes that the specimens reached in the key.

Options: PRINT, KEY, IDENTIFICATION, TERMINALNODE.
Parameter: CHARACTER.

Method

BKIDENTIFY works its way through the key using the standard tree functions, BNBRANCHES and BNEXT. The QUESTION procedure is used to obtain any information that is required in an interactive run.

Action with RESTRICT

BKIDENTIFY takes account of any restrictions on the CHARACTER factors.

See also

Procedures: BKEY, BKDISPLAY, BKKEEP, IDENTIFY.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BKKEEP

Saves information from an identification key (R.W. Payne).

No options

Parameters

KEY = *trees*	Identification key from which the information is to be saved
SUMMARY = *variates*	Saves summary information about each key
CHARACTERS = *pointers*	Saves the identifiers of the characters in each key

Description

BKKEEP saves information from an identification key, constructed by the BKEY procedure. The key can be saved using the KEY option of BKEY, and is specified for BKKEEP using its KEY parameter.

The SUMMARY parameter saves a variate containing summary information. The first element contains the number of nodes, and the second contains the number of terminal nodes.

The CHARACTERS parameter saves a pointer containing the identifiers of the characters in the key.

Options: none.
Parameters: KEY, SUMMARY, CHARACTERS.

See also

Procedures: BKEY, BKDISPLAY, BKIDENTIFY.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

BLANDALTMAN

Produces Bland-Altman plots to assess the agreement between two variates (A.R.G. McLachlan).

Options

PRINT = *string tokens*	Controls printed output (summary, estimates); default * i.e. none
PLOT = *string tokens*	What to plot (blandaltman, normal); default blan
DMETHOD = *string token*	Method for calculating differences (differences, ratios, %differences, percentages); default diff
LMETHOD = *string token*	Method for calculating limits of agreement when regression is not used (normaldistribution, percentile); default norm
REGMETHOD = *string tokens*	Whether to use regression to calculate bias (i.e. mean) or limits (bias, mean, limits, auto); default * i.e. none
CIPROBABILITY = *scalar*	Probability level for limits of agreement, confidence intervals and percentiles; default 0.95
LOWERLIMIT = *scalar*	Lower limit of agreement to use instead of a calculated limit
UPPERLIMIT = *scalar*	Upper limit of agreement to use instead of a calculated limit
ALPHALEVEL = *scalar*	Critical probability level used for regression when REGMETHOD=auto; default 0.05
XBLANDALTMAN = *string token*	X-values to use for the Bland-Altman plot (mean, Y1, Y2); default mean
REFERENCELINECHOICE = *string tokens*	Reference lines to plot on a Bland-Altman plot (bias, mean, limits, zero); default bias
GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
WINDOW = *scalar*	Window for the plot; default 3
SCREEN = *string token*	Whether to clear or keep the screen before displaying the plot (keep, clear); default clea
PENZEROLINE = *scalar*	Pen to use for the zero reference line
PENMEANLINE = *scalar*	Pen to use for the mean reference line
PENLIMITLINES = *scalar*	Pen to use for the reference lines showing limits of agreement

Parameters

Y1 = *variates*	First variate
Y2 = *variates*	Second variate
LABELS = *texts*	Labels for individual points on the Bland-Altman plot
MEANS = *variates*	Saves the means
DIFFERENCES = *variates*	Saves the differences, ratios or % differences (according to the DMETHOD option)
TITLE = *texts*	Title for the Bland-Altman plot
YTITLE = *texts*	Title for y-axis of the Bland-Altman plot
XTITLE = *texts*	Title for x-axis of the Bland-Altman plot
PEN = *scalars*, *variates* or *factors*	Pen for plotting points on the Bland-Altman plot; default 1

Description

Bland-Altman plots provide an effective way of assessing two different methods for measuring some quantity (Bland & Altman 1999; see also Altman & Bland 1983 and Bland & Altman 1986). The data are supplied by the `Y1` and `Y2` parameters, in two variates containing measurements on the same set of samples. The default display plots the differences between the measurements against their mean, so that the sizes of the discrepancies can be assessed while also seeing whether there is any bias or nonlinearity between the methods. Ideally, the points should lie within a rectangle arranged symmetrically around the x-axis i.e. similar amounts of scatter above and below the line of zero difference. The means and differences can be saved, in variates, using the `MEANS` and `DIFFERENCES` parameters, respectively.

The `DMETHOD` option controls the type of difference that is displayed, with settings:

`differences`	differences `Y1 - Y2` (default),
`ratios`	`Y1 / Y2`,
`%differences`	`(Y1 - Y2) / ((Y1 + Y2)/2) * 100`, and
`percentages`	synonym of `%differences`.

The plot can also show "limits of agreement" which are intended to represent boundaries on the acceptable difference between the methods. These can be supplied by the `LOWERLIMIT` and `UPPERLIMIT` options, Alternatively, if `LOWERLIMIT` and `UPPERLIMIT` are not set, the limits are calculated by the procedure according to the setting of the `LMETHOD` option:

`normaldistribution`	uses confidence limits calculated assuming that the differences have a Normal distribution (default), and
`percentile`	takes percentiles of the differences.

The `CIPROBABILITY` option specifies the probability for calculating the limits of agreement when `LMETHOD=norm`, or the percentiles used for the limits when `LMETHOD=perc`. The default of 0.95 gives 95% limits of agreement, and percentiles of 2.5 and 97.5%.

The `REFERENCELINECHOICE` option allows reference lines can be included on the Bland-Altman plot:

`mean` or `bias`	plots a line at the overall mean of the differences (default),
`limits`	plots upper and lower limits of agreement, and
`zero`	plot horizontal line at zero, or one when `DMETHOD=ratio`.

If there seems to be a trend in the plot (differences becoming larger or smaller as the means increase), it can be useful to fit a linear regression (on the mean) to the bias, or to the variation in the bias, or both. This is controlled by the `REGMETHOD` option. Setting `REGMETHOD` to `mean` or `bias` fits a line through the Bland-Altman plot to estimate the mean or bias. Limits of agreement are then calculated assuming a constant variance and a Normal distribution so that, if references lines are plotted for the limits, they will be parallel to the reference line for the mean. Alternatively, if `REGMETHOD=limits`, linear regression is used to estimate the variation in the differences. The limits then form a 'fan-shape' pattern about the horizontal bias line. These two settings can be combined (`REGMETHOD=bias,limits`) so that linear regression is used to estimate both the bias and the variation in the differences. Finally, if you set `REGMETHOD=auto`, the procedure automatically determines whether or not linear regression should be used to estimate either the bias or the variation or both. The `ALPHALEVEL` option then specifies the critical value for testing the significance of the regressions (default 0.05 i.e. 5%), to decide whether they should be used.

The `PLOT` option controls the plots that are produced:

`blandaltman`	produces the Bland-Altman plot (default), and
`normal`	produces a Normal (q-q) plot of the differences.

The x-values to be used in the Bland-Altman plot are controlled by the `XBLANDALTMAN` option. The default is to use the averages of the `Y1` and `Y2` variates (as recommended by Bland & Altman 1995). Alternatively, the settings `Y1` and `Y2` allow one of the two variates to be used instead; Krouwer (2008) recommended plotting against measurements from a reference method,

if this has provided much better precision.

By default high-precision graphics are used, but you can set option GRAPHICS=lineprinter to produce character-based graphs in the output window instead. The WINDOW option can be used to specify which graphics window to use for a high-resolution graph, and the SCREEN option allows you to stop the screen being cleared before plotting the Bland-Altman graph. Note that this does not to apply to the Normal probability plots, as the DPROBABILITY procedure (that is used to produce the plot) does not support the SCREEN option.

There are several options and parameters that can be used to modify the appearance of the Bland-Altman plot. The TITLE parameter can supply an overall title, and the YTITLE and XTITLE parameters can supply titles for the y- and x-axis. You can specify a text containing labels for the points in the Bland-Altman plot using the LABELS parameter. The PEN parameter allows you to specify a pen or pens for the points (default 1). The PENZEROLINE, PENMEANLINE and PENLIMITSLINES options specify pens for the reference lines at zero, mean difference and limits of agreement, respectively. If these options are not set, BLANDALTMAN uses the line colours, thicknesses and styles (if set) from pens 1, 2 and 3, respectively.

The PRINT option controls the printing of the results, with settings:

estimates	to print the estimates, and
summary	to print a summary showing the number and percentage of values above and below zero, and outside the limits of agreement.

When regression is being used, the estimates consist of the slope of the line, with its standard error and confidence interval, together with the sample size. Otherwise, they consist of the mean difference, limits of agreement, standard error of the differences and the sample size. By default, nothing is printed.

Note that the procedure does not cater for repeated measures of subjects. See Bland & Altman (1999, 2007) for information on how different types of repeated measures can be handled.

Options: PRINT, PLOT, DMETHOD, LMETHOD, REGMETHOD, CIPROBABILITY, LOWERLIMIT, UPPERLIMIT, ALPHALEVEL, XBLANDALTMAN, REFERENCELINECHOICE, GRAPHICS, WINDOW, SCREEN, PENZEROLINE, PENMEANLINE, PENLIMITSLINES.

Parameters: Y1, Y2, LABELS, MEANS, DIFFERENCES, TITLE, YTITLE, XTITLE, PEN.

Action with RESTRICT

Y1 and Y2 factor can be restricted to exclude units from the analysis. Restrictions on LABELS and PEN are ignored.

References

Altman, D.G. & Bland, J.M. (1983). Measurement in medicine: the analysis of method comparison studies. *Statistician*, **32**, 307–317.

Bland, J.M. & Altman, D.G. (1986). Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, **i**, 307–310.

Bland J.M. & Altman D.G. (1995). Comparing methods of measurement – why plotting difference against standard method is misleading. *Lancet*, **346**, 1085–1087.

Bland, J. M. & Altman, D. G. (1999). Measuring agreement in method comparison studies. *Statistical Methods in Medical Research*, **8**, 135–160.

Bland, J.M. & Altman, D.G. (2007). Agreement between methods of measurement with multiple observations per individual. *Journal of Biopharmaceutical Statistics*, **17**, 571–582.

Krouwer, J.S. (2008). Why Bland–Altman plots should use X, not (Y+X)/2 when X is a reference method. *Statistics in Medicine*, **27**, 778–780.

See also
Procedure: `LCONCORDANCE`.

BNTEST

Calculates one- and two-sample binomial tests (D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (test, summary, confidence); default test, summ, conf
METHOD = *string token*	Type of test required (twosided, greaterthan, lessthan); default twos
TEST = *string token*	Form of the test for one-sample test (exact, normalapproximation) or for two-sample (normalapproximation, oddsratio); default norm
CIPROBABILITY = *scalar*	The probability level for the confidence interval; default 0.95
NULL = *scalar*	The value of the probability of success under the null hypothesis for the one-sample test; default 0.5

Parameters

R1 = *scalars* or *variates*	Number of successes (scalar) or results (variate) for the first sample
N1 = *scalars*	Sample size of the first sample
R2 = *scalars* or *variates*	Number of successes (scalar) or results (variate) for the second sample
N2 = *scalars*	Sample size of the second sample
STATISTIC = *scalars*	Saves the Normal approximation from the one-sample or two-sample tests, or the odds ratio
PROBABILITY = *scalars*	Saves the probability value from the one-sample or two-sample tests
LOWER = *scalars*	Saves the lower limit of the confidence interval
UPPER = *scalars*	Saves the upper limit of the confidence interval

Description

BNTEST calculates one- and two-sample binomial tests, and odds ratios. For a one-sample test, the number of successes r_1 can be specified using the R1 parameter, and the sample size n_1 using the N1 parameter (both as scalars). Alternatively you can supply the raw data, by setting R1 to a variate containing one in the units corresponding to successful trials and zero in those for unsuccessful trials. The test is for the probability of success under a binomial distribution. The value for the probability under the null hypothesis is 0.5 by default, but you can specify other probabilities using the NULL option. With a two-sample test, R1 and N1 similarly provide the number of successes and sample size for the first sample (r_1 and n_1), and R2 and N2 those for the second sample (r_2 and n_2).

For both one- and two-sample cases, the test is assumed to be two-sided unless otherwise requested by the METHOD option. Setting METHOD=greaterthan gives a one-sided test of the null hypothesis that $r_1/n_1 > r_2/n_2$ or NULL (for a two-sample or one-sample test, respectively). Similarly, METHOD=lessthan produces a test of the null hypothesis $r_1/n_1 < r_2/n_2$ or NULL. A small "p-value" indicates that the data are inconsistent with the null hypothesis.

The TEST option specifies the form of test to be used. For the one-sample test, an exact test or Normal approximation can be selected. For a two-sample test, a Normal approximation or odds ratio can be chosen.

Printed output is controlled by the PRINT option with settings:

summary	number of successes, sample size, proportion, standard error (for Normal approximation and odds ratio) and odds

	ratio (when `TEST=ODDSRATIO` is selected);
`test`	test and probability level;
`confidence`	confidence interval for the probabilities of success; for the odds ratio the confidence interval is displayed for the true log-odds ratio and odds ratio.

The default is to print everything.

By default a 95% confidence interval is calculated, but this can be changed by setting the `CIPROBABILITY` option to the required value (between 0 and 1).

Results can be saved using the `STATISTIC`, `PROBABILITY`, `LOWER` and `UPPER` parameters. `STATISTIC` saves the Normal approximation for the one- and two-sample tests or the odds ratio, `PROBABILITY` saves the probability level. `LOWER` and `UPPER` save the lower and upper limits, respectively, of the confidence interval; for the odds ratio the confidence interval is saved for the true odds ratio.

Options: `PRINT`, `METHOD`, `TEST`, `CIPROBABILITY`, `NULL`.
Parameters: `R1`, `N1`, `R2`, `N2`, `STATISTIC`, `PROBABILITY`, `LOWER`, `UPPER`.

Method

A standard Normal approximation is used for both the one- and two-sample tests. The exact test and confidence intervals are based on the methodology described in Chapter 4 (page 121) of Armitage & Berry (1994). The odds ratio is a relative measure of the odds of a success in one set of data relative to that in the other. The estimate of the ratio is defined as

$$p_1 (1 - p_1) / p_2 (1 - p_2)$$

where p_1 and p_2 are the success probabilities in two sets of data. The calculation of the approximate standard error of the estimated log-odds ratio and confidence intervals is described in Chapter 2 (page 36) of Collett (1991).

References

Armitage, P. & Berry, G. (1994). *Statistical Methods in Medical Research*. Blackwell Science, Oxford.

Collett, D. (1991). *Modelling Binary Data*. Chapman & Hall, London.

See also

Procedures: `PNTEST`, `SBNTEST`, `TTEST`.

Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Regression analysis.

BOOTSTRAP

Produces bootstrapped estimates, standard errors and distributions (P.W. Lane).

Options

PRINT = *string token*	Controls printed output (estimates, graphs, vcovariance); default esti
DATA = *variates*, *factors* or *texts*	Data vectors from which the statistics are to be calculated; no default
AUXILIARY = *pointers*	Further sets of data vectors, each set to be resampled independently
ANCILLARY = *any type*	Other relevant information needed to calculate the statistics
NTIMES = *scalar*	Number of times to resample; default 100
SEED = *scalar*	Seed for random number generator; default continue from previous generation or use system clock
GRAPHICS = *string token*	Type of graphics (lineprinter, highresolution); default high
PROBABILITY = *scalar*	Probability level for confidence interval; default 0.95
METHOD = *string token*	What type of bootstrapping to use (random, balance, permute); default rand
BLOCKSTRUCTURE = *formula*	Block structure to use for random permutations
CIMETHOD = *string token*	What type of confidence intervals to provide (bca, percentile); default perc
VCOVARIANCE = *symmetric matrix*	Saves the variance-covariance matrix of the statistics

Parameters

LABEL = *texts*	Texts, each containing a single line, to label the statistics; default 'Statistic'
ESTIMATE = *scalars*	Saves the bootstrap mean for each statistic
SE = *scalars*	Saves the bootstrap standard error for each statistic
LOWER = *scalars*	Saves the bootstrap lower confidence limit for each statistic
UPPER = *scalars*	Saves the bootstrap upper confidence limit for each statistic
STATISTIC = *variates*	Saves the series of bootstrap estimates of each statistic
WINDOW = *scalars*	Graphical window to use for displaying bootstrap distribution for each statistic; default 4
SCREEN = *string tokens*	Whether to clear graphical frame or draw on top (clear, keep); default clea

Description

The bootstrap is a method of providing distributional information, such as standard errors, about statistical estimates – without making precise distributional assumptions about the data. It can also provide estimates with reduced bias. This is achieved by "resampling" from the data; that is, generating new data sets by sampling with replacement from the data set being investigated. A good introduction to the bootstrap is given by Efron & Tibshirani (1986); a fuller treatment can be found in Efron & Tibshirani (1993).

The BOOTSTRAP procedure can be used for any statistic or set of statistics that can be calculated by Genstat from one or more data matrices. You need to provide a procedure called RESAMPLE which calculates the statistics from the data, as explained in the Method section. There are also several examples of RESAMPLE in the standard examples, which can be extracted

by the commands:

```
LIBEXAMPLE 'BOOTSTRAP'; EXAMPLE=Ex
PRINT Ex; JUSTIF=left
```

The options and parameters of RESAMPLE must not be changed. The body of the procedure should store the required statistics in scalars called STATISTIC[1...s] using variates, factors and texts called DATA[1...d], where each of s and d can be any positive integer. The EXIT parameter of RESAMPLE should be set to indicate when any of the calculations fail, as can sometimes happen if degenerate data-sets are generated (see Example 3).

The data for BOOTSTRAP are provided as a list of vectors (variates, factors or texts) using the DATA option. From this, the procedure will generate new data by resampling from the set of units: all the vectors must have the same length, and each new sample uses the same set of units for all vectors. The procedure RESAMPLE is then called to calculate the statistics.

Extra information required in procedure RESAMPLE to calculate the statistics, which is not to be resampled along with the data matrix, can be passed as a list of data structures using the ANCILLARY option of BOOTSTRAP (see Examples 2 and 3).

The procedure can also deal with statistics calculated from several independent data matrices. For example, the difference in means between two independent samples must be dealt with by resampling independently from each sample, which may have different numbers of observations. In this case, one data matrix is specified as a list of vectors using the DATA option as usual, and the second data matrix is specified as a pointer using the AUXILIARY option. This option may be set to any number of pointers, each storing a list of vectors; resampling is done independently for each set of vectors (see Example 4).

The option NTIMES specifies how many times the resampling is carried out. The default value is 100, which has been found by many users of the bootstrap to be sufficient for producing standard errors and bias-reduced estimates. However, the number should be increased to get reliable distributional information: 1000 or more may be needed for reliable 95% confidence limits.

Printed output is controlled by the PRINT option, with settings estimates for the estimates and their standard errors and confidence limits, and vcovariance for the variance-covariance matrix. The graphs setting draws a histogram of the bootstrap distributions. The default setting is just estimates.

A label should be provided for each statistic, using the LABEL parameter; by default, bootstrapping will be done for a single statistic which will be labelled simply as Statistic. The estimates and their standard errors can be saved by the ESTIMATE and SE parameters. Also, a variance-covariance matrix of the estimates can be saved using the VCOVARIANCE option. The number of labels, s say, must match the number of statistics, called STATISTIC[1...s], calculated in your version of the RESAMPLE procedure.

The parameters LOWER and UPPER allow confidence limits for each statistic to be saved, with the probability level specified in the PROBABILITY option (default 0.95 i.e. 95% confidence intervals). By default the intervals are constructed as percentiles of the empirical distribution of the bootstrap estimates. However, provided there are no auxiliary data vectors, you can request bias-corrected and accelerated limits instead by setting option CIMETHOD=bca (see Efron & Tibshirani, 1993, Section 14.3). The full sets of bootstrap estimates can be saved by setting the STATISTICS parameter; each variate will contain *n* values, where *n* is the setting of the NTIMES option.

Three methods of bootstrapping are provided. By default, resampling is completely pseudo-random, using Genstat's random-number generator. The generator can be initialized by setting option SEED, thereby producing reproducible results; otherwise, the initialization uses the system clock. A second alternative is balanced bootstrapping, requested by setting METHOD=balance. In this case, the resampling is constrained to ensure that each unit of the data matrix occurs the same number of times in the complete set of generated samples (see Examples 3 and 4). The

third method, specified by METHOD=permute, is simply to permute the units of the data matrix. Note that this method gives no variation in results if the statistics are independent of the order of the data, like the sample mean. However, this method provides permutation tests, a type of randomization test that can be applied to grouped data (see Example 4). When METHOD=permute, you can set the BLOCKSTRUCTURE option to a model formula to define how the randomization is to be done (see the RANDOMIZE directive for details).

If the graphics setting of the PRINT option is used, the procedure will display the distribution of each set of bootstrap estimates as a histogram. By default, this will be a high-resolution plot on the current device, but the GRAPHICS option can be set to line to produce a line-printer histogram. In a high-resolution plot, the histogram is enhanced with a smoothed line, giving a clearer indication of the distribution of the statistic. By default, the display for the statistics will appear in graphical window 4, one at a time (this window is set by default to fill the whole graphical frame). But the WINDOW and SCREEN parameters can be set to arrange for concurrent displays of the statistics in differently sized windows.

Options: PRINT, DATA, AUXILIARY, ANCILLARY, NTIMES, SEED, GRAPHICS, PROBABILITY, METHOD, BLOCKSTRUCTURE, CIMETHOD, VCOVARIANCE.
Parameters: LABEL, ESTIMATE, SE, LOWER, UPPER, STATISTIC, WINDOW, SCREEN.

Method

Samples are generated by scaling uniform random numbers produced by the URAND function. For the balanced bootstrap, a list of repeated unit numbers is sorted into random order and used one block at a time. For the permutation test, the RANDOMIZE directive is used to re-order the data at random.

BOOTSTRAP needs a subsidiary procedure RESAMPLE to calculate the statistics of interest. RESAMPLE has an option, DATA, which is used to supply the data vectors (variates, factors or texts) from which the statistics are to be calculated. Other relevant information can be supplied through the AUXILIARY and ANCILLARY options, which correspond to the AUXILIARY and ANCILLARY options of BOOTSTRAP itself. There are two parameters: STATISTIC supplies a list of scalars to store the estimates of each statistic, and EXIT a list of scalars which should be set to zero or one according to whether or not each statistic could be estimated successfully with the supplied data vectors. If the value of EXIT is not calculated in RESAMPLE, the BOOTSTRAP procedure assumes that the calculations succeeded.

This example shows a version of RESAMPLE which calculates the correlation between two variates.

```
PROCEDURE [PARAMETER=pointer] 'RESAMPLE'
OPTION    'DATA',       " (I: variates, factors or texts) data
                          vectors from which to calculate
                          the statistics; no default"\
          'AUXILIARY', " (I: pointers) auxiliary sets of data
                          vectors, each of which is to be
                          resampled independently"\
          'ANCILLARY'; " (I: any type of structure) other
                          relevant information needed to
                          calculate the statistics "\
          MODE=p; TYPE=!t(variate,factor,text),'pointer',*;\
          SET=yes,no,no; LIST=yes; DECLARED=yes; PRESENT=yes
PARAMETER 'STATISTIC', " (O: scalars) to save the calculated
                          statistics "\
          'EXIT';       " (O: scalars) to save an exit code
                          to indicate failure (EXIT[i]=1) or
                          success (EXIT[i]=0) when calculating
                          each STATISTIC[i]"\
          MODE=p; TYPE='scalar'; SET=yes
  CALC STATISTIC[1] = CORRELATION(DATA[1]; DATA[2])
```

```
  & EXIT[1] = STATISTIC[1]==C('missing')
ENDPROCEDURE
VARIATE [VALUES=576,635,558,578,666,580,555,661,651,605, \
  653,575,545,572,594] Y
& [VALUES=3.39,3.30,2.81,3.03,3.44,3.07,3.00,3.43,3.36,3.13,\
  3.12,2.74,2.76,2.88,2.96] Z
BOOTSTRAP [DATA=Y,Z; SEED=77320] 'Correlation'
```

The RESAMPLE procedure is called within a loop, and the statistics that are returned are loaded into variates. If any statistics fail to be calculated, as recorded by the EXIT parameter of RESAMPLE, they are stored as missing values. BOOTSTRAP will then base its estimation on the successful generations, but reports how many failures occurred.

The bootstrap estimates are formed as simple means of the stored variates, and the s.e.s are square roots of the sample variance. The TABULATE directive is used to estimate quantiles from the stored variates, to define confidence limits. The variance-covariance matrix is formed from the statistics using the FSSPM directive.

The graphical representation uses DHISTOGRAM or HISTOGRAM on the stored variates. The smoothed curves are calculated from the transformed percentages from the histogram: LOGIT(CUM(%)). A smoothing spline is fitted on this scale, by the FIT directive with the SSPLINE function, using 4 d.f. The resulting fitted values are then backtransformed and drawn on the plot with the monotonic setting of the PEN directive.

Action with RESTRICT

If any of the data vectors is restricted, BOOTSTRAP will use only the units that are not restricted for any of the vectors. The data vectors that are passed to the RESAMPLE procedure are all restricted to this identified set of units, but otherwise match the original data vectors. Each set of vectors supplied in pointers in the AUXILIARY option are treated separately in this way.

References

Efron, B. & Tibshirani, R. (1986). Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, **1**, 54-77.

Efron, B. & Tibshirani, R.J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, London.

See also

Procedures: JACKKNIFE, APERMTEST, CHIPERMTEST, HBOOTSTRAP, RPERMTEST.

BOXPLOT

Draws box-and-whisker diagrams or schematic plots (P.W. Lane & S.D. Langton).

Options

GRAPHICS = *string token*	What type of graphics to use (highresolution, lineprinter); default high
TITLE = *text*	Title for diagram; default *
AXISTITLE = *text*	Title for axis representing data values; default *
WINDOW = *scalar*	Window in which to draw a high-resolution plot; default 4
SCREEN = *string token*	Whether to clear screen before a high-resolution plot (clear, keep); default clea
ORIENTATION = *string token*	Orientation of plots (horizontal, vertical, across, down); default vert
YORIENTATION = *string token*	Direction of the y-axis for horizontal plots (reverse, normal); default reve
METHOD = *string token*	Type of representation of data in a high-resolution plot (boxandwhisker, schematic); default boxa
SCREEN = *string token*	Whether to clear screen before a high-resolution plot (clear, keep); default clea
BOXTITLE = *text*	Title for axis representing different variates or groups; default *
BOXWIDTH = *string token*	Whether to relate box width to size of sample in high-resolution plot (fixed, variable); default fixe
WHISKER = *number*	Linestyle for whiskers (0...10); default 1
BAR% = *scalar*	Size of bar at the end of the whiskers, as a percentage of the box-width; default 0 (i.e. no bar)
WIDTH% = *scalar*	Width of the boxes, expressed as a percentage of the default width; default 100
SEM = *string token*	Add bar showing a nonparametric standard error of the median (yes, no) default no
BOXORDER = *string token*	Sort order for boxes when there are several DATA variates and GROUPS (groups, variates); default vari
REFERENCELINE = *scalar*	Specifies the position of a reference line to be drawn parallel to the box axis; default * i.e. none

Parameters

DATA = *variates*	Data to be summarized; no default
GROUPS = *factor*	Factor to divide values of a single variate into groups; default *
BOXLABELS = *texts*	Labels for individual boxes; default *, i.e. identifiers of variates or labels or levels of factor
UNITLABELS = *texts*	Labels for extreme points in schematic plot; default is to use unit labels
BOXPOSITIONS = *variates*	Positions of the boxes on the appropriate axis; default defines positions in an equal spacing

Description

BOXPLOT draws pictures to display the distribution of one or more sets of data. In the simplest case, with the DATA parameter set to a single variate, BOXPLOT will draw a box-and-whisker

diagram, as defined by Tukey (1977). The box spans the interquartile range of the values in the variate, so that the middle 50% of the data lie within the box, with a line indicating the median. Whiskers extend beyond the ends of the box as far as the minimum and maximum values. If several variates are supplied, a box is drawn for each of them using the same scale. Alternatively, if a single variate is supplied by the DATA parameter, a factor with the same number of values as the variate may be provided by the GROUPS parameter, and a box will be drawn for each level of the factor. If you specify several DATA variates, and GROUPS factors, the BOXORDER option controls whether the boxes are arranged as groups within variates (BOXORDER=variates, the default) of variates within groups (BOXORDER=groups).

The GRAPHICS option indicates whether high-resolution or line-printer plots are required. The TITLE, AXISTITLE and BOXTITLE options can be set to specify the titles displayed at the top of the plot, along the axis representing the data values, and along the axis representing separate boxes when there are several variates or groups, for either graphics mode. For high-resolution plots, the WINDOW and SCREEN options control the placement of the picture in the graphical frame.

It is not possible to produce line-printer plots with more than 14 boxes. If the page size is small, as in interactive mode, vertical line-printer plots may be very cramped: the PAGE option of the OUTPUT directive can be used to increase the depth of the graphs.

The ORIENTATION option controls the orientation of the boxes, with the following settings:

vertical	plots the boxes vertically i.e. down the screen (default),
horizontal	plots the boxes horizontally i.e. across the screen,
down	synonym of vertical, and
across	synonym of horizontal.

When ORIENTATION=horizontal, the horizontal axis is taken to be the y-axis, so the same XAXIS and YAXIS settings can be used however the boxes are oriented.

The YORIENTATION option controls the orientation of the y-axis when the boxes are plotted horizontally. By default this is reversed, so that the first box is at the top of the screen.

Schematic plots can be drawn (high-resolution only) by setting option METHOD=schematic. These diagrams (also defined by Tukey 1977) are modifications of box-and-whisker diagrams which display individual outlying points as well as the box. The whiskers extend only to the most extreme data values within the inner "fences", which are at a distance of 1.5 times the interquartile range beyond the quartiles, or the maximum value if that is smaller. Individual outliers are plotted with a cross by default, with labels specified by the UNITLABELS parameter. The default for UNITLABELS is to use the unit labels of the DATA variate. The labels can be suppressed by setting option UNITLABELS=*. "Far" outliers, beyond the outer "fences" which are at a distance of three times the interquartile range beyond the quartiles, are plotted with a different pen.

The SEM option adds a central bar to each boxplot, giving a nonparameteric estimate of the standard error of the median. This is calculated as the distance between the quartiles, multiplied by 1.5, and divided by the square root of the number of values in the DATA variate.

By default, all boxes have equal width. High-resolution diagrams can be modified to indicate the number of values being represented by each box. The option BOXWIDTH=variable will scale the box widths by the square root of the number of values represented.

The style of the whiskers can be controlled by setting the WHISKER option to a graphical linestyle in the range 0 to 10. These styles are device dependent, but 0 and 1 always give a solid line (the default) and 2 usually gives a dashed line. The BAR% option allows you to add bars at the end of the whiskers. For example, the setting 100 gives a bar as wide as the box, and 25 would give one a quarter the width. The default is 0, giving no bars. The WIDTH% option specifies the width of the boxes, as a percentage of the default width (default 100).

The REFERENCELINE option allows you to specify the position of a reference line to be drawn parallel to the box axis in a high-resolution plot. If this is not set, no line is drawn.

Six pens are used to draw the high-resolution displays, apart from the axes: pen 1 for the boxes and median line (default colour black), pen 2 for far outliers (red crosses), pen 3 for outliers (green crosses) and pen 4 for the whiskers (set to match the colour of pen 1), pen 5 for the standard error of median bar, and pen 6 for the reference line. You can customize the pictures by setting some aspects of these pens with the PEN directive before calling the procedure: in particular, the colours, symbols and line-thicknesses.

The BOXLABELS parameter allows you to specify labels that will identify each box.

The UNITLABELS parameter allows you to specify labels that will be used to identify outlying observations in schematic plots (but this is not available if you gave a list of variates in the DATA parameter).

The BOXPOSITIONS parameter defines the positions of the boxes on the appropriate axis. If this is unset, the positions are defined with an equal spacing.

Options: GRAPHICS, TITLE, AXISTITLE, WINDOW, ORIENTATION, YORIENTATION, METHOD, SCREEN, BOXTITLE, BOXWIDTH, WHISKER, BAR%, WIDTH%, SEM, BOXORDER, REFERENCELINE.
Parameters: DATA, GROUPS, BOXLABELS, UNITLABELS, BOXPOSITION.

Method
The medians and extremes are calculated by functions MEDIAN, MINIMUM and MAXIMUM, whereas the quartiles are calculated using the PERCENT option of TABULATE.

Action with RESTRICT
Restrictions on the supplied variates are taken into account. The grouping factor and texts holding boxlabels or unitlabels, if specified, should not be restricted.

Reference
Tukey, J.W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

See also
Procedures: DOTHISTOGRAM, RUGPLOT, STEM, DXDENSITY.
Genstat Reference Manual 1 Summary section on: Graphics.

†BPCONVERT

Converts bit patterns between integers, pointers of set bits and textual descriptions (R.W. Payne).

Options

`PRINT` = *string token*	Controls printed output (`description`); default `desc`
`BITS` = *text*, *variate* or *pointer*	Labels for the individual bits; default `!(1...31)`
`SEPARATOR` = *string*	Separator between the bit labels in the description; default `'.'`

Parameters

`DATA` = *scalars*, *texts* or *pointers*	Bit patterns to convert
`BP` = *scalars*	Bit patterns as integers
`CONTENTS` = *pointers*	Bits that are set in each bit pattern
`DESCRIPTION` = *text*	Textual description of each bit pattern

Description

Bit patterns are used by Genstat, and many other programs, to represent sets of objects like, for example, the factors and variates in a model term. Internally, they are stored in integers with a bit (i.e. a digit in the binary representation of the integer) for each object. The bit is set to one if the bit pattern contains the object, and zero otherwise. The contents of the bit patterns are then determined by logical `AND` operations, but this is not easy or convenient for implementers when debugging a program. `BPCONVERT` is therefore provided to enable a bit pattern to be converted between different representations: an integer (as above), a pointer with an element for each of the bits that is set in the bit pattern, a text containing the string (e.g. `blocks.plots`) that would represent the bit pattern in output.

The `DATA` parameter supplies the bit pattern to convert, in any of the three representations. The `BP` parameter can save it as an integer (in a scalar). The `CONTENTS` parameter can save it in a pointer. The `DESCRIPTION` parameter can save the textual description (in a text).

The `BITS` option provides the information to label the bits in `CONTENTS` and `DESCRIPTION`. This can be textual labels (in a text), numbers (in a variate) or identifiers of data structures (in a pointer); the default is to use the integers 1-31. The `SEPARATOR` option specifies the separator to use between the labels in descriptions; by default this is a dot.

Options: `PRINT`, `BITS`, `SEPARATOR`.
Parameters: `DATA`, `BP`, `CONTENTS`, `DESCRIPTION`.

See also

Procedure: `NCONVERT`.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

BPRINT

Displays a tree (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (indented, bracketed, labelleddiagram, numbereddiagram); default inde

Parameter

TREE = *trees*	Trees to be displayed

Description

BPRINT can print a tree in various formats. The tree is specified by the TREE parameter, and the PRINT option indicates what output is required, with settings:

bracketed	display as used to represent an identification key in "bracketed" form (printed node by node);
indented	display as used to represent an identification key in "indented" form (printed branch by branch);
labelleddiagram	diagrammatic display including the node labels;
numbereddiagram	diagrammatic display with the nodes labelled by their numbers.

Option: PRINT.
Parameter: TREE.

Method

BPRINT uses the standard tree functions to parse the tree for printing.

See also

Directive: TREE.
Procedures: BRDISPLAY, BKDISPLAY, BRDISPLAY, BGRAPH.
Genstat Reference Manual 1 Summary section on: Input and output.

BPRUNE

Prunes a tree using minimal cost complexity (R.W. Payne).

Option

`PRINT` = *string tokens*	Controls printed output (`graph`, `table`, `monitoring`); default `tabl`

Parameters

`TREE` = *trees*	Trees to be pruned
`ACCURACY` = *pointers*	Accuracy values for the nodes of each tree; default is to use those stored with the tree
`NEWTREES` = *pointers*	Saves the trees generated during the pruning of each tree
`RTPRUNED` = *variates*	Accuracy of the pruned trees of each tree
`NTERMINAL` = *variates*	Number of terminal nodes in the pruned trees of each tree

Description

The construction of a classification tree or a regression tree generally results in *over fitting*, that is it continues to extend the branches of the tree beyond the point that can be justified statistically. The solution is to prune the tree to remove the uninformative sub-branches.

The tree to be pruned is specified by the `TREE` parameter. `BPRUNE` assumes that there is an *accuracy* figure R(t) available for each node t of the tree. By default this is assumed to be stored with the tree itself, but you can specify other values using the `ACCURACY` parameter. This should be set to a pointer whose suffixes are the same as the numbers of the nodes in the tree, and whose elements are scalars storing the relevant accuracy values.

For a classification tree the accuracy measures the impurity of the subset of individuals at that node (how far it is from being homogeneous i.e. with individuals from a single group). For a regression tree it is the average squared distance of the values of the response variate from their mean for the subset of observations at that node. The accuracy R(T) of the whole tree T is the sum of the accuracies of its terminal nodes.

`BPRUNE` uses the principle of minimal cost complexity (Breiman *et al*. 1984, Chapter 3) to produce a sequence of pruned trees. At each stage it prunes at the node which is the *weakest link*. Define R(T_t) to be the accuracy of the subtree with root at node t, and nterm(t) to be its number of terminal nodes. The weakest link is then the node for which

$$(R(t) - R(T_t)) / (nterm(t) - 1)$$

is a minimum. The pruned trees can be saved, in a pointer, using the `NEWTREES` parameter. Their accuracies can be saved (in a variate) using the `RTPRUNED` parameter, and their numbers of terminal nodes can be saved (also in a variate) using the `NTERMINAL` parameter.

Printed output is controlled by the `PRINT` option, with settings:

`graph`	plots `RTPRUNED` against `NTERMINAL`;
`table`	prints a table with `RTPRUNED` and `NTERMINAL`;
`monitoring`	provides monitoring information during the pruning.

The plot of `RTPRUNED` against `NTERMINAL` demonstates the trade-off between accuracy and complexity (number of terminal nodes). It should show an initial rapid decrease, followed by a long flat region, and then often a gradual increase. The aim is to select a tree that is accurate but not over-complex. One possibility is to take the tree at the point where the graph levels off. However, `RTPRUNED` contains only an estimate of the accuracy of the trees. So Breiman *et al*. (1984) recommend taking a tree a little above that (in fact at one standard error of `RTPRUNED` above the minimium point in the graph: see Chapters 3 and 11). In practice though a small amount of over-fitting should not be a problem, so the exact choice of pruned tree should not be crucial.

Option: PRINT.
Parameter: TREE, ACCURACY, NEWTREES, RTPRUNED, NTERMINAL.

Method
BPRUNE uses the BSCAN function to move around the tree, function BBELOW to obtain the numbers of the nodes below each node, and directive BCUT to perform each pruning operation.

Reference
Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth, Monterey.

See also
Procedures: BCLASSIFICATION, BCVALUES, BREGRESSION, BRVALUES.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Multivariate and cluster analysis.

BRDISPLAY

Displays a regression tree (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (summary, details, indented, bracketed, labelleddiagram, numbereddiagram, graph); default * i.e. none

Parameter

TREE = *tree*	Tree to be displayed

Description

BRDISPLAY displays a regression tree, as constructed by the BREGRESSION procedure. The key can be saved from BREGRESSION (using the TREE option of BREGRESSION), and is specified for BRDISPLAY using the TREE parameter. The type of output is specified by the PRINT option, with settings:

summary	prints a summary of the properties of the tree;
details	gives detailed information about the nodes of the tree;
bracketed	display as used to represent an identification key in "bracketed" form (printed node by node).
indented	display as used to represent an identification key in "indented" form (printed branch by branch);
labelleddiagram	diagrammatic display including the node labels;
numbereddiagram	diagrammatic display with the nodes labelled by their numbers;
graph	plots the tree using high-resolution graphics.

Option: PRINT.
Parameter: TREE.

Method

BRDISPLAY displays the tree using procedures BPRINT and BGRAPH.

See also

Procedures: BREGRESSION, BRKEEP, BRPREDICT.
Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

BREGRESSION

Constructs a regression tree (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (summary, details, indented, bracketed, labelleddiagram, numbereddiagram, graph, monitoring); default * i.e. none
Y = *variate*	Response variate for the regression
TREE = *tree*	Saves the tree that has been constructed
MSLIMIT = *scalar*	Limit on the mean square of the observations at a node at which to stop making splits; default 0
NSTOP = *scalar*	Specifies the number of observations at a node at which to stop making splits; default 1
OWNBSELECT = *string token*	Indicates whether or not your own version of the BSELECT procedure is to be used, as explained in the Method section (yes, no); default no

Parameters

X = *variates* or *factors*	Independent variables available for constructing the tree
ORDERED = *string tokens*	Whether factor levels are ordered (yes, no); default no

Description

A regression tree is a mechanism for predicting a response variable from a set of independent variables (see Chapter 8 of Breiman *et al.*). The tree is constructed using data on a set of observations. Their values for the response variable are specified (in a variate) using the Y option, and their values for the independent variables are specified (in a list of variates or factors) using the X parameter. Factors may have either ordered or unordered levels, according to whether the corresponding value ORDERED parameter is set to yes or no. For example, a factor called Dose with levels 1, 1.5, 2 and 2.5 would usually be treated as having ordered levels, whereas levels labelled 'Morphine', 'Amidone', 'Phenadoxone' and 'Pethidine' of a factor called Drug would be regarded as unordered.

The construction process splits the observations into subsets. With an x-variate or a factor with ordered levels, the subsets are formed by taking the observations with values less then or greater than some split point *p*. For a factor with unordered levels, all possible ways of dividing its levels into two subsets are tried. The aim is to form subsets that have similar values for the response variate. The predicted value of the response variable for each node of the tree is the mean of its value for the subset of observations at that node. The *accuracy* of the node is the squared distance of the values of the response variate from their mean for the observations at the node, divided by the total number of observations. The potential splits at the node are assessed by their effect on the accuracy, that is the difference between the accuracy of the node and the sum of the accuracies of the two potential successor nodes. The node will become a terminal node if none of the splits provides any improvement in accuracy, or if the mean square of the observations at the node is less than or equal to a limit specified by the MSLIMIT option (default 0), or if the number of observations at the node is less than or equal to the number specified by the NSTOP option (default 1).

The resulting tree can be saved using the TREE option. Details of the tree can be printed as selected by the PRINT option, with settings:

summary	prints a summary of the properties of the tree;
details	gives detailed information about the nodes of the tree;
bracketed	display as used to represent an identification key in

	"bracketed" form (printed node by node).
`indented`	display as used to represent an identification key in "indented" form (printed branch by branch);
`labelleddiagram`	diagrammatic display including the node labels;
`numbereddiagram`	diagrammatic display with the nodes labelled by their numbers;
`graph`	plots the tree using high-resolution graphics.
`monitoring`	prints information monitoring the construction process.

`BREGRESSION` stores the information required for printing as part of the tree. For variates and ordered factors, the labels are generally formed as "*identifier*<*p*" and "*identifier*>*p*", where *p* is the value chosen to partition the data for the variate concerned. Alternatively, if you have defined an "extra" text for the variate (using the `EXTRA` parameter of the `VARIATE` command), this will be used instead. The labels are then "*extra-text* < *p*" and "*extra-text* > *p*". The style is similar for unordered factors, but here the labels involve the operators `.IN.` and `.NI.` instead of < and >.

Generally the construction will result in *over-fitting*, that is it will form a tree that keeps making splits beyond the point that can be justified statistically. The solution is to prune the tree to remove the uninformative sub-branches, and this can be performed using the `BPRUNE` procedure. It is best, if possible, to base the pruning on an independent set of data. The pruning uses the *accuracy* figures, which are stored with the tree. The `BRVALUES` procedure can be used to calculate new accuracy (and prediction) values, from another data set.

Finally, once the tree has been pruned, the value predicted for a new set of independent values can be obtained by supplying their values to the `BRPREDICT` procedure. This runs the values through the tree to see which terminal node they reach. The prediction is then provided by the value predicted at that node.

Options: `PRINT`, `Y`, `TREE`, `MSLIMIT`, `NSTOP`, `OWNBSELECT`.
Parameters: `X`, `ORDERED`.

Method

`BREGRESSION` calls procedure `BCONSTRUCT` to form the tree. This uses a special-purpose procedure `BSELECT`, which is customized specifically to select splits for use in regression trees and stored with `BREGRESSION`. You can use your own method of selection by providing your own `BSELECT` and setting option `OWNBSELECT=yes`. In the standard version of `BSELECT`, the `BASSESS` directive is used to assess the potential splits.

Action with `RESTRICT`

Any restrictions on the `Y` or `X` variates are removed.

Reference

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth, Monterey.

See also

Procedures: `BRDISPLAY`, `BRKEEP`, `BRPREDICT`, `BRVALUES`, `BRFOREST`, `BGRAPH`, `BPRUNE`, `BCLASSIFICATION`, `BCFOREST`.

Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

BRFDISPLAY

Displays information about a random regression forest (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (outofbagerror, youtofbagestimates, importance orderedimportance); default * i.e. none

Parameter

SAVE = *pointers*	Save structure from BRFOREST providing information about the random forest

Description

BRFDISPLAY displays information about a random regression forest, constructed by the BRFOREST procedure. The SAVE parameter can be set to a pointer, saved using the SAVE option of BRFOREST, containing the necessary information about the forest. Alternatively, if you do not set SAVE, information will be printed about the forest most recently constructed by BRFOREST.

The output is controlled by the PRINT option, with settings:

outofbagerror	out-of-bag error rate,
youtofbagestimates	out-of-bag predictions of the y-values,
importance	importance ratings of the X variates and factors, and
orderedimportance	importance ratings of the X variates and factors in decreasing order.

The default is PRINT=* i.e. no printing

Option: PRINT.
Parameter: SAVE.

See also

Procedures: BRFOREST, BRFPREDICT.
Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

BRFOREST

Constructs a random regression forest (R.W. Payne).

Options

`PRINT` = *string tokens*	Controls printed output (`outofbagerror`, `youtofbagestimates`, `importance`, `orderedimportance`, `monitoring`); default `outo`, `impo`
`Y` = *variate*	Response variate for the regression
`NTREES` = *scalar*	Number of trees in the forest; no default – must be specified
`NXTRY` = *scalar*	Number of `X` variables to select at random at each node from which to choose the `X` variable to use there; default is the square root of number of `X` variables
`NUNITSTRY` = *scalar*	Number of units of the X variables to select at random to use in the construction of each tree; default is two thirds of the number of units
`MSLIMIT` = *scalar*	Limit on the mean square of the observations at a node at which to stop making splits; default 0
`NSTOP` = *scalar*	Specifies the number of observations at a node at which to stop making splits; default 1
`SEED` = *scalar*	Seed for random numbers to select the `NXTRY` `X`-variables and `NUMITSTRY` units; default 0
`OWNBSELECT` = *string token*	Indicates whether or not your own version of the `BSELECT` procedure is to be used, as explained in the Method section (`yes`, `no`); default `no`
`OUTOFBAGERROR` = *string token*	Saves the "out-of-bag" error rate
`YOUTOFBAGESTIMATES` = *variate*	Saves the "out-of-bag" estimates of `Y`
`SAVE` = *pointer*	Saves details of the forest that has been constructed

Parameters

`X` = *factors* or *variates*	X-variables available for constructing the tree
`ORDERED` = *string tokens*	Whether factor levels are ordered (`yes`, `no`); default `no`
`IMPORTANCE` = *scalars*	Saves the importance of each x-variable

Description

A regression tree is a mechanism for predicting a response variable from a set of independent variables (see Chapter 8 of Breiman *et al*.). A random regression forest is a set of regression trees that are used collectively to form the prediction, by averaging the predictions from the individual trees (see e.g. Breiman 2001). The number of trees in the forest is specified by the `NTREES` option. Constructing a large forest can be time consuming, so it may be best to investigate first with a relatively small number of trees (e.g. 10).

The trees are constructed using data on a set of observations. Their values for the response variable are specified (in a variate) using the `Y` option, and their values for the independent variables are specified (in a list of variates or factors) using the `X` parameter. Factors may have either ordered or unordered levels, according to whether the corresponding value `ORDERED` parameter is set to `yes` or `no`. For example, a factor called `Dose` with levels 1, 1.5, 2 and 2.5 would usually be treated as having ordered levels, whereas levels labelled `'Morphine'`, `'Amidone'`, `'Phenadoxone'` and `'Pethidine'` of a factor called `Drug` would be regarded as unordered.

Each regression tree is formed using a random sample of the `X` variables in the data set, and

a bootstrap random sample of their units (i.e. sampled with replacement). The NXTRY option defines how many X variables to select, and the NUNITSTRY option defines how many units to take. The default for NXTRY is the square root of the number of variables, and the default for NUNITSTRY is two thirds of the number of units. The SEED option specifies a seed for the random numbers that are used to select the variables and to select the units. The default of zero continues an existing sequence of random numbers, if any of the random functions (GRSELECT etc) has already been used in the current Genstat run. Otherwise, a seed is chosen at random.

The construction process splits the observations into subsets. With an x-variate or a factor with ordered levels, the subsets are formed by taking the observations with values less than or greater than some split point *p*. For a factor with unordered levels, all possible ways of dividing its levels into two subsets are tried. The aim is to form subsets that have similar values for the response variate. The predicted value of the response variable for each node of the tree is the mean of its value for the subset of observations at that node. The *accuracy* of the node is the squared distance of the values of the response variate from their mean for the observations at the node, divided by the total number of observations. The potential splits at the node are assessed by their effect on the accuracy, that is the difference between the accuracy of the node and the sum of the accuracies of the two potential successor nodes. The node will become a terminal node if none of the splits provides any improvement in accuracy, or if the mean square of the observations at the node is less than or equal to a limit specified by the MSLIMIT option (default 0), or if the number of observations at the node is less than or equal to the number specified by the NSTOP option (default 1).

The resulting forest (and its associated information) can be saved using the SAVE option. This can then be used in the BRFDISPLAY procedure to produce further output, or in the BRFPREDICT procedure to predict the response for new values of the x-variables.

The OUTOFBAGERROR parameter can save the "out-of-bag" error rate. This is calculated using the individuals that were not involved in the construction of each tree. So, it gives an independent measure of the reliability of the forest. The idea is to put the x-values in each observation through all of the trees where it was not used, and predict its y-value by taking the average of the predictions from the individual trees. The out-of-bag error is the square root of the mean of the squared differences of the predictions from the values in the response variate. The YOUTOFBAGESTIMATES can save a variate containing the out-of-bag predictions, and the %VARIANCE option can save the percentage of the variance in the y-values that is accounted for by the forest. Note: the out-of-bag prediction will be missing for any observation that has been selected in all the random samples (i.e. that has been used to construct every tree).

The IMPORTANCE parameter can save a variate giving the "importance" of each X variate or factor in the forest, calculated as the total amount by which the variable increases the accuracy in the forest.

Printed output is controlled by the PRINT option, with settings:

outofbagerror	out-of-bag error rate,
youtofbagestimates	out-of-bag predictions of the y-values,
importance	importance ratings of the X variates and factors,
orderedimportance	importance ratings of the X variates and factors in decreasing order, and
monitoring	monitoring information during the construction process.

The default is PRINT=outofbagerror,importance.

Options: PRINT, Y, NTREES, NXTRY, NUNITSTRY, MSLIMIT, NSTOP, SEED, OWNBSELECT, OUTOFBAGERROR, YOUTOFBAGESTIMATES, SAVE.

Parameters: X, ORDERED, IMPORTANCE.

Method

BRFOREST calls procedure BCONSTRUCT to form the tree. This uses a special-purpose procedure BSELECT, which is customized specifically to select splits for use in regression trees. You can use your own method of selection by providing your own BSELECT and setting option OWNBSELECT=yes. In the standard version of BSELECT, the BASSESS directive is used to assess the potential splits.

Action with RESTRICT

Restrictions on the X or Y vectors are ignored.

References

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth, Monterey.

Breiman, L. (2001) Random forests. *Machine Learning*, **45**, 5-32.

See also

Procedures: BRFDISPLAY, BRFPREDICT, BREGRESSION.

Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

BRFPREDICT

Makes predictions using a random regression forest (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (prediction); default pred
PREDICTION = *variate*	Saves the prediction for the observations
SAVE = *pointer*	Save structure from BRFOREST providing information about the random forest

Parameters

X = *variates* or *factors*	Explanatory variables
VALUES = *scalars*, *variates* or *texts*	Values to use for the explanatory variables; if these are unset for any variable, its existing values are used

Description

BRFPREDICT makes predictions using a regression forest, as constructed by the BRFOREST procedure. The SAVE parameter can be set to a pointer, saved using the SAVE option of BRFOREST, containing the necessary information about the forest. Alternatively, if you do not set SAVE, BRFPREDICT will use the forest most recently constructed by BRFOREST.

The x-values for the predictions can be specified in the variates listed by the X parameter. These must have identical names (and levels) to those used originally to construct the tree. You can use the VALUES parameter to supply new values, if those stored in any of the variates or factors are unsuitable.

By default, BRFPREDICT prints the predictions, but you can set option PRINT=* to suppress this. The PREDICTION option allows you to save the predictions.

Options: PRINT, PREDICTION, SAVE.
Parameters: X, VALUES.

Method

BRFPREDICT takes the mean of predictions from the individual trees, made using BIDENTIFY.

Action with RESTRICT

Restrictions are ignored.

See also

Procedures: BRFOREST, BRFDISPLAY, BREGRESSION, BRPREDICT.
Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

BRKEEP

Saves information from a regression tree (R.W. Payne).

No options

Parameters

TREE = *trees*	Tree from which the information is to be saved
SUMMARY = *variates*	Saves summary information about each tree
XVARIABLES = *pointers*	Saves the identifiers of the x-variables in each tree

Description

BRKEEP saves information about a regression tree, constructed by the BREGRESSION procedure. The tree can be saved using the TREE option of BREGRESSION, and is specified for BRKEEP using its TREE parameter.

The SUMMARY parameter saves a variate containing summary information: number of nodes, number of terminal nodes, residual sum of squares, residual degrees of freedom, residual mean square and percentage variance accounted for (in that order).

The XVARIABLES parameter saves a pointer containing the identifiers of the x-variables in the tree.

Options: none.
Parameters: TREE, SUMMARY, XVARIABLES.

See also

Procedures: BREGRESSION, BRDISPLAY, BRPREDICT.
Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

BRPREDICT

Makes predictions using a regression tree (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (prediction, transcript); if PRINT is unset in an interactive run BRPREDICT will ask what you want to print, in a batch run the default is pred
TREE = *tree*	Specifies the tree
PREDICTIONS = *variate*	Saves the prediction for the observations
TERMINALNODES = *pointer*	Saves the numbers of the terminal nodes from which each prediction was obtained
MVINCLUDE = *string token*	Whether to provide predictions for units with missing or unavailable values of the x-variables (explanatory); default expl

Parameters

X = *variates* or *factors*	Explanatory variables
VALUES = *scalars*, *variates* or *texts*	Values to use for the explanatory variables; if these are unset for any variable, its existing values are used

Description

BRPREDICT makes predictions using a regression tree, as constructed by the BREGRESSION procedure. The tree can be saved from BREGRESSION (using the TREE option of BREGRESSION), and specified for BRPREDICT using its own TREE option. Alternatively, BRPREDICT will ask you for the identifier of the tree if you do not specify TREE when running interactively.

The x-values for the predictions can be specified in the variates or factors listed by the X parameter. These must have identical names (and levels) to those used originally to construct the tree. You can use the VALUES parameter to supply new values, if those stored in any of the variates or factors are unsuitable.

If you do not set X when running interactively, BRPREDICT will ask you to supply the relevant x-values in turn, as required by the tree. Otherwise, if an x-variable in the tree is not specified in the X parameter list, its values are assumed to be unavailable (i.e. missing).

By default, when the x-variable required at a node in the tree is unavailable or contains a missing value, BRPREDICT will follow all the branches from that node, and average the predictions that they generate. You can set option MVINCLUDE=*, if you would prefer the prediction to be missing.

The PRINT option controls printed output, with settings:

prediction	prints the predictions obtained using the tree;
transcript	prints the x-values supplied in response to questions in an interactive run.

If you do not set PRINT in an interactive run, BRPREDICT will ask what you would like to print. In batch, the default is to print the predictions.

You can save the predictions, in a variate, using the PREDICTIONS option. The TERMINALNODES option allows you to save a pointer, with an element for each prediction, containing the numbers of the terminal nodes reached in the tree to provide the predictions. This will be a scalar if the prediction was derived from a single node, or a variate if it involved more than one (because several branches have been taken, as the result of a missing x-value).

Options: PRINT, TREE, PREDICTIONS, TERMINALNODES.
Parameters: X, VALUES.

Method
BRPREDICT uses BIDENTIFY to find the terminal nodes of the tree that correspond to the values of the explanatory variables.

Action with RESTRICT
Restrictions are ignored.

See also
Procedures: BREGRESSION, BRKEEP, BRDISPLAY.
Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

BRVALUES

Forms values for nodes of a regression tree (R.W. Payne).

Options

Y = *variate*	Values of the response variate for the new data set
TREE = *tree*	Tree for which predictions and accuracy values are to be formed
REPLACE = *string token*	Whether to replace the values stored in the tree (yes, no); default no
PREDICTION = *pointer*	New predictions for the nodes of the tree
ACCURACY = *pointer*	New accuracy values for the nodes of the tree
NOBSERVATIONS = *pointer*	New numbers of observarions for the nodes of the tree

Parameter

X = *variates*	Values of the x-variates for the new data set

Description

When pruning a regression tree, it is best to use "accuracy" figures that are derived from a different set or sets of data from that which was used to construct the tree. BRVALUES allows these to be calculated, together with predictions for the nodes of the tree.

The TREE option specifies the tree for which the values are to be formed. The Y option specifies the values of the response variate for the observations in the new data set, and the X parameter defines their values for the x-variates as used to construct the tree. You can set option REPLACE=yes to use the new values to replace those already stored in the tree. Alternatively, you can use the PREDICTION parameter to save the predictions, in a pointer. This has an element for each node of the tree (and with the same suffix as that node) pointing to a scalar storing the prediction for the node. Similarly, the ACCURACY parameter saves the accuracies, and the NOBSERVATIONS parameter saves the numbers of observations at each node. You can use these later to replace the prediction and accuracy values in the original tree by

```
CALCULATE Tree[]['accuracy'] = ACCURACY[]
&         Tree[]['prediction'] = PREDICTION[]
&         Tree[]['nobservations'] = NOBSERVATIONS[]
```

Alternatively, you may want to combine them first with other estimates, for example to form bootstrapped estimates.

Options: Y, TREE, REPLACE, PREDICTION, ACCURACY, NOBSERVATIONS.
Parameter: X

Method

BRVALUES uses the standard Genstat tree functions to obtain the necessary information about the tree.

Action with RESTRICT

BRVALUES takes account of any restrictions on the Y or X variates.

See also

Procedures: BREGRESSION, BRDISPLAY, BRKEEP, BRPREDICT, BPRUNE.
Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

CABIPLOT

Plots results from correspondence analysis or multiple correspondence analysis (A.I. Glaser).

Options

DIMENSIONS = *scalars*	Two numbers specifying which axes of the ordinations to plot; default 1,2
PLOT = *string tokens*	Which scores to plot (rowscores, rowactive, rowpassive, colscores, colactive, colpassive); default rows, cols for correspondence analysis and cols for multiple correspondence analysis
ROWSCALING = *string token*	Scaling to use for row coordinates (principal, standard, mass, sqrtmass); default prin
COLSCALING = *string token*	Scaling to use for column coordinates (principal, standard, mass, sqrtmass); default prin
COLOURMETHOD = *string tokens*	Whether colour of symbol should show level of inertia of rows or columns (rowinertia, colinertia); default *
SIZEMETHOD = *string tokens*	Whether size of symbol should show row or column masses (rowmass, colmass); default *
FACCOLOURS = *text*, *variate* or *scalar*	Specifies a colour or colours for the factors in a multiple correspondence analysis; if this is unset, a different colour is selected automatically for every factor
WINDOW = *scalar*	Which graphical window to use; default 1
KEYWINDOW = *scalar*	Graphical window for the key
SAVE = *pointer*	Supplies results from a analysis by CORANALYSIS or MCORANALYSIS; default uses the most recent analysis

Parameters

TITLE = *texts*	Titles for the plot
LMROWVARIABLES = *string tokens*	How to label the row scores (identifiers, labels, none, numbers); default labe if LROWVARIABLES is set, otherwise iden
LMCOLVARIABLES = *string tokens*	How to label the column scores (identifiers, labels, none, numbers); default labe if LCOLVARIABLES is set, otherwise iden
LROWVARIABLES = *texts*	Labels for row variables
LCOLVARIABLES = *texts*	Labels for column variables

Description

CABIPLOT provides a graphical representation of results from CORANALYSIS or MCORANALYSIS. By default CABIPLOT plots both sets of scores (rowscores, colscores) for correspondence analysis or just columns scores for multiple correspondence analysis, but you can set option PLOT to select which ones are required. For correspondence analysis, you can also select settings that will plot only active or passive scores (see CORANALYSIS for further explanation).

The row scores are plotted as blue circles, while the column scores are plotted as red squares; active scores have filled symbols, but passive scores are not filled. With multiple correspondence analysis, the FACCOLOURS option can be used to define the colour to use for each factor, using either RGB values (in a variate or scalar) or the standard Genstat colour names (in a text); see PEN for more details. If insufficient colours are specified, CABIPLOT will recycle the list. So you

can set FACCOLOURS to a scalar or to a text with a single string if you want to use the same colour for all the factors. If FACCOLOURS is not set, CABIPLOT will select a different colour for each factor automatically.

The ROWSCALING and COLSCALING options are define the scaling to use for the row and columns coordinates respectively, with settings:

principal	plots principal coordinates (default),
standard	plots standard coordinates,
mass	plots standard coordinates multiplied by the row (or column) mass,
sqrtmass	plots standard coordinates multiplied by the square root of the row (or column) mass.

These are based on the row and column scores obtained from CORANALYSIS or MCORANALYSIS. Principal coordinates are scaled so that they have inertia equal to the square of the singular values, whereas the weighted sum-of-squares of the standard coordinates are equal to one. At least one of ROWSCALING or COLSCALING must be set to principal, which is the default for both options. These default settings produce a plot, which is not a biplot, but which is used very often to illustrate relationships between and amongst variables. The reasoning behind multiplying the standard coordinates by the corresponding mass or its square root is to "pull" the rarer categories to be closer to the origin; see Chapter 13 of Greenacre (2007).

The COLOURMETHOD option has settings rowinertia and colinertia that plot the row or coordinates scores, respectively, at a different level of shading; the coordinates with higher inertias are plotted with darker colours then those with low inertias. The shading is proportional to the square root of the inertia relative to the row or column with the highest inertia. Symbols representing passive points will appear completely transparent on the plot as they are perceived to have zero inertia.

The SIZEMETHOD option similarly has settings rowmass and colmass that plot the row and column coordinates, respectively, in sizes that depend on the row and column mass. The sizes of the symbols are proportional to the square root of the mass compared to the square root of the row or column with the highest mass, plus a constant to ensure all symbols are visible.

By default the first two dimensions are plotted, but you can specify other dimensions to be plotted using the DIMENSIONS option.

The data used in MCORANALYSIS may have many repeated values (particularly in survey data). To avoid replotting the same points in a large data set (i.e. with more than 500 units), only one point is plotted and the label refers to the first point in the data set. If the COLOURMETHOD or SIZEMETHOD options are set, these will use the mass and/or inertia of the labelled point.

The labels for the row and column scores can be set using the LMROWVARIABLES and LMCOLVARIABLES parameters, by selecting one of the following settings:

identifiers	uses the identifiers of the row or column scores,
labels	expects labels to be supplied (in a text) using the LROWVARIABLES or LCOLVARIABLES parameter,
none	gives no labels, and
numbers	uses the row or column numbers of the original matrix.

The default for both parameters is identifiers, unless LROWVARIABLES or LCOLVARIABLES is set, when the corresponding default becomes labels. Note that the texts supplied by LROWVARIABLES or LCOLVARIABLES must have the same number of values as number of the rows or columns in the original data matrix, even if active or passive points are being omitted from the plot. Similarly, if the setting numbers is chosen, these will refer to the corresponding row or column of the original matrix, ignoring any any active or passive rows or columns, or subsetting of rows or columns in CORANALYSIS.

By default CABIPLOT uses the results from the most recent analysis from by CORANALYSIS or MCORANALYSIS. However, you can display results from an earlier analysis by saving the

information about the analysis with the SAVE parameter of CORANALYSIS or MCORANALYSIS, and then using this as the setting of the SAVE option of CABIPLOT.

Options: DIMENSIONS, PLOT, ROWSCALING, COLSCALING, COLOURMETHOD, SIZEMETHOD, FACCOLOURS, WINDOW, KEYWINDOW, SAVE.
Parameters: TITLE, LMROWVARIABLES, LMCOLVARIABLES, LROWVARIABLES, LCOLVARIABLES.

Method
The plots are explained in Chapter 13 and 18 of Greenacre (2007).

Reference
Greenacre, M. (2007). *Correspondence Analysis in Practice, second edition*. Chapman & Hall, London.

See also
Procedures: CORANALYSIS, MCORANALYSIS.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

CANCORRELATION

Does canonical correlation analysis (P.G.N. Digby).

Option

PRINT = *string tokens*	Printed output from the analysis (correlations, pcoeff, qcoeff, pscores, qscores); default * i.e. no output

Parameters

PVARIATES = *pointers*	Pointer to P-set of variates to be analysed
QVARIATES = *pointers*	Pointer to Q-set of variates to be analysed
CORRELATIONS = *diagonal matrices*	Stores the canonical correlations from each analysis
PCOEFF = *matrices*	Stores the coefficients for the P-set of variates
QCOEFF = *matrices*	Stores the coefficients for the Q-set of variates
PSCORES = *matrices*	Stores the unit scores from the P-set of variates
QSCORES = *matrices*	Stores the unit scores from the Q-set of variates

Description

CANCORRELATION does canonical correlation analysis; see, for example, Mardia, Kent & Bibby (1979) or Digby & Kempton (1987).

The data for the procedure are two pointers specified by the PVARIATES and QVARIATES parameters; these must point directly to two sets of variates. The variates may have missing values, or be restricted: any units with any values missing will be excluded from the analysis; any restrictions on the variates must be consistent (the rules here are exactly as used by the FSSPM directive).

Printed output is controlled by the option PRINT with settings: correlations to print the canonical correlations (also expressed as percentages, and cumulative percentages, of their total); pcoeff to print the canonical correlation coefficients for the P-set of variates; qcoeff to print the canonical correlation coefficients for the Q-set of variates; pscores to print the canonical correlation scores for the units calculated from the P-set of variates; qscores to print the canonical correlation scores for the units calculated from the Q-set of variates.

Results from the analysis can be saved using the parameters CORRELATIONS, PCOEFF, QCOEFF, PSCORES and QSCORES. The structures specified for these parameters need not be declared in advance.

Option: PRINT.
Parameters: PVARIATES, QVARIATES, CORRELATIONS, PCOEFF, QCOEFF, PSCORES, QSCORES.

Method

The method used is as described in Digby & Kempton (1987). Spectral decompositions (LRL') of the SSPMs for the P-set and Q-set are used to form the inverse square root matrices, F and G (as $L R^{-½}$)). The singular value decomposition ($U S V'$) of ($F' C G$)is then formed, where C is the matrix of sums of products between the two sets of variates. The diagonal matrix S contains the canonical correlations; the canonical correlation coefficients for the two sets of variates are ($F U$) and ($G V$). The scores for the units from the two sets of variates are formed by subtracting the variate means and applying the matrices of coefficients as loadings.

References

Digby, P.G.N. & Kempton, R.A. (1987). *Multivariate Analysis of Ecological Communities*. Chapman & Hall, London.

Mardia, K.V., Kent, J.T. & Bibby, J.M. (1979). *Multivariate Analysis*. Academic Press, London.

See also
Procedures: CCA, RDA.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

CASSOCIATION

Calculates measures of association for circular data (S.J. Clark).

Options

PRINT = *string token*	What to print (tests); default test
NRANDOMIZATIONS = *scalar*	Number of randomizations to use in the randomization tests; default 999
ASCALE = *string token*	Units of the circular variables (degrees, radians); default degr

Parameters

Y = *variates*	Response variable
X = *variates*	Circular explanatory variable
YTYPE = *string tokens*	Type of response variable (circular, linear); default circ
SEED = *variates*	Variate of length two, firstly to supply a seed for the randomization tests and secondly to supply a seed to use for randomly-selecting sets of data points; default !(0,0)
STATISTICS = *variates*	Saves the test statistics

Description

CASSOCIATION calculates measures of association between a linear response variate and a circular explanatory variate (i.e. *linear-circular*) or between a circular response variate and a circular explanatory variate (i.e. *circular-circular*), as described in Fisher (1993, Chapter 6, Sections 6.1 - 6.3). The case of a circular response variate and a linear explanatory variable is not covered by CASSOCIATION; instead see procedure RCIRCULAR.

The data variates are supplied by the Y and X parameters. X should always be a circular variable. Y may be a linear or circular variable; its type is specified by the YTYPE parameter. So YTYPE=circular defines circular-circular data, and YTYPE=linear defines linear-circular data. Circular variables should represent vectorial data (i.e. directed lines). If they originally represent axial data (i.e. undirected lines), they should be transformed to vectorial data before using CASSOCIATION, by doubling and reducing their values modulo 360° i.e. by

```
CALCULATE X = MODULO(2*X; 360)
```

(see Fisher 1993, page xvii). By default, circular variables should be supplied as degrees, but you can supply radians instead by setting option ASCALE=radians.

Printed output is controlled by the PRINT option, with setting

tests	to print the results of the relevant tests (default).

The NRANDOMIZATIONS option specifies the number of randomizations to use with each of the randomization tests (see Method); the default is 999.

The SEED parameter can be set to a variate of length two, to supply seeds for the random numbers that may be used by CASSOCIATION with each y-variate. The first value provides a seed for the RANDOMIZE directive when calculating the randomization tests. The second value provides a seed for the CALCULATE directive when selecting random sets of points to calculate some of the statistics when there are too many data values to form all the sets (see the Method section for details). These both have a default setting of zero, which continues the existing sequence of random numbers if any have already been used in the current Genstat job; otherwise Genstat picks a seed at random. The seeds can be any positive integer, but only the last six digits of its integer part are used.

The test statistics can be saved using the STATISTICS parameter. For both linear-circular and circular-circular data the result will be a variate of length three containing either D_n, λ_n and R_n^2,

or Δ_n, Π_n and ρ_T, respectively (see Method).

Options: PRINT, NRANDOMIZATIONS, ASCALE.
Parameters: Y, X, YTYPE, SEED, STATISTICS.

Method

Full details of the terminology and methodology are given in Fisher (1993, Chapter 6, Sections 6.1 - 6.3). The various tests, test statistics and methods for assessing significance are outlined here. In the equations below, n represents the sample size.

Linear-circular association can be represented as a curve on the surface of a cylinder: the response variate is the height of the curve on the cylinder, and the explanatory variate is the angle around the cylinder. A curve that performs a sine wave around the cylinder is said to show *C-linear association*. The more general form, that has one minimum and one maximum around the cylinder (and that joins up at zero and 360 degrees) is said to show *C-association*. Three tests are provided for linear-circular data. The first tests for the presence of C-association using a test statistic D_n (Mardia 1976), which has a range [0,1] and is zero if there is no C-association. The value of D_n is assessed by calculating an associated statistic U_n. For $5 \le n \le 100$, upper 100α% critical values of U_n from Appendix A10 of Fisher (1993) are printed in the output (with linear interpolation where appropriate). For $n > 100$ the probability of U_n can be approximated by $\exp(-U_n^2/2)$. No probability values are available for $n < 5$.

The second test assesses the extent of C-association using a statistic λ_n; see Fisher & Lee (1981). This represents the probability that a randomly-selected sequence of four data points is *c-concordant*, i.e. whether they go up and down (or down and up) successively in their progress around the cylinder (see page 142 of Fisher 1993). When there is no C-association, λ_n=2/3. Larger values of λ_n indicate presence of a "C-monotone relationship", whilst smaller values represent an ordinary monotone relationship (as between two ordinary linear random variables). The exact statistic is calculated for samples of size $n \le 30$, by forming all possible ordered subsets of size four; otherwise it is estimated by taking 30000 randomly-selected subsets. For $n < 6$, no probability values are available. For $6 \le n \le 8$, the cumulative probability distribution of the test statistic λ_n from Appendix A11(a) of Fisher (1993) is printed in the output. For $9 \le n \le 20$, a randomization test is used to assess the significance. For $n > 20$, the statistic $\Lambda_n = n(\lambda_n - 2/3)$ is referred to tables of upper 100α% critical values from Appendix 11(c) of Fisher (1993).

The third test assess the extent of C-linear dependence using a test statistic R_n^2 (Mardia 1976; Liddell & Ord 1978), which represents the multiple correlation of X with (cos(Y), sin(Y)). The significance is assessed using a randomization test. The null hypothesis of no C-linear association is rejected if R_n^2 is large.

With circular-circular data the two variables are said to have *T-monotone association* if, whenever we choose three values from the response variate and arrange them in a clockwise order, the equivalent three values from the explanatory variate will be in either a clockwise order or an anti-clockwise order (i.e. the two sets of values will be met in the same order, one then two then three, going either clockwise or anti-clockwise). They are said to have a *T-linear association* if either

$$Y = X + \theta_0 \text{ (modulo } 360^\circ)$$

(representing complete positive association), or

$$Y = -X + \theta_0 \text{ (modulo } 360^\circ)$$

(representing complete negative association). Again, three tests are provided. The first is a test for T-monotone association based on quantifying the amount of T-monotone association directly, i.e. by estimating a statistic Δ_n which represents a circular correlation coefficient; see Fisher & Lee (1982). When Y and X are dependent, Δ takes the value -1 or 1, but $\Delta = 0$ does not imply independence, only that the association is not of T-monotone form. The estimation of Δ_n is based on calculation of T-concordancy/discordancy for all distinct subsets of three pairs of data values.

The null hypothesis that there is no T-monotone association is rejected if Δ_n differs significantly from zero. The exact statistic is calculated for samples of size $n \leq 50$, by forming all possible ordered subsets of size three; otherwise it is estimated by taking 20000 randomly-selected subsets. For $3 \leq n \leq 7$, the probability is calculated from the critical values of $n \times \Delta_n$ given in Appendix A12(a) of Fisher (1993). For $n > 7$, upper 100α% critical values of $n \times \Delta_n$ from Appendix A12(b) of Fisher (1993) are printed in the output (with linear interpolation where appropriate). For a one-sided (or two-sided) test with significance level α, the value $n \times \Delta_n$ of (or $|n \times \Delta_n|$) should be compared with the upper 100α% (or 100(α/2)%) critical values.

The second test for T-monotone association is based on circular ranks, with test statistic Π_n (again representing a correlation coefficient); see Fisher & Lee (1982, 1983). The null hypothesis of no T-monotone association (i.e. `Y` and `X` independent) is is rejected if Π_n differs significantly from zero. For $3 \leq n \leq 7$, the probability is calculated from the probability distribution of $(n-1) \times \Pi_n$ given in Appendix A13(a) of Fisher (1993). [Note that the penultimate value of x given there for $n = 7$ is assumed to be 0.21.] For $n \geq 8$, upper 100α% critical values of the distribution of $(n-1) \times \Pi_n$ from Appendix A13(b) of Fisher (1993) are printed in the output (with linear interpolation where appropriate). For a one-sided (or two-sided) test with significance level α, the value of $(n-1) \times \Pi_n$ (or $|(n-1) \times \Pi_n|$) should be compared with the upper 100α% (or 100(α/2)%) critical values.

The third test checks for T-linear association using the test statistic ρ_T (Fisher & Lee 1983, 1986) which has range [-1,1]. The null hypothesis of no T-linear association is rejected if $|\rho_T|$ is large. For $n < 25$ a randomization test is used. For $n \geq 25$ the test depends on the marginal distributions of `Y` and `X`. If either distribution has a mean resultant length zero, the null hypothesis is rejected if $|n \times \rho_T| > -\log(\alpha)$. Alternatively, if neither of the mean resultant lengths is equal to zero, a related statistic Z is used that has an approximate Normal distribution (see Fisher 1993, page 152). An approximate 95% Jackknife confidence interval is constructed for ρ_T.

Action with `RESTRICT`

`Y` and `X` may be restricted but must have compatible numbers of values.

References

Fisher, N.I. (1993). *Statistical Analysis of Circular Data*. Cambridge University Press, Cambridge, UK.

Fisher, N.I. & Lee, A.J. (1981). *Nonparametric measures of angular-linear association*. *Biometrika*, **68**, 629-36.

Fisher, N.I. & Lee, A.J. (1982). Nonparametric measures of angular-angular association. *Biometrika*, **69**, 315-21.

Fisher, N.I. & Lee, A.J. (1983). A correlation coefficient for circular data. *Biometrika*, **70**, 327-32.

Fisher, N.I. & Lee, A.J. (1986). Correlation coefficients for random variables on a unit sphere or hypersphere. *Biometrika*, **73**, 159-64.

Liddell, I.G. & Ord, J.K. (1978). Linear-circular correlation coefficients: some further results. *Biometrika*, **65**, 448-50.

Mardia, K.V. (1976). Linear-circular correlation coefficients and rhythmometry. *Biometrika*, **63**, 403-5.

See also

Procedures: `CCOMPARE`, `CDESCRIBE`, `DCIRCULAR`, `RCIRCULAR`, `WINDROSE`.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

CATRENDTEST

Calculates the Cochran-Armitage chi-square test for trend (A.I. Glaser).

Option

PRINT = *string token*	Output required (test); default test

Parameters

DATA = *tables*	Table containing observed data
TREND = *factors*	Dimension of the table representing the trend; can default if only one dimension of size greater than 2
CHISQUARE = *scalars*	Saves the chi-square for trend
PROBABILITY = *scalars*	Saves the probability value for trend
DEVCHISQUARE = *scalars*	Saves the chi-square for deviations from a linear trend
DEVDF = *scalars*	Saves the degrees of freedom for the chi-square for deviations
DEVPROBABILITY = *scalars*	Saves the probability value for the chi-square for deviations

Description

The CATRENDTEST procedure calculates the Cochran-Armitage chi-square test for trend. Categorical data can be collected and categorized by explanatory factors (such as dosage or treatment level), and any analysis will try to indicate relationships between the response (binary) factor and explanatory factors. The Cochran-Armitage chi-square test calculates a chi-square statistic on 1 degree of freedom for a linear trend in the responses. The data are represented by a ($2\times K$ or $K\times 2$) contingency table, where K represents the explanatory factor (known as the *trend*).

The DATA parameter supplies the data values in a two-way table. The TREND parameter can be set to a factor to indicate which dimension of the table represents the trend; if this is omitted CATRENDTEST assumes that the trend is in the dimension with more than 2 rows or columns (the other dimension must have exactly 2 rows or columns).

By default CATRENDTEST prints the results of tests for trend and for deviation from a trend (chi-square values, degrees of freedom and probabilities), but you can suppress these by setting option PRINT=*.

Parameters CHISQUARE, PROBABILITY, DEVCHISQUARE, DEVDF and DEVPROBABILITY allow the results to be saved (in scalars).

Option: PRINT.

Parameters: DATA, TREND, CHISQUARE, PROBABILITY, DEVCHISQUARE, DEVDF, DEVPROBABILITY.

Method

The method is described in Section 15.2 of Armitage, Berry & Matthews (1994).

Reference

Armitage, P., Berry, G. & Matthews, J.N.S. (1994). *Statistical Methods in Medical Research*. Blackwell Science, Oxford.

See also

Procedures: MCNEMAR, QCOCHRAN.

Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Regression analysis.

CCA

Performs canonical correspondence analysis (A.I. Glaser).

Options

PRINT = *string tokens*	Controls printed output (variance, loadings, roots, evalues, evectors, speciesscores, sitescores, fitsitescores, correlations, fitcorrelations); default vari, root
NROOTS = *scalar*	Number of eigenvalues and eigenvectors to include in output; default * takes all the non-zero eigenvalues
NORMALIZE = *string tokens*	Whether to normalize the Y, X and/or Z variates to have unit sums-of-squares before the analysis (x, y, z); default x, z
SCALING = *string tokens*	Whether to scale for species or site score (species, site); default spec
TOLERANCE = *scalar*	Tolerance for detecting non-zero eigenvalues; default 10^{-5}

Parameters

Y = *pointers*	Each pointer defines a set of response variates to be modelled
X = *pointers*	Explanatory variates or factors to use for each pointer of y-variates
Z = *pointers*	Conditioning variates or factors to remove ("partial out") before the analysis
LRV = *LRVs*	LRV structure from each analysis, storing the eigenvectors, eigenvalues and total variance
SPECIESSCORES = *matrices*	Save the "species scores" from each analysis
SITESCORES = *matrices*	Save the "site scores" from each analysis
FITSITESCORES = *matrices*	Save the fitted "site scores" from each analysis
CORRELATIONS = *matrices*	Saves the correlations between the site scores and the x-variates
FITCORRELATIONS = *matrices*	Saves the correlations between the fitted site scores and the x-variates
SAVE = *pointers*	Save structure which provides information for use in CRBIPLOT and CRTRIPLOT

Description

CCA performs canonical correspondence analysis and partial canonical correspondence analysis.

Canonical correspondence analysis is the canonical form of correspondence analysis. It is similar to redundancy analysis (see RDA). However, in CCA, we apply weighted multiple regression to a transformed data matrix with the fitted values subjected to correspondence analysis.

The Y parameter specifies the response data as a pointer to a set of y-variates. Each variate contains observations of numbers of a particular species at a set of sites (the same sites and in the same order for each species). The explanatory variables, which may be either variates or factors, are specified in a pointer by the X parameter. Similarly, the Z parameter can be used to specify conditioning variables, which again may be either variates or factors. When a pointer of z-variables is supplied, CCA performs a partial canonical correspondence analysis, in which the effects of the z-variables are removed prior to the canonical correspondence analysis. This can be useful when the effects of the elements of Z on Y are well known, or if we wish to isolate the

effect of an single explanatory variable (in which case we would place all but one of the explanatory variables in Z). When all elements of a variable are equal to zero, CCA removes the variable.

The PRINT option controls printed output, with settings:

roots	the eigenvalues of the fitted values;
evalues	synonym of roots;
loadings	the eigenvectors associated with each eigenvalue, also known as the "species scores";
evectors	synonym of loadings;
speciesscores	the "species scores" from the analysis (synonym of loadings and evectors);
variance	the fraction of the variance of the y-variates associated with each eigenvalue;
sitescores	the "site scores" of the y-variates (i.e. the ordination of the units in the y-variate space);
fitsitescores	the fitted "site scores" of the fitted values of the y-variates (i.e. the ordination of the units in the y-variate space);
correlations	the correlation between the site scores and the x-variables;
fitcorrelations	the correlation between the fitted site scores and the x-variables.

By default PRINT=roots,variance. The LRV, SPECIESSCORES, SITESCORES, FITSITESCORES, CORRELATIONS and FITCORRELATIONS parameters allow this information to be saved.

The NROOTS option specifies the number of eigenvalues and eigenvectors to include in the output. By default all the non-zero eigenvalues are included. The NORMALIZE option controls whether to normalize the Y variates, or X or Z variables to have unit sums-of-squares before the analysis. The default is to normalize the x and z-variables but not the y-variates. (Note: normalization of only the x's and z's does not affect the variances accounted for in the y-variates.)

The SCALING option controls which scores are scaled by CCA: either the species scores or the site scores. The scaling is done by multiplying them by their corresponding eigenvalues. Choosing 'site' is equivalent to Scaling type 1 in Legendre & Legendre (1998), whilst 'species' is equivalent to their Scaling type 2.

The TOLERANCE option specifies a threshold for the detection of non-zero eigenvalues (default 10^{-5}). An eigenvalue is taken to be non-zero if is it greater than TOLERANCE.

The SAVE parameter allows you to save a pointer containing full details of the analysis. This can then be used to generate plots using the CRBIPLOT or CRTRIPLOT procedures. The most recent save structure is kept automatically inside Genstat to use as a default for the SAVE options of CRBIPLOT and CRTRIPLOT. So, you need save the pointer explicitly only if you want to display output from more than one analysis at a time.

Options: PRINT, NROOTS, NORMALIZE, SCALING, TOLERANCE.
Parameters: Y, X, Z, LRV, SPECIESSCORES, SITESCORES, FITSITESCORES, CORRELATIONS, FITCORRELATIONS, SAVE.

Method

CCA and partial CCA are explained in Sections 11.2 and 11.3 of Legendre & Legendre (1998). In Genstat the multiple regression is carried out using the QR decomposition (see QRD).

Action with RESTRICT

If any of the variate or factors in the Y, X or Z pointers are restricted, only the defined subset of the units will be used in the analysis.

Reference

Legendre, P. & Legendre, L. (1998). *Numerical Ecology, Second English Edition*. Elsevier, Amsterdam.

See also

Procedures: CRBIPLOT, CRTRIPLOT, CANCORRELATION, PLS, RDA.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

CCOMPARE

Tests whether samples from circular distributions have a common mean direction or have identical distributions (S.J. Clark).

Options

PRINT = *string token*	What to print (tests); default test
TEST = *string token*	Which tests to perform (compare, identical); default comp, iden
ASCALE = *string token*	Units of the circular variables (degrees, radians); default degr
STATISTICS = *variate*	Saves the test statistics
COMMON = *scalar*	Saves the common mean direction
LOWER = *scalar*	Saves the lower 95% confidence limit for common mean
UPPER = *scalar*	Saves the upper 95% confidence limit for common mean

Parameter

DATA = *variates*	Circular response variables to be compared

Description

CCOMPARE implements two nonparametric tests for comparing samples from circular distributions, as described in Fisher (1993, Chapter 5, Sections 5.3.4 - 5.3.6). These are selected using the TEST option, with the following settings.

compare	tests whether the samples have a common mean direction, and estimates the common direction if the directions are not significantly different. There must be 25 or more observations in each sample. The test assumes, without checking, that each distribution is unimodal. (Rayleigh's test of uniformity against a unimodel alternative is available in procedure CDESCRIBE).
identical	tests whether the samples come from identical distributions. Each sample must have ten or more observations.

By default, TEST=compare,identical.

Printed output is controlled by the PRINT option, with setting

tests	to print the results of the requested tests (default).

The sample observations are supplied, each in a separate variate, using the DATA parameter. Usually they all supply angles measured in degrees, but you can set option ASCALE=radians to supply them all in radians instead.

The test statistics can be saved, in a variate, using the STATISTICS option. The COMMON option can save the estimated common direction, and the LOWER and UPPER options can save its lower and upper 95% confidence limits (all in scalars).

Options: PRINT, TEST, ASCALE, STATISTICS, COMMON, LOWER, UPPER.
Parameter: DATA.

Method

Full details of the terminology and methodology are given in Fisher (1993, Chapter 5, Sections 5.3.4 - 5.3.6). The two tests are outlined only briefly here.

The test for a common mean direction of two or more unimodal distributions is described in Fisher (1993, Section 5.3.5(b)). Only a large-sample test is implemented, where all samples have 25 or more observations. The method that is used depends on the comparability of the circular

dispersions. If the ratio of the maximum to minimum dispersion is less than or equal to four, "Method P" is used (Fisher 1993, page 116). Otherwise "Method M" is used (Fisher 1993, page 117). In either case the resulting test statistic is denoted by Y_r, and the null hypothesis of a common mean direction is assessed using a chi-square distribution with degrees of freedom equal to the number of samples minus one. If there is no evidence that the mean directions are different, the common mean direction is estimated, with a 95% confidence interval, using the weighting scheme appropriate to the method (P or M).

The test for whether two or more distributions are identical is described in Fisher (1993, Section 5.3.6). The test requires all the samples to have ten or more observations, but it does not assume unimodality. The test statistic, W_r, is based on the circular ranks of the data values, and can be assessed using a chi-square distribution with degrees of freedom equal to the number of samples minus two. If there is no evidence that the distributions are different, you can use procedure CDESCRIBE to estimate the common direction (after first combining the samples into a single variate, for example using the APPEND procedure).

Action with RESTRICT

Any of the DATA variates may be restricted.

Reference

Fisher, N.I. (1993). *Statistical Analysis of Circular Data*. Cambridge University Press, Cambridge, UK.

See also

Procedures: CASSOCIATION, CDESCRIBE, DCIRCULAR, RCIRCULAR, WINDROSE.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

CDESCRIBE

Calculates summary statistics and tests of circular data (P.W. Goedhart & R.W. Payne).

Options

PRINT = *string tokens*	What to print (summary, fittedvalues); default summ
SEGMENT = *scalar*	Width of sectors (in degrees) into which to group an ANGLES variate for calculation of the test of randomness and the chi-square goodness of fit statistic for the von Mises distribution; default 20
MSEGMENT = *scalar*	Defines the centre (in degrees) of the sectors; default 0
DIRECTION = *scalar*	Direction (in degrees) of the unimodal alternative distribution for the Rayleigh test; default * i.e. not known

Parameters

ANGLES = *factors* or *variates*	Directional observations (in degrees)
RESULTS = *variates*	Saves the summary statistics
VONMISESCOUNTS = *pointers*	Saves structures relevant for calculation of the chi-square goodness of fit statistic for the von Mises distribution

Description

CDESCRIBE summarizes data values that consist of directional observations recorded as angles between 0 and 360 degrees. These are supplied using the ANGLES parameter, in either a variate or a factor. The procedure mainly uses the methods presented in the book by Fisher (1993). The various statistics are cross-referenced below with the relevant page numbers.

CDESCRIBE prints the following summary statistics: number of observations, mean direction (page 31), circular standard deviation (page 32), mean resultant length (page 32), skewness (page 34) and estimate of the parameter Kappa (which provides the concentration parameter of the von Mises distribution for circular data; pages 39 and 88). If the angles are supplied in a factor, a grouping correction is applied to the mean resultant length and to the skewness (page 35).

Two tests of uniformity are presented. The null hypothesis for both of these is that the observations come from a uniform distribution around the circle. The first is a test of randomness against any alternative model. The test is based on counts of the number of observations in a set of angular sectors of equal size (page 67). If ANGLES is set to a variate, the width of the sectors is defined by the SEGMENT option (in degrees), with centres defined by the MSEGMENT option. The sectors are centred at MSEGMENT, MSEGMENT+SEGMENT, MSEGMENT+2*SEGMENT, and so on. The default values for SEGMENT and MSEGMENT are 20 and 0 respectively. If ANGLES is set to a factor with equidistant levels, it is assumed that the levels define the centres of the segments and that the limits of the sectors are at the midpoints between each pair of factor levels. If ANGLES is set to factor with non-equidistant levels, the SEGMENT and MSEGMENT options are used to define the angular sectors.

The second is Rayleigh's test of uniformity against a unimodel alternative. The test is based on the mean resultant length and has two forms which differ according to whether or not the mean direction of the alternative distribution is known (pages 69 and 70). The direction, if known, is specified using the DIRECTION option.

Finally a goodness of fit test is calculated to assess whether the observations follow a von Mises distribution. This is a chi-square test, which compares the observed distribution with the expected distribution from a von Mises distribution with mean direction and concentration parameter (kappa) taking the values estimated from the observations. The observed and expected values are calculated for grouped directional data defined by the (M)SEGMENT options for a

variate or by the factor levels if ANGLES is set to a factor.

The PRINT options controls whether the summary statistics are printed and whether a table of observed and expected counts for the fit of the von Mises distribution is printed. The summary statistics can be saved by means of the RESULTS parameter. The VONMISESCOUNTS parameter saves the grouped directional data used for calculation of the chi-square goodness of fit test and tables of observed and expected counts. Note that when ANGLES is set to factor, the saved grouped directional data is identical to ANGLES.

Options: PRINT, SEGMENT, MSEGMENT, DIRECTION.
Parameters: ANGLES, RESULTS, VONMISESCOUNTS.

Method

CDESCRIBE uses methods described by Fisher (1993). A private version (_SPECIALFUNCTION) of the Biometris procedure SPECIALFUNCTION is used to calculate modified Bessel functions and related functions.

Action with RESTRICT

If ANGLES is restricted, only the unrestricted units are analysed.

Reference

Fisher, N.I. (1993). *Statistical Analysis of Circular Data*. Cambridge University Press, Cambridge.

See also

Procedures: CASSOCIATION, CCOMPARE, DCIRCULAR, RCIRCULAR, WINDROSE.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

CDNAUGMENTEDDESIGN

Constructs an augmented block design, using CycDesigN if the controls are in an incomplete-block design (R.W. Payne).

Options

PRINT = *strings*	Controls printed output (design, controldesign, factors, monitor); default * i.e. none
LEVELS = *scalar* or *variate*	Levels for the unreplicated treatments
LEVCONTROLS = *scalar* or *variate*	Levels for the control treatments
NROWS = *scalar*	Number of rows
NCOLUMNS = *scalar*	Number of columns
NRBLOCKS = *scalar*	Number of rows in each block
NCBLOCKS = *scalar*	Number of columns in each block
NCONTROLSPERBLOCK = *scalar*	Number of control treatments in each block
TREATMENTS = *factor*	Treatment factor
ROWS = *factor*	Row factor
COLUMNS = *factor*	Column factor
BLOCKS = *factor*	Block factor
ROWBLOCKS = *factor*	Row block factor
COLBLOCKS = *factor*	Column block factor
NTIMES = *scalar*	Number of times to try allocations of controls within blocks
SEED = *scalar* or *variate*	Scalar or variate with three values specifying seeds for the random numbers used by CycDesigN to search for the control design, for the allocation of controls within blocks, and for the allocation of the unreplicated treatments – if a scalar is specified the same seed is used for all purposes; default 0 i.e. set automatically
SPREADSHEET = *string*	Whether to put the design factors into a spreadsheet (design); default *
TIMELIMIT = *scalar*	Time in minutes for CycDesigN to search; default 1

No parameters

Description

An augmented design starts with a basic design containing control treatments. This is then augmented by adding extra plots for the unreplicated test treatments. The basic design in CDNAUGMENTEDDESIGN is a block design, where the blocks can be in either a one- or a two-dimensional layout. The augmenting stage expands the blocks with extra plots to contain the unreplicated treatments.

The levels of the treatment factor to be used for the unreplicated treatments are specified by the LEVELS option. This can be a scalar, to specify levels 1, 2 etc., or a variate specifying the actual levels to use. The levels for the control treatments are specified similarly by the LEVCONTROLS option. There must be more than two controls, and their levels must be distinct from those for the unreplicated treatments.

The NROWS option can define the number of rows in the design, and the ROWS option can supply a factor to save the levels generated for the row factor. You can omit NROWS if ROWS is set to a factor that has already been defined with the correct number of levels. Similarly, the NCOLUMNS option can define the number of columns, and the COLUMNS option can supply a factor to save the levels generated for the column factor.

The NRBLOCKS and NCBLOCKS options specifies the number of rows and columns,

respectively, to be used for each block. The numbers of rows and columns in the design must be exact multiples of NRBLOCKS and NCBLOCKS. You can specify a one-dimensional layout for the blocks by setting either NRBLOCKS equal to the number of rows in the design, or NCBLOCKS equal to the number of columns in the design. Otherwise the blocks are in a two-dimensional (row-by-column) layout. The BLOCKS option can save a factor containing the levels generated for the blocks. Also, the ROWBLOCKS option can save a factor containing the row location of each block in the two-dimensional (row-by-column) layout, the and the COLBLOCKS option can save a factor containing its column location.

The NCONTROLSPERBLOCK option specifies the number of plots for controls in each block. If this is less than the number of controls, CycDesigN is used to find an efficient incomplete-block design for the controls. (Otherwise the basic design is a randomized complete block design and CycDesigN is not needed.) The TIMELIMIT option specifies the time in minutes for CycDesigN to search.

CDNAUGMENTEDDESIGN then tries several randomizations for the controls within the blocks, and takes the one that has the most uniform allocation of the control treatments over the rows and columns within the blocks. The NTIMES option specifies the number of randomizations to try (default 1000).

The SEED option allows you to supply seeds for the random numbers to be used for the random numbers used by CycDesigN to search for the control design, for the allocation of controls within blocks, and for the allocation of the unreplicated treatments . You can specify a variate with three values to supply a different seed for each purpose, or a scalar to use the same one for both. If a zero value is specified, the corresponding seed is set automatically. The default is the scalar zero.

Printed output is controlled by the PRINT option, with settings:

design	to print the design,
controldesign	to print the design showing just the control treatments,
factors	to print the factor values, and
monitor	to print a report by CycDesigN on the design and monitor the randomizations of the control design.

By default nothing is printed.

You can set option SPREADSHEET=design to put the design factors into a Genstat spreadsheet.

Options: PRINT, LEVELS, LEVCONTROLS, NROWS, NCOLUMNS, NRBLOCKS, NCBLOCKS, NCONTROLSPERBLOCK, TREATMENTS, ROWS, COLUMNS, BLOCKS, ROWBLOCKS, COLBLOCKS, NTIMES, SEED, SPREADSHEET, TIMELIMIT.
Parameters: none.

Method

The batch CycDesigN program is called using the SUSPEND directive.

See also

Procedures: AFAUGMENTED, CDNPREP, CDNBLOCKDESIGN, CDNROWCOLUMNDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

CDNBLOCKDESIGN

Constructs a block design using CycDesigN (R.W. Payne).

Options

PRINT = *strings*	Controls printed output (design, report, factors); default * i.e. none
LEVELS = *scalar* or *variate*	Numbers of levels of the treatment factors; if unset, takes the numbers of levels declared for the factors in the TREATMENTSTRUCTURE model
NREPLICATES = *scalar*	Number of replicates
NBLOCKS = *scalar*	Number of blocks
NUNITS = *scalar*	Number of units per block
NGROUPS = *variate*	Group sizes for a two-factor nested treatment structure
TREATMENTFACTORS = *factors*	Up to four factors to use in the treatment model: one factor for a one-way treatment model, two factors for a nested structure when NGROUPS is set, or two to four factors for a factorial treatment structure when NGROUPS is not set
REPLICATES = *factor*	Replicate factor
BLOCKS = *factor*	Block factor
UNITS = *factor*	Unit-within-block factor
RESOLVABLE = *string*	Whether the design is resolvable (yes, no); default no
ALPHADESIGN = *string*	Whether an alpha design is constructed for a resolvable design (yes, no); default no
CYCLIC = *string*	Whether a cyclic design is constructed for a non-resolvable design (yes, no); default no
NBLATIN = *scalar*	Number of contiguous blocks to latinize; default 0 i.e. not latinized
REPLATINGROUPS = *variate*	Sizes of groups defining the positions of the replicates when constructing latinized designs; default * i.e. no groupings
SPATIALMODEL = *string*	Spatial model to use with a single-treatment-factor resolvable design (integer, linearvariance, seconddifference, ev); default * i.e. none
EVDECAY = *scalar*	Decay parameter to use when SPATIALMODEL=ev; default 0.5
WEIGHTS = *variate*	Variate with two values specifying weightings for the main effects and for the interactions in factorial treatment structures; default !(1, 0.25)
SEED = *scalar* or *variate*	Scalar or variate with two values specifying seeds for the random numbers used by CycDesigN to search for the best design and to randomize it – if a scalar is specified the same seed is used for both purposes; default 0 i.e. set automatically
SPREADSHEET = *string*	Whether to put the design factors into a spreadsheet (design); default *
TIMELIMITS = *scalar* or *variate*	A scalar or a variate containing up to three numbers defining the time in minutes to spend on the first phase, the second phase and the spatial phase of the search (if the 2nd or 3rd numbers are omitted they default to the maximum of those specified); default 1

NRANDOMIZATIONS = *scalar*	Number of randomizations to generate from the best design; default 1
TRIALS = *factor*	Trials factor

No parameters

Description

CycDesigN is a package for the computer generation of experimental designs, which constructs optimal or near-optimal block and row-column designs; see the book *Cyclic and Computer Generated Designs* by John & Williams (1995). CycDesigN can also operate as a batch program, that can be called from within Genstat. This program is distributed with Genstat, and there are procedures to call the program, read its output back into Genstat, and form the relevant design factors. There are also Genstat add-in and resource files to define user menus, which can be downloaded from the VSNi website. However, before CycDesigN can be used, a license must be obtained; see vsni.co.uk/software/cycdesign for details.

This procedure, CDNBLOCKDESIGN, uses the CycDesigN algorithms to form a block design. The treatment factors, whose values are to be formed, are specified by the TREATMENTFACTORS option. This can be set to a single factor if you want a one-way treatment structure. Alternatively, if option NGROUPS is set, you can supply two factors to define a nested model

```
factor_1 / factor_2
```

The group sizes (i.e. the number of levels of the second factor within each level of the first factor) are supplied by NGROUPS, in a variate. Otherwise, if option NGROUPS is not set, you can supply from two to four factors, to define a factorial model.

The LEVELS option can be used to define the numbers of levels of the factors, as a scalar if there is only one factor, or as a variate if there are several. The levels specified in the variate are assumed to be in the same order as the order in which the factors occur in the TREATMENTFACTORS list. LEVELS can be omitted if the factors have already been declared with the right numbers of levels. Alternatively, if you want only a single treatment factor, and do not want to save its generated levels, you can specify its number of levels using LEVELS, and leave TREATMENTFACTORS unset.

The RESOLVABLE option controls whether or not the design is *resolvable* i.e. whether the blocks can be partitioned into sets, each of which contains a single replicate of each treatment combination. Suppose that there are v treatment combinations, b blocks, k units per block and r replicates of each treatment combination. Then in a non-resolvable design these numbers must satisfy the condition

$$v \times r = b \times k.$$

In a resolvable design some blocks may have only $k-1$ units, and following condition must be satisfied

$$(b - 1) \times k < v \leq b \times k.$$

The NBLOCKS option can define the number of blocks, and the BLOCKS option can supply a factor to save the levels generated for the block factor. You can omit NBLOCKS if BLOCKS is set to a factor that has already been defined with the correct number of levels. Similarly, the NUNITS option can define the number of units within each block, and the UNITS option can supply a factor to save the levels generated for the unit-within-block factor. Finally, the NREPLICATES option can define the number of replicates of each treatment combination, and the REPLICATES option can supply a factor to save the levels generated for a replication factor.

Either NREPLICATES or REPLICATES must be specified if the design is resolvable. Apart from this constraint, any one of the pairs of options NBLOCKS and BLOCKS, or NUNITS and UNITS, or NREPLICATES and REPLICATES can be left completely unset. (CDNBLOCKDESIGN will then deduce the number of blocks, units per block or replicates, as required, from the equations above.)

The ALPHADESIGN option controls whether or not an alpha design is constructed for a resolvable design, and the CYCLIC option controls whether or not a cyclic design is constructed for a non-resolvable design.

Latinized designs are useful when the replicates of a resolvable design are set out next, or contiguous, to each other. The blocks of each replicate then form long blocks running down the replicates. You may than want to set option NBLATIN=1 to request a *latinized* design in which the replication of each treatment combination is equalized as far as possible within in each long block. (So, if $v \geq r \times k$, each treatment combination should occur no more than once.) Alternatively, setting NBLATIN to *n*, say, aims to equalize the occurrence of the treatment combinations within each set of *n* contiguous blocks. If NBLATIN is not set, the design is not latinized. By default, the replicates in a latinized design are assumed to be in a single row, side by side. The REPLATINGROUPS option allows you to to define an alternative layout. For example, setting REPLATINGROUPS=!(1,2) when you have three replicates, defines two columns of replicates, the first with one replicate (replicate 1), and the second with two replicates (replicate 2 alongside replicate 1, and replicate 3 below replicate 2).

The SPATIALMODEL option allows you to request that the construction of a resolvable design should take account of the separation of different treatments in blocks. The principle is that plots close together are assumed to be correlated more than plots further apart; a spatial model attempts to model this correlation decay. The criterion used to generate spatial designs is the neighbour efficiency factor of Williams (1985), which has been extended to two-dimensional blocking structures by Williams, John & Whitaker (2005) and to cater for different decay functions. The available settings are

integer	integer,
linearvariance	linear variance,
seconddifference	modified second difference, and
ev	modified exponential variance.

The weights used with the first three settings are described by Williams (1985). The fourth setting, ev, is appropriate for a model specifying an autoregressive variance matrix. Its decay parameter is specified by the EVDECAY option; default 0.5. However, the spatial designs generated by CycDesigN are usually quite robust to the choice of weight function. Spatial models cannot be used with latinized designs.

The WEIGHTS option specifies a variate with two values to define how to weight the efficiencies of the terms when there is a factorial treatment structure. The first value defines a weight for the main effects (default 1), and the second defines a weight for the interactions (default 0.25) These defaults are the same as those used in the stand-alone CycDesigN system.

The SEED option allows you to supply seeds for the random numbers to be used within CycDesigN to search for the best design and to randomize it. You can specify a variate with two values to supply a different seed for each purpose, or a scalar to use the same one for both. If a zero value is specified, the corresponding seed is set automatically. The default is the scalar zero.

By default only one randomization is done with the best design. However, you can use the NRANDOMIZATION option to provide several randomizations of that design, for use in different trials. The TRIALS option can save a factor to identify the trials.

Printed output is controlled by the PRINT option, with settings:

design	to print the design,
report	to print a report by CycDesigN on the design, and
factors	to print the factor values.

You can set option SPREADSHEET=design to put the design factors into a Genstat spreadsheet.

The TIMELIMITS option can be set to a scalar or a variate containing up to three numbers to define the time in minutes to spend on the first phase, the second phase and the spatial phase of the design search. If the second or third numbers are omitted, they default to the maximum of

those specified. The default is 1.

Options: PRINT, LEVELS, NREPLICATES, NBLOCKS, NUNITS, NGROUPS, TREATMENTFACTORS, REPLICATES, BLOCKS, UNITS, RESOLVABLE, ALPHADESIGN, CYCLIC, NBLATIN, REPLATINGROUPS, SPATIALMODEL, EVDECAY, WEIGHTS, SEED, SPREADSHEET, TIMELIMITS, NRANDOMIZATIONS, TRIALS.
Parameters: none.

Method
The batch CycDesigN program is called using the SUSPEND directive.

References
John, J.A. & Williams, E.R. (1995). *Cyclic and Computer Generated Designs*. London: Chapman and Hall.
Williams, E.R. (1985). A criterion for the construction of optimal neighbour designs. *J.R. Statist. Soc. B*, **47**, 487-497.
Williams, E.R., John, J.A. & Whitaker, D. (2005). Construction of resolvable spatial row-column designs. *Biometrics*, **62**, 103-108.

See also
Procedures: CDNAUGMENTEDDESIGN, CDNPREP, CDNROWCOLUMNDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

CDNPREP

Constructs a multi-location partially-replicated design using CycDesigN (R.W. Payne).

Options

PRINT = *strings*	Controls printed output (design, report, factors, blocknumbers); default * i.e. none
LEVELS = *scalar*	Numbers of levels of the treatment factor; if unset, takes the numbers of levels declared for the factor specified by the TREATMENTS option
NLOCATIONS = *scalar*	Number of locations
NBLOCKS = *scalar*	Number of blocks at each location
NUNITSPERLOCATION = *scalar*	Number of units at each location
NREPLICATEDPERBLOCK = *scalar*	Number of treatments in each block that are replicated at the location containing the block
TREATMENTS = *factor*	Treatment factor
LOCATIONS = *factor*	Locations factor
BLOCKS = *factor*	Block factor
UNITS = *factor*	Unit-within-block factor
SEED = *scalar* or *variate*	Scalar or variate with two values specifying seeds for the random numbers used by CycDesigN to search for the best design and to randomize it – if a scalar is specified the same seed is used for both purposes; default 0 i.e. set automatically
SPREADSHEET = *string*	Whether to put the design factors into a spreadsheet (design); default *
TIMELIMIT = *scalar*	Time in minutes to search; default 1

No parameters

Description

CycDesigN is a package for the computer generation of experimental designs, which constructs optimal or near-optimal block and row-column designs; see the book *Cyclic and Computer Generated Designs* by John & Williams (1995). CycDesigN can also operate as a batch program, that can be called from within Genstat. This program is distributed with Genstat, and there are procedures to call the program, read its output back into Genstat, and form the relevant design factors. There are also Genstat add-in and resource files to define user menus, which can be downloaded from the VSNi website. However, before CycDesigN can be used, a license must be obtained; see vsni.co.uk/software/cycdesign for details.

This procedure, CDNPREP, uses the CycDesigN algorithms to form a partially-replicated block design. The assumption in CycDesigN is that the experiment will contain incomplete-block designs conducted at several locations and that, at each location, some treatments will occur twice, others may occur only once, and others may not occur at all. However, the treatments are all replicated the same number of times over the whole design. So there is the constraint that the total number of units, or plots, in the design must be a multiple of the number of treatments. Also, the number of units at each location must be greater than the number of treatments, and less than twice the number of treatments.

The LEVELS option can be set to a scalar to define the number of treatments, and the TREATMENTS option can save a factor containing the generated values. LEVELS can be omitted if the TREATMENTS factor has already been declared with the right numbers of levels. Alternatively, if you only want to print the design and do not want to save the values, you can specify the number of levels using LEVELS, and leave TREATMENTS unset. Similarly, the

NLOCATIONS option can define the number of locations, and the LOCATIONS option can supply a factor to save the values generated for the locations factor. You can omit NLOCATIONS if LOCATIONS is set to a factor that has already been defined with the correct number of levels.

The number of units, or plots, at each location must be specified by the NUNITSPERLOCATION option, and must satisfy the constraints mentioned above. CycDesigN also needs to know the number of blocks at each location, and the number of treatments in each block that will be amongst those that are replicated (i.e. occur twice) at each location. These can be specified by the NBLOCKS and NREPLICATEDPERBLOCK options, respectively. However, designs are available for only limited combinations of values, and CDNPREP will give a fault diagnostic if you specify values that are not included in the feasible combinations. You can set option PRINT=blocknumbers to print the possibilities, and CDNPREP will then stop unless NBLOCKS and NREPLICATEDPERBLOCK are both set. Alternatively, if you are running Genstat interactively, CDNPREP will use the QUESTION procedure to prompt you to choose values from those that are feasible. Finally, if you are running Genstat in batch, CDNPREP will take the median number of feasible blocks and the corresponding median number of replicated treatments per block. Smaller values for NREPLICATEDPERBLOCK allow more of the treatments to be represented at each location, while larger values provide more residual degrees of freedom.

The BLOCKS option can supply a factor to save the values generated for the block factor, and the UNITS option can supply a factor to save the values generated for the unit-within-block factor (which identifies the units within each block).

Printed output is controlled by the PRINT option, with settings:

design	to print the design,
report	to print a report by CycDesigN on the design,
factors	to print the factor values, and
blocksizes	to print the feasible block sizes, and corresponding minimum and maximum numbers of replicated treatments in each block.

The SEED option allows you to supply seeds for the random numbers to be used within CycDesigN to search for the best design and to randomize it. You can specify a variate with two values to supply a different seed for each purpose, or a scalar to use the same one for both. If a zero value is specified, the corresponding seed is set automatically. The default is the scalar zero.

You can set option SPREADSHEET=design to put the design factors into a Genstat spreadsheet.

The TIMELIMIT defines the time in minutes to search. The default is 1.

Options: PRINT, LEVELS, NLOCATIONS, NBLOCKS, NUNITSPERLOCATION, NREPLICATEDPERBLOCK, TREATMENTS, LOCATIONS, BLOCKS, UNITS, SEED, SPREADSHEET, TIMELIMIT.

Parameters: none.

Method

The batch CycDesigN program is called using the SUSPEND directive. The underlying algorithm is described by Williams, John & Whitaker (2014).

References

John, J.A. & Williams, E.R. (1995). *Cyclic and Computer Generated Designs*. London: Chapman and Hall.

Williams, E.R., John, J.A. & Whitaker, D. (2014). Construction of more flexible and efficient p-rep designs. *Australian & New Zealand Journal of Statistics*, **56**, 89-96.

See also
Procedures: AFPREP, CDNAUGMENTEDDESIGN, CDNBLOCKDESIGN, CDNROWCOLUMNDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

CDNROWCOLUMNDESIGN

Constructs a row-column design using CycDesigN (R.W. Payne).

Options

`PRINT` = *strings*	Controls printed output (`design`, `report`, `factors`); default `*` i.e. none
`LEVELS` = *scalar* or *variate*	Numbers of levels of the treatment factors; if unset, takes the numbers of levels declared for the factors in the `TREATMENTSTRUCTURE` model
`NREPLICATES` = *scalar*	Number of replicates
`NROWS` = *scalar*	Number of rows
`NCOLUMNS` = *scalar*	Number of columns
`NGROUPS` = *variate*	Group sizes for a two-factor nested treatment structure
`TREATMENTFACTORS` = *factors*	Up to four factors to use in the treatment model: one factor for a one-way treatment model, two factors for a nested structure when `NGROUPS` is set, or two to four factors for a factorial treatment structure when `NGROUPS` is not set
`REPLICATES` = *factor*	Replicate factor
`ROWS` = *factor*	Row factor
`COLUMNS` = *factor*	Column factor
`RESOLVABLE` = *string*	Whether the design is resolvable (`yes`, `no`); default `no`
`METHOD` = *string*	How to construct the design (`onestage`, `twostage`, `unrestrictedtwostage`); default `ones`
`NRLATIN` = *scalar*	Number of contiguous rows to latinize; default 0 i.e. not latinized
`NCLATIN` = *scalar*	Number of contiguous columns to latinize; default 0 i.e. not latinized
`REPLATINGROUPS` = *variate*	Specifies the number of replicates in each column when constructing latinized designs; default * i.e. all in one column
`SPATIALMODEL` = *string*	Spatial model to use with a single-treatment-factor resolvable design (`integer`, `linearvariance`, `seconddifference`, `ev`); default * i.e. none
`EVDECAY` = *scalar*	Decay parameter to use when `SPATIALMODEL=ev`; default 0.5
`WEIGHTS` = *variate*	Variate with two values specifying weightings for the main effects and for the interactions in factorial treatment structures; default `!(1, 0.25)`
`RCWEIGHTS` = *variate*	Variate with three values specifying weightings for the within-row-and-column, between-row and between-column information; default has weight one for the within-row-and-column information, and the reciprocal of their numbers of levels for the rows and columns
`SEED` = *scalar* or *variate*	Scalar or variate with two values specifying seeds for the random numbers used by CycDesigN to search for the best design and to randomize it – if a scalar is specified the same seed is used for both purposes; default 0 i.e. set automatically
`SPREADSHEET` = *string*	Whether to put the design factors into a spreadsheet

	(`design`); default `*`
`TIMELIMITS` = *scalar* or *variate*	A scalar or a variate containing up to three numbers defining the time in minutes to spend on the first phase, the second phase and the spatial phase of the search (if the 2nd or 3rd numbers are omitted they default to the maximum of those specified); default 1
`NRANDOMIZATIONS` = *scalar*	Number of randomizations to generate from the best design; default 1
`TRIALS` = *factor*	Trials factor

No parameters

Description

CycDesigN is a package for the computer generation of experimental designs, which constructs optimal or near-optimal block and row-column designs; see the book *Cyclic and Computer Generated Designs* by John & Williams (1995). CycDesigN can also operate as a batch program, that can be called from within Genstat. This program is distributed with Genstat, and there are procedures to call the program, read its output back into Genstat, and form the relevant design factors. There are also Genstat add-in and resource files to define user menus, which can be downloaded from the VSNi website. However, before CycDesigN can be used, a license must be obtained; see vsni.co.uk/software/cycdesign for details.

This procedure, `CDNROWCOLUMNDESIGN`, uses the CycDesigN algoriths to form a row-column design. The treatment factors, whose values are to be formed, are specified by the `TREATMENTFACTORS` option. This can be set to a single factor if you want a one-way treatment structure. Alternatively, if option `NGROUPS` is set, you can supply two factors to define a nested model

```
factor_1 / factor_2
```

The group sizes (i.e. the number of levels of the second factor within each level of the first factor) are supplied by `NGROUPS`, in a variate. Otherwise, if option `NGROUPS` is not set, you can supply from two to four factors, to define a factorial model.

The `LEVELS` option can be used to define the numbers of levels of the factors, as a scalar if there is only one factor, or as a variate if there are several. The levels specified in the variate are assumed to be in the same order as the order in which the factors occur in the `TREATMENTFACTORS` list. `LEVELS` can be omitted if the factors have already been been declared with the right numbers of levels. Alternatively, if you want only a single treatment factor, and do not want to save its generated levels, you can specify its number of levels using `LEVELS`, and leave `TREATMENTFACTORS` unset.

The `NROWS` option can define the number of rows, and the `ROWS` option can supply a factor to save the levels generated for the row factor. You can omit `NROWS` if `ROWS` is set to a factor that has already been defined with the correct number of levels. Similarly, the `NCOLUMNS` option can define the number of columns, and the `COLUMNS` option can supply a factor to save the levels generated for the column factor. Finally, the `NREPLICATES` option can define the number of replicates of each treatment combination, and the `REPLICATES` option can supply a factor to save the levels generated for a replication factor.

Printed output is controlled by the `PRINT` option, with settings:

`design`	to print the design,
`report`	to print a report by CycDesigN on the design, and
`factors`	to print the factor values.

The `RESOLVABLE` option controls whether or not the design is *resolvable* i.e. whether the rows and columns can be grouped into replicates, each of which contains a single replicate of each treatment combination. Suppose that there are v treatment combinations, k rows, s columns and

r replicates of each treatment combination. Then in a non-resolvable design these numbers must satisfy the condition

$$v \times r = k \times s$$

whereas in a resolvable design the condition is

$$v = k \times s.$$

By default the design search has a single stage. However, you can use the METHOD option to request that two stages are used. The first constructs the column component design, and second forms the row-column design. (At the second stage the column component design is not changed; this is achieved by allowing only treatment interchanges to take place within columns.) If METHOD=unrestrictedtwostage, there is no restriction on the way in which the design is constructed during the first stage. Alternatively, if METHOD=twostage, an alpha design is constructed during the first stage for a resolvable design, or a cyclic design is constructed during the first stage for a non-resolvable design.

You can set option NRLATIN to request that the design is *latinized* by rows. If NRLATIN=1, the replication of each treatment combination is equalized as far as possible within each row. Alternatively, setting NRLATIN to *n*, say, aims to equalize the occurrence of the treatment combinations within each set of *n* contiguous rows. Similarly option NCLATIN can request that the design is latinized by columns. By default, the replicates in a latinized design are assumed to be in a single row, side by side. The REPLATINGROUPS option allows you to to define an alternative layout. For example, setting REPLATINGROUPS=!(1,2) when you have three replicates, defines two columns of replicates, the first with one replicate (replicate 1), and the second with two replicates (replicate 2 alongside replicate 1, and replicate 3 below replicate 2).

The SPATIALMODEL option allows you to request that the construction of a resolvable design should take account of the separation of different treatments in rows and columns. The principle is that plots close together are assumed to be correlated more than plots further apart; a spatial model attempts to model this correlation decay. The criterion used to generate spatial designs is the neighbour efficiency factor of Williams (1985), which has been extended to two-dimensional blocking structures by Williams, John & Whitaker (2005) and to cater for different decay functions. The available settings are

integer	integer,
linearvariance	linear variance,
seconddifference	modified second difference, and
ev	modified exponential variance.

The weights used with the first three settings are described by Williams (1985). The fourth setting, ev, is appropriate for a model specifying an autoregressive variance matrix. Its decay parameter is specified by the EVDECAY option; default 0.5. However, the spatial designs generated by CycDesigN are usually quite robust to the choice of weight function. Spatial models cannot be used with latinized designs.

The WEIGHTS option and the RCWEIGHTS option specify how to weight the importance of the information in the design. WEIGHTS is used when the design has a factorial treatment structure, and RCWEIGHTS is used when there is a single treatment factor or a nested treatment structure.

The WEIGHTS option specifies a variate with two values to define how to weight the efficiencies of the terms when there is a factorial treatment structure. The first value defines a weight for the main effects (default 1), and the second defines a weight for the interactions (default 0.25) These defaults are the same as those used in the stand-alone CycDesigN system.

For designs without a factorial treatment structure, the RCWEIGHTS option specifies a variate with three values to define the weightings to use for the within-row-and-column, between-row and between-column information. By default, the within-row-and-column information is given a weight of one, the between-row information has a weight equal to the reciprocal of the number of levels of the row factor, and the between-column information has a weight equal to the reciprocal of the number of levels of the column factor. These defaults are again the same as

those in the stand-alone CycDesigN system.

The SEED option allows you to supply seeds for the random numbers to be used within CycDesigN to search for the best design and to randomize it. You can specify a variate with two values to supply a different seed for each purpose, or a scalar to use the same one for both. If a zero value is specified, the corresponding seed is set automatically. The default is the scalar zero.

By default only one randomization is done with the best design. However, you can use the NRANDOMIZATION option to provide several randomizations of that design, for use in different trials. The TRIALS option can save a factor to identify the trials.

You can set option SPREADSHEET=design to put the design factors into a Genstat spreadsheet.

The TIMELIMITS option can be set to a scalar or a variate containing up to three numbers to define the time in minutes to spend on the first phase, the second phase and the spatial phase of the design search. If the second or third numbers are omitted, they default to the maximum of those specified. The default is 1.

Options: PRINT, LEVELS, NREPLICATES, NROWS, NCOLUMNS, NGROUPS, TREATMENTFACTORS, REPLICATES, ROWS, COLUMNS, RESOLVABLE, METHOD, NRLATIN, NCLATIN, REPLATINGROUPS, SPATIALMODEL, EVDECAY, SEED, WEIGHTS, RCWEIGHTS, SPREADSHEET, TIMELIMITS, NRANDOMIZATIONS, TRIALS.
Parameters: none.

Method

The batch CycDesigN program is called using the SUSPEND directive.

References

John, J.A. & Williams, E.R. (1995). *Cyclic and Computer Generated Designs*. London: Chapman and Hall.

Williams, E.R. (1985). A criterion for the construction of optimal neighbour designs. *J.R. Statist. Soc. B*, **47**, 487-497.

Williams, E.R., John, J.A. & Whitaker, D. (2005). Construction of resolvable spatial row-column designs. *Biometrics*, **62**, 103-108.

See also

Procedures: CDNAUGMENTEDDESIGN, CDNPREP, CDNBLOCKDESIGN, ARCSPLITPLOT.
Genstat Reference Manual 1 Summary section on: Design of experiments.

CENSOR

Pre-processes censored data before analysis by ANOVA (P.W. Lane).

Options

PRINT = *string token*	Whether to monitor convergence (monitor); default * implies no monitoring
TERM = *formula*	Formula for lowest stratum residual term; no default – this option must be set
DESIGN = *pointer*	Identifier specifying design information for ANOVA, or to save design information; default *
MAXCYCLE = *scalar*	Maximum number of iterations; default 20

Parameters

Y = *variates*	Observed variate with censored values represented by values greater than or equal to the bound; no default – this parameter must be set
BOUND = *scalars* or *variates*	Upper bound for censoring for each unit; no default – this parameter must be set
DF = *scalars*	Estimated residual d.f. for lowest stratum, adjusting for censoring; default *
NEWY = *variates*	Saves a variate with the censored values replaced by their estimates; if unset, the censored values are replaced in the original Y variate
SAVE = *identifiers*	Save details of each analysis for use in subsequent ADISPLAY or AKEEP statements

Description

An observation is said to be censored if it is known only that it is less than (or greater than) a particular value. Such observations can occur in designed experiments when the observed variable is the time until some event takes place for each experimental unit. For example, if the observed measurement is the lifetime of electric light-bulbs, it may happen that some bulbs are still alight when the experiment has to be concluded.

The response variate should be specified using the Y parameter, representing the censored values as values that are greater than or equal to the censoring bound. The bound is specified using the BOUND parameter, either as a scalar – if the bound is constant over the experiment – or as a variate of the same length as the response variate. Missing values in Y will be treated as usual, not as censored values. The procedure deals with the case of censoring with an upper bound. For a problem involving a lower bound, the structures in Y and BOUND should be multiplied by –1 before using the procedure (and the analysis interpreted accordingly).

The results for any experiment analysable by the ANOVA directive may be processed by the procedure. You must give BLOCKSTRUCTURE, TREATMENTSTRUCTURE and COVARIATE statements, as relevant, before using the procedure. If the analysis of the experiment requires a setting of the WEIGHTS, FACTORIAL, CONTRASTS or DEVIATIONS options of the ANOVA directive, you should give an ANOVA statement with these settings before using the procedure, setting the DESIGN option and then using the same identifier in the DESIGN option of CENSOR. The lowest stratum of the experiment must be identified explicitly in the BLOCKSTRUCTURE statement, rather than being implicitly taken as the *units* stratum by ANOVA; the model term representing this stratum must be specified using the TERM option of CENSOR. For example, a split-plot experiment with blocks might be specified by

```
BLOCKS block/plot/subplot
CENSOR [TERM=block.plot.subplot; ...
```

If you set the option PRINT=monitor, the procedure will print the values of the standard error of the lowest stratum at each cycle of the iterative estimation process. The maximum number of iterations is specified by the MAXCYCLE option, with a default of 20. The NEWY parameter allows you to specify a copy of the Y variate with the censored values replaced by their estimates. If NEWY is unset, the censored values are replaced in the original Y variate. The analysis of this variate can be displayed with ADISPLAY, or results saved with AKEEP. The save structure for the corresponding analysis of variance can be saved using the SAVE parameter.

The analysis will not be exact, due to the estimation of the censored values. However, the residual degrees of freedom in the lowest stratum may be corrected to the value output by the DF parameter (this is the absolute correction referred to by Taylor 1973; see the Method section).

Options: PRINT, TERM, DESIGN, MAXCYCLE. Parameters: Y, BOUND, DF, NEWY, SAVE.

Method

The censored units in the observed variate are replaced by estimated values, using the method outlined by Taylor (1973). This method estimates the expected value of each censored observation iteratively conditional on the fact that the value must be greater than the fixed bound, and using the relevant information from the other observations in the experiment. The iterative process is deemed to have converged when the relative change in each fitted value, and in the stratum standard error, is less than 0.1%. If convergence is not reached within the number of iteractions specified by the MAXCYCLE option, a message will be printed and the process will terminate. This should occur only if there is a high proportion of censored values, or if the design affords no information about one or more of the censored values.

Action with RESTRICT

The Y variate can be restricted, but the BOUNDS variate must not be.

Reference

Taylor, J. (1973). The analysis of designed experiments with censored observations. *Biometrics*, **29**, 35-43.

See also

Directive: ANOVA.
Procedure: TOBIT.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

CHECKARGUMENT

Checks the arguments of a procedure (R.W. Payne).

Option

ERROR = *scalar*	This scalar is given the value 1 if any errors are detected; it should have the value 0 on entry

Parameters

STRUCTURE = *identifiers*	Lists the structures (arguments) to be checked
VALUES = *variates* or *texts*	Defines the allowed values for a structure of type variate or text
DEFAULT = *identifiers*	Default to be used if STRUCTURE is set to an unset dummy
SET = *texts*	Indicates whether or not each structure must be set (no, yes); default no
DECLARED = *texts*	Indicates whether or not each structure must have been declared (no, yes); default no
TYPE = *texts*	Text for each structure whose values indicate the types allowed (scalar, factor, text, variate, matrix, diagonalmatrix, symmetricmatrix, table, expression, formula, dummy, pointer, LRV, SSPM, TSM, tree, asave, rsave, tsave, vsave); default *
PRESENT = *texts*	Indicates whether or not each structure must have values (no, yes); default no

Description

This procedure can be used to check that each argument of a procedure is set, that it is set to a structure of a valid type, that the structure has values, and (for structures of type text or variate) that the values belong to a defined set; unset arguments can be assigned a default. The information about each argument is specified by the parameters of CHECKARGUMENT; if there is anything that is not to be checked, the corresponding parameter should be left unset. The scalar specified by the ERROR option is set to 1 if an error is discovered, and an explanatory message is printed; this scalar should be initialized to zero before calling CHECKARGUMENT.

Option: ERROR.
Parameters: STRUCTURE, VALUES, DEFAULT, SET, DECLARED, TYPE, PRESENT.

Method

CHECKARGUMENT uses GETATTRIBUTE to obtain details of the structures being checked, ASSIGN to set defaults, and if-blocks to make the various tests.

See also

Directives: OPTION, PARAMETER, PROCEDURE.
Genstat Reference Manual 1 Summary section on: Program control.

CHIPERMTEST

Performs a random permutation test for a two-dimensional contingency table (L.H. Schmitt, M.C. Hannah & S.J. Welham).

Options

PRINT = *string tokens*	Output required (summary, observed, expected); default summ
PLOT = *string token*	What to plot (histogram); default hist
METHOD = *string token*	Method for calculating chi-square (pearson, maximumlikelihood); default pear
NTIMES = *scalar*	Number of permutations to make; default 999
SEED = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically

Parameters

DATA = *tables*	Table containing observed data
CHISQUARE = *scalars*	Saves the observed chi-square value
CHIPERMUTED = *variates*	Saves the chi-square values from the permuted data sets
PROBABILITY = *scalars*	Saves the probability value from the test

Description

The CHIPERMTEST procedure uses a permutation test to calculate the significance probability for a chi-square test of the independence of rows and columns in a two-dimensional contingency table. This provides a nonparametric alternative to the more usual chi-square test of independence (see the CHISQUARE procedure). The usual test depends upon the fact that the distribution of its so-called "chi-square" test statistic becomes a chi-square distribution as the numbers of observations become infinite. (Technically, we would say that the distribution is *asymptotically* chi-square.) However, the test is unreliable with smaller numbers, especially when the expected number in any cell of the table is less than five.

The permutation test simulates the random distribution of table values that may occur in tables that have the same overall distribution of numbers over the columns, and over the rows, as in the original table. We can assess the significance of the chi-square statistic that we can calculate from the observed table, by seeing where it lies in the distribution of statistics that we obtain from the permuted data.

The NTIMES option specifies how many permutations are done (default 999). The SEED option supplies the seed that is used in the RANDOMIZE directive to generate the permutations. The default of zero continues the existing sequence of random numbers if RANDOMIZE has already been used in the current Genstat job. If RANDOMIZE has not yet been used, Genstat picks a seed at random.

The DATA parameter supplies the observed data values, in a table with two classifying factors. The CHISQUARE can save the chi-square statistic calculated from the DATA table (in a scalar). The CHIPERMUTED parameter can save the chi-square statistics calculated from the permuted data sets (in a variate), and the PROBABILITY parameter can save the significance probability from the permutation test (in a scalar).

The PRINT option controls the output, with the following settings:

summary	prints a summary, containing the chi-square statistic, the minimum and maximum statistics calculated from the permuted data sets, and the probability (default);
observed	prints the DATA table; and
expected	prints the expected values for tables with the same overall

distribution of numbers over rows and over columns, but no interaction between the row and column factors (i.e. in a table where the rows and columns are independent).

By default, CHIPERMTEST plots a histogram showing the distribution of statistics obtained from the permuted data sets, with the chi-square statistic from the observed data superimposed as a vertical line. You can suppress this by setting option PLOT=*.

The METHOD option controls how the chi-square statistic is calculated. The default is to use the usual Pearson approximation (see the *Method* section), but you can set METHOD=likelihood to calculate it by maximum likelihood instead (using the Genstat facilities for generalized linear models).

Options: PRINT, PLOT, METHOD, NTIMES, SEED.
Parameters: DATA, CHISQUARE, CHIPERMUTED, PROBABILITY.

Method

The Pearson statistic is calculated as

chi-square = sum($(o-e) \times (o-e) / e$),

where o = observed, and e = expected. The alternative, maximum-likelihood method takes the deviance from fitting a generalized linear model with a log link and a Poisson distribution.

The permutations are constructed using the method Roff & Bentzen (1989).

Reference

Roff, D.A. & Bentzen, P. (1989). The statistical analysis of mitochondrial DNA polymorphisms: χ^2 and the problem of small samples. *Mol. Biol. Evol.*, **6**, 539-545.

See also

Procedure: CHISQUARE.
Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Regression analysis.

CHISQUARE

Calculates chi-square statistics for one- and two-way tables (A.D. Todd & P.K. Leech).

Options

PRINT = *string tokens*	Output required (test, probability, fittedvalues, tchisquare); default test, prob
METHOD = *string token*	Method for calculating chi-square (pearson, maximumlikelihood); default pear
GOODNESSOFFIT = *string token*	Whether to carry out a goodness-of-fit test for the DATA values against a supplied set of FITTEDVALUES (yes, no); default no

Parameters

DATA = *tables*	Table containing observed data
CHISQUARE = *scalars*	Scalar to save the chi-square value
DF = *scalars*	Scalar to supply or save the degrees of freedom
PROBABILITY = *scalars*	Scalar to save the probability value
FITTEDVALUES = *tables*	Table of expected values
RESIDUALS = *tables*	Table of standardized residuals
TCHISQUARE = *tables*	Table whose cells show the individual contributions to the chi-square value

Description

The CHISQUARE procedure calculates chi-square statistics. The DATA parameter supplies the data values. If these are in a two-way table, CHISQUARE produces the usual test of association between the row and column factor of the table; if a one-way table is supplied, the statistic assesses whether the different cells of the table contain different proportions of the data. Alternatively, you can set option GOODNESSOFFIT=yes to request a goodness-of-fit test between the data values and the set of expected values supplied by the FITTEDVALUES parameter; if you provide the degrees of freedom, using the DF parameter, the procedure can also calculate the probability value.

The PRINT option controls the printed output, with the settings: test to print the chi-square value and degrees of freedom; probability for the probability value; fittedvalues data, fitted (expected) values and standardized residuals; and tchisquare to show the contribution of each cell of the table to the chi-square value. By default, the statistic is calculated by the usual Pearson approximation (see the *Method* section), but you can set option METHOD=likelihood to calculate the chi-square by maximum likelihood (using the Genstat facilities for generalized linear models).

Parameters CHISQUARE, DF, PROBABILITY, FITTEDVALUES, RESIDUALS and TCHISQUARE allow the results to be saved in appropriate Genstat data structures.

Options: PRINT, METHOD, GOODNESSOFFIT.

Parameters: DATA, CHISQUARE, DF, PROBABILITY, FITTEDVALUES, RESIDUALS, TCHISQUARE.

Method

With option METHOD=pearson, the statistic is calculated by the usual Pearson formula:

$$\text{chi-square} = \text{sum}((o-e) \times (o-e) / e),$$

where o = observed, and e = expected.

If GOODNESSOFFIT=yes, the table e is supplied by the FITTEDVALUES parameter. Otherwise, for a one-way table e is the mean of the DATA values, while for a two-way table

e = (row total) × (column total) / (total in table).

For METHOD=maximumlikelihood, CHISQUARE takes the deviance from fitting a generalized linear model with a log link and a Poisson distribution.

See also

Procedures: CHIPERMTEST, CATRENDTEST, CMHTEST.

Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Regression analysis.

CINTERACTION

Clusters rows and columns of a two-way interaction table (J.T.N.M. Thissen & J. de Bree).

Options

PRINT = *string tokens*	What information to print (sortedtable, aovtable, summary, monitoring, variance, amalgamations, dendrogram); default sort, aov, summ, moni, vari, amal, dend
PRMONITOR = *scalar*	If option VARIANCE is set this provides a P-value to indicate when to start monitoring, if VARIANCE is unset PRMONITOR is ignored; default 0.95
VARIANCE = *scalar*	Variance of a mean in TABLE; default *
DF = *scalar*	Degrees of freedom of VARIANCE; default *
SSTHRESHOLD = *scalar*	Specifies a value of cumSS at which to partition the dendrograms and to define factors ROWGROUPS and COLGROUPS; default 0 i.e. no partitioning
TITLE = *text*	General title for the high-resolution graph; default *
PENSIZE = *scalar*	Pen size for y-labels of dendrograms; default 1

Parameters

TABLE = *tables*	Two-way table whose interaction structure is to be clarified
ROWAMALGAMATIONS = *matrices*	To either save or specify amalgamations for rows
COLAMALGAMATIONS = *matrices*	To either save or specify amalgamations for columns
ROWPERMUTATIONS = *variates*	To specify order of labels in the row dendrogram
COLPERMUTATIONS = *variates*	To specify order of labels in the column dendrogram
ROWGROUPS = *factors*	To save the grouping of the rows specified by the SSTHRESHOLD option
COLGROUPS = *factors*	To save the grouping of the columns specified by the SSTHRESHOLD option
SORTEDTABLE = *tables*	To save the sorted TABLE with increasing row and column means

Description

Consider an orthogonal table of uncorrelated, normally distributed means with common variance σ^2. Let the table be classified by two unstructured, qualitative factors between which an interaction has been detected. Such a table may emerge as a table of means from ANOVA, but this is not necessary. CINTERACTION performs a grouping of rows and columns of the table to identify a hopefully minimum number of groups which account for the overall interaction, but which are internally homogeneous. Grouping is accomplished by means of agglomerative hierarchical clustering as described in Corsten & Denis (1990).

The procedure goes through a sequence of steps. In each step the mean square for interaction is calculated for all possible subtables consisting of a pair of rows or a pair of columns of the full table. The pair of rows or columns with minimal mean square is merged, giving an updated table, and the process is repeated. Thus a sequence of amalgamations of rows and columns is produced, eventually leading to a 2×2 table. In this way the total sum of squares for the interaction is built up from orthogonal increments, each connected with a merge as described above, and insight into a possible structure of the interaction may be obtained. As a stopping rule for the merging process, Corsten & Denis (1990) suggest a simultaneous test procedure, which provides a probability of stopping too early, i.e. of ending up with too many groups.

The data for the procedure is a table, specified by the TABLE parameter. Missing values are

not allowed. You can also provide an estimate of σ^2, which is the common variance of the means in the table, together with its degrees of freedom by means of the options VARIANCE and DF.

Printed output is controlled by the PRINT option as follows. Setting PRINT=sortedtable prints a sorted table with increasing row and column means. PRINT=aovtable gives an analysis of variance which decomposes the total variation into that contributed by rows, columns and the interaction between rows and columns. If options VARIANCE and DF have been specified, setting PRINT=variance prints the estimate of the variance σ^2 together with its degrees of freedom. The effect of PRINT=monitoring depends on whether VARIANCE is specified or not. If VARIANCE is specified, PRINT=monitoring displays the sequence of merges starting just before the step in which the probability of stopping too early drops below the setting of the PRMONITOR option, the default setting of which is 0.95. Setting PRMONITOR=1 then displays the full sequence of merges. PRMONITOR is ignored if VARIANCE is not specified. PRINT=summary produces a summary of the clustering process giving, for each step, its number (in the column entitled *step*), the corresponding reduction of degrees of freedom due to the merging of the subtable (*df*), the mean square for interaction of the subtable which is merged (*ms*), the cumulated degrees of freedom (*cumdf*), the interaction sum of squares explained (*cumSS*), and (if options VARIANCE and DF have been set) the P-value of stopping too early (*P*). PRINT=amalgamations prints the amalgamation matrices. Finally, setting PRINT=dendrogram produces two dendrograms in a high-resolution graph, one for rows above the horizontal axis and one for columns below. The TITLE option can be used to supply a title for the dendrograms. By default all this information is printed.

The ROWAMALGAMATIONS and COLAMALGAMATIONS parameters may be used for saving the amalgamation matrices of rows and columns respectively, and contain information for drawing the dendrograms (as from directive HCLUSTER). The sorted table may be saved using the parameter SORTEDTABLE.

Saving the amalgamations matrices can be useful if you wish to modify the layout of the dendrogram after inspecting the results. To do this, you set the TABLE parameter as before, and set ROWAMALGAMATIONS and COLAMALGAMATIONS to the saved amalgamation matrices. Options SSTHRESHOLD and PENSIZE, and parameters ROWPERMUTATION and COLPERMUTATION, can then be used to control the layout of the dendrogram. By setting option SSTHRESHOLD to a specific value for the abscissa (i.e. cumSS) the grouping of rows and columns corresponding to this value is represented by a vertical dotted line in the dendrograms; by default value no line is drawn. The line is drawn in the default style of pen 3. You may wish to define a particular line style for this pen, since the appearance of lines is device specific. The groupings at this point can be saved using the ROWGROUPS and COLGROUPS parameters. Parameters ROWPERMUTATION and COLPERMUTATION specify the order of the labels in the row and column dendrograms, starting from the horizontal axis.

Options: PRINT, PRMONITOR, VARIANCE, DF, SSTHRESHOLD, TITLE, PENSIZE.
Parameters: TABLE, ROWAMALGAMATIONS, COLAMALGAMATIONS, ROWPERMUTATIONS, COLPERMUTATIONS, ROWGROUPS, COLGROUPS, SORTEDTABLE.

Method

In each step of the merging process the pair of rows or columns which contributes least to the total sum of squares for interaction is traced by repeated use of (weighted) ANOVA. When the number of data in each cell, which may have increased by previous merges, is properly accounted for, each of the successive sum of squares is the squared projection of the (tabulated) data vector on a subspace of interactions, orthogonal to all preceding such subspaces. In this way cumSS increases with hopefully minimal speed, but this is not guaranteed because of the sequential character of the procedure.

The P-values as presented in the summary table in Corsten & Denis (1990), are incorrect in

that they apparently have been obtained by using D instead of *n* degrees of freedom in the denominator of the F-statistic concerned.

Action with `RESTRICT`
Restrictions are not allowed.

Reference
Corsten, L.C.A. and Denis, J.B. (1990). Structuring interaction in two-way tables by clustering. *Biometrics*, **46**, 207-215.

See also
Directives: `ANOVA`, `CLUSTER`, `HCLUSTER`.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, Multivariate and cluster analysis.

CLASSIFY

Obtains a starting classification for non-hierarchical clustering (S.A. Harding).

No options

Parameters

DATA = *pointers*	Each pointer contains a set of variates giving the properties of the units to be grouped
NGROUPS = *scalars*	Indicates the number of groups required
GROUPS = *factors*	Stores the classifications formed

Description

In non-hierarchical classification an initial classification is required, and it is advantageous to have these classes as homogeneous as possible. This reduces the risk of converging to a local optimum, and also encourages faster convergence of the iterative transfer algorithm used by the CLUSTER directive.

The attributes of the units to be formed into groups are specified in a set of variates; these should be placed into a pointer for use as the setting for the DATA parameter. The number of groups required is specified by the NGROUPS parameter; this must not be greater than the number of units. The group allocations that are formed are stored in the factor indicated by the GROUPS parameter. This factor need not be declared in advance but will be formed by the procedure.

Options: none. Parameters: DATA, NGROUPS, GROUPS.

Method

When the number of groups is greater than the number of data variates plus one, CLASSIFY forms the groups according to the positions of the units in the first dimension of a principal coordinates analysis (PCO) of the DATA variates.

Otherwise it tries to find a suitable classification into the k groups by finding the k units that are furthest apart in p-dimensional space (where p is the number of variates). These are then used as nuclei for the classes, with each of the remaining units being allocated to the class with the nearest nucleus.

The units defining the nuclei are found by first finding the two units that are furthest apart. The third unit is the unit with greatest distance from the line joining the first two units. The fourth is the unit with greatest distance from the plane containing the first three units, and so on until the kth unit is the unit furthest from the $(k-2)$ dimensional space spanned by the $(k-1)$ units already found.

Action with RESTRICT

The variates must not be restricted.

See also

Directive: CLUSTER.

Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

CMHTEST

Performs the Cochran-Mantel-Haenszel test (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (test); default test
CLASSIFICATION = *factors*	Classifying factors for a DATA variate or classifying factors for the *R*×*C* tables in a DATA table
CONTINUITY = *string token*	Continuity correction for 2×2×*K* Mantel-Haenszel test (correct, none); default corr
CIPROBABILITY = *scalar*	Size of confidence interval for common odds ratio in 2×2×*K* tables; default 0.95

Parameters

DATA = *tables* or *variates*	Data values
STATISTIC = *scalars*	Save the test statistic
PROBABILITY = *scalars*	Save the probability for the test
ODDSRATIO = *scalars*	Save the common odds ratio for the 2×2×*K* table case
LOWER = *scalars*	Save lower limit of the confidence interval of odds ratio
UPPER = *scalars*	Save upper limit of the confidence interval of odds ratio

Description

CMHTEST performs the Cochran-Mantel-Haenszel test for average partial association between two nominal variables adjusting for control variables. The data are represented by a series of *K* (*R*×*C*) contingency tables, where *K* represents the strata for the control variables. If there are two or more control variables then these are combined to form a single factor (K) with a level for every combination of the control factors. For the case where there are two dichotomous variables of interest, i.e. a series of *K* (2×2) tables, CMHTEST calculates the Mantel-Haenszel chi-square statistic, and an overall estimate of relative risk as described in Mantel & Haenszel (1959). Otherwise the Generalized Cochran-Mantel-Haenszel test is used, as in Landis *et al*. (1978).

The data can be supplied as a table using the DATA parameter where the first two classifying factors of the table indicate the variables of interest, and the remaining factors are combined to form a factor with a level for every combination of the remaining factors. If the first two classifying factors are not the ones of interest, then the CLASSIFICATION option can be used to supply the names of the classifying factors to use. The data can also be supplied in variates, with the CLASSIFICATION option set to the classifying factors and the first two factors in the list indicating the variables of interest. For a series of *K* (2×2) tables the CONTINUITY option can be used to control whether to apply a continuity correction to the Mantel-Haenszel chi-square test.

The PRINT option controls printed output, with settings:

test	the test statistic and probability, also the common odds ratio and confidence interval when there are *K* (2×2) tables

A 95% confidence interval is calculated for the common odds ratio, but this can be changed by setting the CIPROBABILITY option to the required value (between 0 and 1).

The test statistic can be saved using the STATISTIC parameter, and the probability can be saved using the PROBABILITY parameter. For a series of *K* (2×2) tables the odds ratio, lower and upper odds-ratio confidence interval can be saved with the ODDSRATIO, LOWER and UPPER parameters respectively.

Options: PRINT, CLASSIFICATION, CONTINUITY, CIPROBABILITY.
Parameters: DATA, STATISTIC, PROBABILITY, ODDSRATIO, LOWER, UPPER.

Method

For each table i, $i = 1...K$

a_i	b_i	n_{1i}
c_i	d_i	n_{2i}
m_{1i}	m_{2i}	N_i

the Mantel-Haenszel Test is calculated by:

$$MH = (|(\sum a_i - \sum((n_{1i} \times m_{1i}) / N_i))| - 0.5)^2 / \sum((n1i \times n_{2i} \times m_{1i} \times m_{2i}) / (N_i^2 \times (N_i - 1)))$$

where the continuity correction (0.5) is used if option `CONTINUITY=correct`. The common odds-ratio is calculated by

$$OR = \sum_{i=1 \text{ to } K} R_i / \sum_{i=1 \text{ to } K} S_i$$

where

$$R_i = (a_i \times d_i) / N_i$$

$$S_i = (b_i \times c_i) / N_i$$

The variance for the odds-ratio is estimated using the method outlined in Robins *et al.* (1986).

The Generalized Cochran-Mantel-Haenszel test is calculated using the method of Landis *et al.* (1978).

Action with `RESTRICT`

If a parameter is restricted the statistics will be calculated using only those units included in the restriction.

References

Landis J,L., Heyman, E,R. & Koch, G.G. (1978). Average Partial Association in Three-way Contingency Tables: a Review and Discussion of Alternative Tests. *International Statistical Review*, **46**, 237-254.

Mantel N. & Haenszel W. (1959). Statistical Aspects of the Analysis of Data From Retrospective Studies of Disease. *Journal National Cancer Institute*, **22(4)**, 719-748.

Robins J, Breslow N, & Greenland S. (1986). Estimators of the Mantel-Haenszel variance consistent in both sparse data and large-strata limiting models. *Biometrics*, **42**, 311-323.

See also

Procedures: `CHISQUARE`, `CHIPERMTEST`, `FCORRELATION`, `KCONCORDANCE`, `KTAU`, `LCONCORDANCE`, `SPEARMAN`.

Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Regression analysis.

CONFIDENCE

Calculates simultaneous confidence intervals (D.M. Smith).

Options

PRINT = *string token*	Controls printed output (intervals); default inte
METHOD = *string token*	Type of interval (individual, smm, product, Bonferroni, Scheffe); default smm
MU = *scalar*	Value for population mean checked as to whether in the confidence interval; default * i.e. no checking
PROBABILITY = *scalar*	The required significance level; default 0.05

Parameters

MEANS = *tables* or *variates*	Mean values
REPLICATIONS = *scalars* or *tables* or *variates*	Number(s) of observations per mean
VARIANCE = *scalars*	Estimate of variance
DF = *scalars*	Degrees of freedom
XCONTRASTS = *matrices*	Matrix of coefficients of orthogonal contrasts
LABELS = *texts*	Identifiers of mean values
LOWER = *tables* or *variates*	Lower values of confidence intervals
UPPER = *tables* or *variates*	Upper values of confidence intervals

Description

CONFIDENCE calculates a set of simultaneous confidence intervals i.e. intervals whose formation takes account of the number of intervals formed and the fact that the intervals are (slightly) correlated because of the use of a common variance (see Hsu 1996 and Bechhofer, Santner & Goldsman 1995). The methodology implemented in the procedure closely follows that described in Section 1.3 of Hsu (1996).

The means are input using the MEANS parameter, either in a table saved e.g. from AKEEP, or in a variate. The replication (or number of observations in each mean) is supplied by the REPLICATIONS parameter, either in a scalar (if all the replications are equal) or in a structure of the same type as the means. The estimate of the variance (usually a pooled estimate as given by the residual mean square in ANOVA, and accessible using the VARIANCE parameter of AKEEP) and its corresponding degrees of freedom are input as scalars using the VARIANCE and DF parameters respectively. Confidence limits can be formed for contrasts amongst the means by supplying the matrix defining the contrasts using the XCONTRASTS parameter. Each row of the matrix contains a contrast similarly to the specification in the REG function in ANOVA but, unlike REG, the contrasts must all be orthogonal. The LABELS parameter can be used to supply labels for the means or for the contrasts, while the LOWER and UPPER parameters allow the limits of the confidence intervals to be saved.

The type of interval to be formed is specified by the METHOD option, with settings individual, smm (studentized maximum modulus), product (inequality), Bonferroni and Scheffe. The setting individual calculates the intervals as if they were independent, each with the input probability. The setting smm calculates the intervals as correlated, each with a probability adjusted for the multiplicity of intervals. The two settings product and Bonferroni calculate the intervals as independent, but with a probability adjusted for the multiplicity of intervals. These two settings produce very similar intervals although the Bonferroni intervals are always slightly larger. The final setting Scheffe calculates the intervals using privoted F statistics. Hsu (1996, Section 1.3.7) should be referred to for details of this last setting. The default setting is smm because it produces exact simultaneous confidence intervals.

The MU option allows you to supply a (population) mean to be tested for inclusion in each

interval, and the PROBABILITY option allows the experiment-wise significance level for the intervals to be changed from the default of 0.05 (i.e. 5%). The interval-wise significance level is calculated according to the setting of METHOD.

You can set option PRINT=* to suppress printing of the intervals; by default PRINT=intervals.

Options: PRINT, METHOD, MU, PROBABILITY.
Parameters: MEANS, REPLICATIONS, VARIANCE, DF, XCONTRASTS, LABELS, LOWER, UPPER.

Method

The methodology implemented is based on that described and reviewed in Hsu (1996), and Bechhofer, Santner & Goldsman (1995).

References

Bechhofer, R.E., Santner, T.J. & Goldsman, D.M. (1995). *Design and Analysis of Experiments for Statistical Selection, Screening, and Multiple Comparisons*. Wiley, New York.
Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also

Procedures: AMCOMPARISON, AUMCOMPARISON, AMDUNNETT, VMCOMPARISON.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

CONVEXHULL

Finds the points of a single or a full peel of convex hulls (P.G.N. Digby).

Options

PEELING = *string token*	Specifies whether the procedure is to form the full set of peels, or just the convex hull (no, yes); default no
SCALE = *scalar*	Scaling factor for hulls; default 1.0

Parameters

Y = *variate*	Y-coordinates of the points
X = *variate*	X-coordinates of the points
YHULL = *variate* or *pointer*	Variate storing the y-coordinates of the points defining the convex hull (for PEELING=no) or pointer to a set of variates storing the y-coordinates of the convex hulls forming the complete set of peels
XHULL = *variate* or *pointer*	Variate storing the x-coordinates of the points defining the convex hull (for PEELING=no) or pointer to a set of variates storing the x-coordinates of the convex hulls forming the complete set of peels
PEEL = *variate*	Stores the number of the peel to which each point belongs

Description

The convex hull of a set of points in two dimensions is the convex polygon that exactly encloses all the points. The operation of repeatedly finding the convex hull for a set of points, removing all the points that lie on hull, and then finding the next hull is known as (convex hull) peeling; eventually, either all the points will be deleted, or a single point will remain. Peeling can be used, for example, to calculate robust estimates of location or to give a bivariate analogue of a box-and-whisker plot; see Green (1981).

The coordinates of the set of points are given by the X and Y parameters, and the convex hull or hulls are output by the XHULL and YHULL parameters. By default, a single hull is formed and the parameters XHULL and YHULL return a pair of variates. To construct hulls representing a full set of peels, option PEELING should be set to yes. Parameters XHULL and YHULL then return pointers containing a variate for each hull, and the convex hull can be displayed by plotting YHULL[1] against XHULL[1], the second peel by plotting YHULL[2] against XHULL[2], and so on. The variates defining each convex polygon need not contain all of the points on the polygon but only those at the vertices. The first point in each variate is repeated as an extra last point, so that the polygon is closed. For high-resolution plotting, the parameters of PEN are best set as

```
LINESTYLE=1; METHOD=line; JOIN=given; SYMBOL=0.
```

The SCALE option supplies a scale factor that can be applied to the convex hulls, so that they are suitable for plotting as cosmetic "envelopes" enclosing the points. Values in the range 1.15 to 1.20 have been found suitable with test data sets. Plots of these data envelopes can be further enhanced by setting the PEN method to closed.

The fifth parameter, PEEL, returns a variate that indicates the number of the peel on which each point lies. Thus, for example, the PEEL variate will store value 1 for all the points on the convex hull itself; if PEELING=no, all the other points will have the value zero.

Option: PEELING.
Parameters: Y, X, YHULL, XHULL, PEEL.

Method
Each convex hull is defined by a set of points. The first point is at the minimum y value; if there are several such points then one with the maximum absolute value of x is chosen. The search for subsequent points proceeds in an anti-clockwise direction: at each stage the furthest possible point is used to define the convex hull; however, any intervening (or coincident) points are deemed to lie on the convex hull, and so will be removed if peeling is taking place. Peeling is performed by repeatedly restricting the relevant variates at each stage.

Action with RESTRICT
If either the X or the Y variates is restricted, the procedure will consider only the points not excluded by the restriction. The PEEL variate will be of the same length as X and Y and will be similarly restricted.

Reference
Green, P.J. (1981). Peeling bivariate data. In: *Interpreting Multivariate Data* (ed. V.Barnett). Wiley, New York.

See also
Procedures: PTBOX, PTSINPOLYGON.
Genstat Reference Manual 1 Summary sections on: Graphics, Spatial statistics.

CORANALYSIS

Does correspondence analysis, or reciprocal averaging (P.G.N. Digby & A.I. Glaser).

Options

PRINT = *string tokens*	Printed output from the analysis (roots, rowscores, rowinertias, rowchisquare, rowmass, rowquality, colscores, colinertias, colchisquare, colmass, colquality); default * i.e. no output
METHOD = *string token*	Type of analysis required (correspondence, digbycorrespondence, biplot, reciprocal); default corr
NROOTS = *scalar*	Number of latent roots for printed output; default * requests them all to be printed
%METHOD = *string token*	How to represent proportions or %s in quality statistics (permills, percentages, proportions); default prop
NDIMENSIONS = *scalar*	Number of dimensions for which quality statistics are required; default 2
ROWSUBSET = *scalars*	Indexes of subset rows
COLSUBSET = *scalars*	Indexes of subset columns
ROWPASSIVE = *scalars*	Indexes of passive rows
COLPASSIVE = *scalars*	Indexes of passive columns

Parameters

DATA = *matrices* or *data matrices*	Data to be analysed
ROOTS = *diagonal matrices*	Saves the squared singular values from each analysis
ROWSCORES = *matrices*	Saves the scores for the rows of the data matrix
COLSCORES = *matrices*	Saves the scores for the columns of the data matrix
ROWINERTIAS = *matrices*	Saves the inertias for the rows of the data matrix
COLINERTIAS = *matrices*	Saves the inertias for the columns of the data matrix
ROWQUALITY = *matrices*	Saves the quality statistics for rows of the data
COLQUALITY = *matrices*	Saves the quality statistics for columns of the data
SAVE = *pointers*	Saves details of the analysis for use by CABIPLOT

Description

Correspondence analysis is an ordination technique used to analyse two-way categorical data tables. Ordination techniques approximate relationships between variables in a reduced number of dimensions.

The type of analysis is specified by the METHOD option, with one of the following settings:

correspondence	correspondence analysis (Greenacre 1984),
digbycorrespondence	an alternative implementation of correspondence analysis described by Digby & Kempton (1987),
reciprocal	reciprocal averaging (see Digby & Kempton 1987), or
biplot	a similar biplot-style analysis (again see Digby & Kempton 1987).

The default setting is correspondence, and this should be retained if either of the options to subset rows or columns are set.

The data for the procedure are specified by the DATA parameter as either a matrix or a datamatrix (i.e. a pointer to variates, all with the same length). The matrix must not contain any missing values; it is unchanged on exit from the procedure.

Printed output is controlled by the PRINT option with settings:

roots	to print the roots (together with the roots expressed as percentages and cumulative percentages),
rowscores	to print the scores for the rows of the data matrix,
rowinertias	to print the inertias for the rows of the data matrix,
rowmass	to print the row masses,
rowchisquare	to print the row chi-square distances,
rowquality	to print the quality statistics for the rows,
colscores	to print the scores for the columns of the data matrix,
colinertias	to print the inertias for the columns of the data matrix,
colmass	to print the column masses,
colchisquare	to print the column chi-square distances, and
colquality	to print the quality statistics for the columns.

The NROOTS option controls the printed output of roots, scores and inertias. By default, results are printed for all the roots, but you can set the NROOTS option to specify a lesser number.

The quality settings produce tables with the following columns:

- the mass of the row (or column), in proportion to the total mass;
- the "quality" of the representation i.e. how much of the inertia of a row (or column) is represented by the dimensions shown;
- the proportion of the total inertia of the row (or column) compared to the total inertia for all rows (or columns);
- principal coordinates of the rows (or columns) in the specified dimension;
- the amount of inertia for each row (or column) in the specified dimension relative to the total amount of inertia given by the value of the quality statistic – hence the sum of a specific row (or column) across the dimensions shown will be equal to the value given by the quality statistic;
- the proportion of inertia explained by a row (or column) in a dimension, compared to the total inertia in that dimension.

The representation of the columns of proportions is controlled by the %METHOD option; these can be printed either as proportions (default), percentages or as permills i.e. tenths of a percent. The NDIMENSIONS option specifies the number of dimensions for which to print quality statistics; default 2.

When carrying out correspondence analysis, there may be rows and/or columns (for example outliers with low mass) that you would like to ignore during the calculation of the roots or inertia, so that they have no influence. Instead of removing these rows and/or columns from the data before running CORANALYSIS, an alternative is to list the indexes of the rows or columns that are to be ignored using the ROWPASSIVE and/or COLPASSIVE options. These "passive" rows will still be included in the table of quality statistics, where their relative contributions will be shown and compared to total for all the passive rows or columns.

You may want to apply a correspondence analysis calculated from the whole data set onto only a subset of the rows and/or columns when some of the rows and/or columns divide into groups with common traits. This can be done by setting the ROWSUBSET and/or COLSUBSET options to the indexes of the rows and/or columns indexes in the subset of interest. If any of these options is set, the METHOD option must be set to correspondence. If ROWPASSIVE and ROWSUBSET (or COLPASSIVE and COLSUBSET) are both set, any indexes that occur in both will be removed from the ROWSUBSET (or COLSUBSET).

Results from the analysis can be saved using the parameters ROOTS, ROWSCORES, COLSCORES, ROWINERTIAS, COLINERTIAS, ROWQUALITY and COLQUALITY. The structures specified for these parameters need not be declared in advance. The SAVE parameter can save full details of the analysis for use by the CABIPLOT procedure.

Options: PRINT, METHOD, NROOTS, %METHOD, NDIMENSIONS, ROWSUBSET, COLSUBSET, ROWPASSIVE, COLPASSIVE.

Parameters: DATA, ROOTS, ROWSCORES, COLSCORES, ROWINERTIAS, COLINERTIAS, ROWQUALITY, COLQUALITY, SAVE.

Method

Full details of correspondence analysis (i.e. METHOD=correspondence) are given by Greenacre (1984 & 2007). The other methods are described by Digby & Kempton (1987).

The data matrix X, is scaled to have sum one for METHOD settings correspondence and digbycorrespondence. The matrices U, S and V are taken from the singular-value decomposition of

$$Y = (X - R\,C) / \sqrt{(R\,C)}$$

for METHOD=correspondence and

$$Y = (R^{-½}\,X\,C^{-½})$$

for the other methods, where R and C are diagonal matrices of row and column totals of the data matrix X. The scores for the rows and columns from METHOD=correspondence are

$$A = (R^{-½}\,U)$$

and

$$B = (C^{-½}\,V)$$

The scores from METHOD=digbycorrespondence are similar, but are multiplied by S. This makes the row scores obtained here the same as the principal coordinates given with the quality statistics.

With the other two methods X is not scaled to total one, and the scores are given by $A = (R^{-½}\,U\,S^m)$ and $B = (C^{-½}\,V\,S^m)$: the parameter m is zero for METHOD=reciprocal, and 0.5 for METHOD=biplot.

The inertia values for the rows and columns are given by

$$(R\,A\,A')\,S'$$

and

$$(C\,B\,B')\,S'$$

where $S' = S$ for METHOD=correspondence, and $S = 1$ for the other methods; see Greenacre (1984) for further information.

The roots are the squares of the singular values. Note that the first singular value will always be one for methods other than correspondence; this corresponds to a trivial solution given in the first column of A and B above, which is automatically removed from the results printed and saved from CORANALYSIS.

Rows and/or columns chosen as passive rows and/or columns are separated from the original data matrix before it is scaled. Rows and/or columns chosen as subset rows and/or columns are separated from Y after this scaling.

For the quality statistics, the weighted sum-of-squares of the principal coordinates on the ith dimension is equal to the ith squared singular value. The row and column scores for METHOD=digbycorrespondence are equivalent to the principal coordinates. Conversely the row and column scores for METHOD=correspondence or reciprocal are equivalent to standard coordinates, where the weighted sum-of-squares for each dimension is equal to one.

References

Digby, P.G.N. & Kempton, R.A. (1987). *Multivariate Analysis of Ecological Communities*. Chapman & Hall, London.

Greenacre, M.J. (1984). *Theory and Applications of Correspondence Analysis*. Academic Press, London.

Greenacre, M. (2007). *Correspondence Analysis in Practice, second edition*. Chapman & Hall, London.

See also
Procedures: CABIPLOT, MCORANALYSIS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

COVDESIGN

Produces experimental designs efficient under analysis of covariance (D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output (design, cefficiency, means, histogram, cutoff); default desi, ceff, cuto
TREATMENTSTRUCTURE = *formula*	Treatment terms to be fitted
BLOCKSTRUCTURE = *formula*	Block model for the design
COVARIATES = *variates*	Covariates for the design
FACTORIAL = *scalar*	Limit on number of factors in a treatment term; default 3
GRBLOCKSTRUCTURE = *formula*	Formula use for randomization; default uses BLOCKSTRUCTURE
EXCLUDE = *factors*	(Block) factors whose levels are not to be randomized
UNITS = *text*, *variate* or *factor*	Labels for the units of the design

Parameters

PROPORTION = *scalars*	Upper proportion of the combined cov. ef. distribution from which the design is to be chosen (or zero to take the best design found); default 0.5
NSIMULATIONS = *scalars*	Number of designs to simulate for the empirical distribution of combined cov. ef.'s; default 100
WEIGHTS = *variates*	Weighting for the treatment terms to use when calculating the combined cov. ef.; default 1 (i.e. all equal)
CEFLIMIT = *scalars*	Minimum value of the cov. ef. for each or variates treatment term for a design to be included in the set of acceptable designs; default 0 (i.e. all designs acceptable).
ORDER = *scalars*	Order of polynomial to fit for each covariate; or variates default 1 (i.e. only linear covariates)
SEED = *scalars*	Seed for random number generator for randomizing the simulated designs; default 0
SAVE = *pointers*	Saves the treatment factor allocations for the selected design; if unset, these overwrite the values of the treatment factors themselves
CUTOFF = *scalars*	Critical value of the combined cov. ef. from the simulated distribution
CEFFICIENCY = *variates*	Covariate efficiencies for the treatment terms from the selected design
SIMULATIONS = *variates*	Simulated combined cov. ef.'s

Description

When a covariate is fitted in an analysis of variance, there can be a loss of efficiency in the estimation of the treatment effects. A measure of this loss of efficiency is printed in a column of the analysis of variance table headed. "cov. ef." (an abbreviation for the covariance efficiency factor). A value of the cov. ef. close to 1 represents very little loss in efficiency through fitting the covariate. In good designs, the treatment means for a covariate will be similar, so that only small adjustments will be required in estimating the response-variate treatment means in the analysis of covariance.

Where the covariates are available before the allocation of treatments to units, the randomization of the design may be restricted to ensure high covariate efficiency for all

treatment factors. Cox (1957, 1982) suggests the approach of restricting the randomization, so that only designs with values of cov. ef. in the top ranked set of the full randomization set are chosen. The proportion of acceptable designs is set via the PROPORTION option. If this is greater than zero, COVDESIGN randomly generates the number of designs specified by the NSIMULATIONS parameter to obtain an empirical distribution for the covariance efficiencies. It then generates further designs until it finds one within the acceptable proportion. Alternatively, Harville (1974, 1975) suggests ignoring randomization, and only taking the design with the optimal cov. ef. value. If PROPORTION is set to zero, COVDESIGN instead takes the best design out of the NSIMULATIONS randomly generated designs. You can provide a seed, using the SEED parameter, for the randomizations used to generate the designs. The default, SEED=0, sets the seed automatically.

The treatment and block models for the design are specified by the TREATMENTSTRUCTURE and BLOCKSTUCTURE options, respectively, and the COVARIATES option lists the covariates. The FACTORIAL option specifies the maximum order of treatment term to fit; default 3. Usually the design is randomized according to the BLOCKSTRUCTURE, but you can specify an alternative model for randomization using the GRBLOCKSTRUCTURE option. The EXCLUDE option can supply a list of blocking factors that are not to be randomized, similarly to the EXCLUDE option of the RANDOMIZE directive. COVDESIGN usually uses RANDOMIZE for the randomization, and so there is the constraint that the block-factor combinations must all have replication one, and the block model must contain only the operators * and /. However, if EXCLUDE is unset and the randomization structure consists of a single factor, COVDESIGN uses the URAND function of CALCULATE and the SORT directive. Under these circumstances, for example, the blocks need not be of equal sizes.

When there are several treatment terms, each one may have a different cov. ef. COVDESIGN combines these into a combined cov. ef. over all the terms, calculated as the geometric mean. You can provide a variate of weights for the terms, using the WEIGHTS parameter.

The CEFLIMIT parameter can be set to restrict the designs from which the resulting design is selected. A design is acceptable if the values of cov. ef. for each treatment term (main effects and interactions, if fitted) are greater than the corresponding value in CEFLIMIT. The values in CEFLIMIT must be between 0 and 1. If CEFLIMIT is a scalar, a common minimum value is applied to each treatment term.

Higher order covariate balance on the treatments can be obtained by including polynomial covariate terms. The ORDER parameter specifies the degree of the polynomial to be included for each covariate. For example, setting ORDER to 2 would force both the mean and variance of the covariates in each treatment group to be balanced. The default ORDER=1 includes only the usual linear covariates.

The PRINT option controls prointed output, with settings:

cutoff	the critical value for the combined cov. ef. defining the acceptable PROPORTION of designs;
design	the treatment allocations in the resulting design;
efficiency	the cov. ef.'s in the resulting design;
histogram	histogram of the combined cov. ef.'s in the simulations;
means	covariate means by treatments

By default PRINT=design,cefficiency,cutoff. The UNITS option allows you to specify a text, variate or factor to label the units in the design output.

The SAVE parameter allows you to supply a pointer to save the values of the treatment factors for the best design. If this is not set, COVDESIGN saves them by redefining the values of the original treatment factors. The cov. ef.'s for the best design can be saved using the CEFFICIENCY parameter, in a scalar if there is only one treatment term, or in a variate if there are several. The combined cov. ef.'s from the simulations can be saved using the SIMULATIONS parameter. The CUTOFF parameter can save the critical value for the combined cov. ef. defining

the acceptable PROPORTION of designs.

Options: PRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATES, FACTORIAL, GRBLOCKSTRUCTURE, EXCLUDE, UNITS.
Parameters: PROPORTION, NSIMULATIONS, WEIGHTS, CEFLIMIT, ORDER, SEED, SAVE, CUTOFF, CEFFICIENCY, SIMULATIONS.

Action with RESTRICT
Any restrictions on covariate, treatment factors, block factors or units structures are cancelled.

References
Cox, D.R. (1957). The use of a concomitant variable in selecting an experimental design. *Biometrics*, **13**, 150-158.
Cox, D.R. (1982). Randomization and concomitant variables in the design of experiments. In: *Statistics and Probability* (ed. G. Kallianpur, R. Krishnaiah & J.K. Ghosh), 777-790. North Holland, New York.
Harville, D.A. (1974). Nearly optimal allocation of experimental units using observed covariate values. *Technometrics*, **16**, 589-599.
Harville, D.A. (1975). Computing optimum designs for covariance models. In: *A Survey of Statistical Design and Linear Models* (ed. J.N. Srivastava). North-Holland. Amsterdam.

See also
Directives: ANOVA, COVARIATE.
Genstat Reference Manual 1 Summary sections on: Design of experiments, Analysis of variance.

CRBIPLOT

Plots correlation or distance biplots after RDA, or ranking biplots after CCA (A.I. Glaser).

Options

DIMENSIONS = *scalars*	Two numbers specifying which axes of the ordinations to plot; default 1,2
PLOT = *string token*	Whether to plot site or species scores (sitescores, speciesscores); default spec
WINDOW = *scalar*	Which graphical window to use; default 1
SAVE = *pointer*	Supplies results from an ordination analysis by CCA or RDA; default uses the most recent analysis

Parameters

X1 = *scalars*, *variates* or *texts*	First explanatory variable to plot; default 1
X2 = *scalars*, *variates* or *texts*	Second explanatory variable to plot; default * i.e. none
LMXVARIABLES = *string tokens*	How to label the x-variables (identifiers, labels, none, numbers); default labe if LXVARIABLES is set, otherwise iden
LMSPECIES = *string tokens*	How to label the species scores (identifiers, labels, none, numbers); default labe if LSPECIES is set, otherwise numb
LMSITES = *string tokens*	How to label the site scores (labels, none, numbers); default labe if LSITES is set, otherwise numb
LXVARIABLES = *texts*	Labels for variables
LSPECIES = *texts*	Labels for species scores
LSITES = *texts*	Labels for site scores

Description

CRBIPLOT provides biplot representations of the results from CCA or RDA, showing projections of species or site scores onto one or two environmental variables. By default CRBIPLOT plots the species scores, but you can set option PLOT=sitescores to plot site scores instead.

The type of biplot depends on the scaling method used in the analysis. In RDA, Scaling Type 1 (i.e. no scaling) produces a distance biplot, while Scaling Type 2 (which scales both species and site scores) gives a correlation biplot. Similarly, for CCA, Scaling Type 1 (species scaling) produces a biplot with the sites at the centroids of the species, and Scaling Type 2 (site scaling) plots the species at the centroids of the site.

A distance biplot has the following features:

- distances among elements of Y show approximations of their Euclidean distances in multidimensional space;
- when an element of Y is projected at right angles onto a variable this approximates the position of the object on that variable;
- since the eigenvectors have length one, the length of a projection of an element of Y onto a variable shows its contribution to the formation of that space;
- the angle amongst variables is meaningless.

A correlation biplot has the following features:

- distances among elements of Y are *not* approximations of the Euclidean distances between objects in multidimensional space (so the distance biplot is preferable if you want to interpret relationships amongst the elements of Y);
- when an element of Y is projected at right angles onto a variable this approximates the position of the object on that variable;
- the length of a projection of an element of Y onto a variable shows its contribution to the

formation of that space;
- the angles between variables approximate their correlation.

In addition when we carry out CCA Scaling Type 1:
- distances among sites show approximations in reduced space of their chi-square distances;
- the sites are at the centroids of the species, and the centroids are calculated using weights equal to the relative frequencies of the species (see Makarenkov & Legendre 2002);
- the position of an object on an explanatory variable can be obtained by projecting the objects at right angle on the variable. This scaling is appropriate when the primary interest is the ordination of sites.

With CCA Scaling Type 2:
- it is the distances among species in reduced space that are approximations of their chi-square distances;
- the species are at the centroids of the sites in the graph;
- any species scores that lie close to the point representing an explanatory variable are more likely to be found with higher frequency at that site than others further away (or more likely to be in State '1' with binary data).

This scaling is appropriate when the primary interest is the relationship between species.

The explanatory variables to display can be specified using the X1 and X2 parameters. If the variable is a variate, you can set them to its identifier. Alternatively, if it is either a variate or a variable representing one of the levels of a factor, you can set them to the position of the variable in the list of variables involved in the analysis. Finally, if the variable represents the level of a factor, you can set them to a text containing the label used for the variable in the analysis (you can see the labels by looking at the row labels of the matrix showing the correlations between the environmental variables and the site scores). The DIMENSIONS option lists the numbers of the two canonical axes to plot; default 1,2.

The labels for the species scores, site scores and x-variable(s) can be set using the LMSPECIES, LMSITES and LMXVARIABLES parameters respectively, by selecting one of the following settings:

identifiers	uses the identifiers of the X variates with LMXVARIABLES, or of the Y variates with LMSPECIES (not available with LMSITES),
labels	expects labels to be supplied (in a text) using the LSPECIES, LSITES or LXVARIABLES parameter,
none	gives no labels, and
numbers	uses the column numbers of X and Y.

The defaults are LMSPECIES=numbers, LMSITES=numbers and LMXVARIABLES=identifiers, unless LSPECIES, LSITES or LXVARIABLES is set when the corresponding default becomes labels.

By default CRBIPLOT uses the results from the most recent analysis from RDA or CCA, but you can display results from an earlier analysis by saving the information about the analysis with the SAVE parameter of CCA or RDA, and then providing this to CRBIPLOT using its own SAVE option.

Options: DIMENSIONS, PLOT, WINDOW, SAVE.
Parameters: X1, X2, LMXVARIABLES, LMSPECIES, LMSITES, LXVARIABLES, LSPECIES, LSITES.

Method

CCA and RDA are explained in Chapter 11 of Legendre & Legendre (1998).

References

Legendre, P. & Legendre, L. (1998). *Numerical Ecology, Second English Edition*. Elsevier,

Amsterdam.
Makarenkov, V. & Legendre, P. (2002). Nonlinear redundancy analysis and canonical correspondence analysis based on polynomial regression. *Ecology*, **83**, 1146-1161.

See also

Procedures: CCA, RDA, CRTRIPLOT.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

CRTRIPLOT

Plots ordination biplots or triplots after CCA or RDA (A.I. Glaser).

Options

DIMENSIONS = *scalars*	Which dimensions of the ordinations to display; default 1,2
PLOT = *string token*	What to plot (sitescores, speciesscores, xvariables); default spec, site, xvar
DGROUPS = *string token*	Features to plot for the XGROUPS variate (ellipse, hull, lines, spider); default * i.e. none
DBINARY = *string token*	What to plot for binary variables (biplot, centroid); default bipl
MULTIPLIER = *scalar*	Value to multiply species and environmental variables scores by when plotting RDA; default *, i.e. none chosen
WINDOW = *scalar*	Which graphical window to use; default 1
SAVE = *pointer*	Supplies results from an ordination analysis by CCA or RDA; default uses the most recent analysis

Parameters

LMXVARIABLES = *string tokens*	How to label the x-variables (identifiers, labels, none, numbers); default labe if LXVARIABLES is set, otherwise iden
LMSPECIES = *string tokens*	How to label the species scores (identifiers, labels, none, numbers); default labe if LSPECIES is set, otherwise numb
LMSITES = *string tokens*	How to label the site scores (labels, none, numbers); default labe if LSITES is set, otherwise numb
LXVARIABLES = *texts*	Labels for variables
LSPECIES = *texts*	Labels for species scores
LSITES = *texts*	Labels for site scores
XGROUPS = *variates*, *factors* or *scalars*	X-variate to generate grouping information to appear on the plot (see the DGROUPS option)

Description

CRTRIPLOT plots ordination biplots or triplots following an analysis from either the CCA or RDA procedures. By default it uses the results from the most recent RDA or CCA, but you can display results from an earlier analysis by saving the information about the analysis with the SAVE parameter of CCA or RDA, and then providing this to CRTRIPLOT using its own SAVE option.

An ordination biplot displays the site scores, species scores and biplot scores of environmental variables in a two or three dimensional plot. The site scores are plotted as crosses, the species scores are plotted as dashed arrows. The biplot scores of non-binary variables are represented as full lines. The DBINARY option controls how any binary variables are plotted: they can be represented either by triangles plotted at the centroid of the site scores associated with the value '1', or as arrows showing the biplot scores.

The DIMENSIONS option lists the dimensions of the ordination that you want to use. You can list either two or three of these. The default is a two dimensional plot of dimensions 1 and 2. The PLOT option allows you to control what results are plotted, using the following settings:

sitescores	sites scores,
speciesscores	species scores,
xvariables	biplot scores of the environmental variables.

However, if any of the specified DIMENSIONS is higher than the number of canonical axes, the biplot scores of the environmental variables will not be plotted.

In RDA plots, the species scores and biplot scores of environmental variables are usually much smaller than the site scores. So their values are multiplied by a scalar to make them easier to read. The value is set by the procedure and displayed in the output, but you can set your own multiplier by using the MULTIPLIER option.

You can display additional information for one of the explanatory variables by setting the XGROUPS option either to the identifier of the relevant variate or factor, or to a scalar containing its position in the X pointer (see the X parameter of CCA and RDA). The information that appears is controlled by the DGROUPS option, with settings:

ellipse	draws an ellipse showing an approximate 95% confidence interval for the group centroid (2-dimensional plots only),
hull	draws an enclosing convex hull around the species scores by XGROUPS (2-dimensional plots only),
lines	links the species scores by XGROUPS, and
spider	draws lines from the group centroid to each site score.

The group centroid is the (weighted) group mean of the site scores.

The labels for the species scores, site scores and x-variable(s) can be set using the LMSPECIES, LMSITES and LMXVARIABLES parameters respectively, by selecting one of the following settings:

identifiers	uses the identifiers of the X and Y variates,
labels	expects labels to be supplied (in a text) using the LSPECIES, LSITES or LXVARIABLES parameter,
none	gives no labels, and
numbers	uses the column numbers of X and Y.

The defaults are LMSPECIES=numbers, LMSITES=numbers and LMXVARIABLES=identifiers, unless LSPECIES, LSITES or LXVARIABLES is set when the corresponding default becomes labels.

Options: DIMENSIONS, PLOT, DGROUPS, DBINARY, MULTIPLIER, WINDOW, SAVE.
Parameters: LMXVARIABLES, LMSPECIES, LMSITES, LXVARIABLES, LSPECIES, LSITES, XGROUPS.

Method

CCA and RDA are explained in Chapter 11 of Legendre & Legendre (1998).

Reference

Legendre, P. & Legendre, L. (1998). *Numerical Ecology, Second English Edition*. Elsevier, Amsterdam.

See also

Procedures: CCA, RDA, CRBIPLOT.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

CSPRO

Reads a data set from a CSPro survey data file and dictionary, and loads it into Genstat or puts it into a spreadsheet file (D.B. Baird).

Options

PRINT = *string token*	What to print (catalogue); default cata
FACMETHOD = *string token*	Which factors to create (convertall, keepandconvertall, none, noranges); default keep
MISSINGCODES = *string tokens*	Which special values to convert to Genstat missing values (missing, na); default miss
FVALUESETS = *string token*	Whether form to a set of columns containing all the valueset information (yes, no); default no
SUBITEMS = *string token*	Whether to create a set of columns for the sub-items (yes, no); default no
MERGE = *string token*	Whether to merge the records into a single set of columns all of the same length (yes, no); default no
FUNKNOWNGROUP = *string token*	Whether to create a specific level for values not in the value set, rather than setting them to missing values (yes, no); default no
INCLUDEEXTRA = *string token*	Whether to include a row of column descriptions in the Excel output file after the column heading row (yes, no); default no
WARNONEMPTYGROUPS = *string token*	Whether to warn that groups in a factor are empty and offer to remove them when loading the data from a saved GWB file (yes, no); default no
DUPLICATELABELS = *string token*	What to do with factor groups that have identical labels (combine, ignore, rename); default comb
SCOPE = *string token*	Whether to read the data into global data structures or into data structures local to a procedure calling CSPRO (local, global); default loca
INOPTIONS = *text*	Optional extra input options to be passed to the Dataload.dll
OUTOPTIONS = *text*	Optional extra output options to be passed to the Dataload.dll

Parameters

FILENAME = *text*	Survey data file to be read
DICTIONARY = *text*	Survey dictionary for interpreting the data file
OUTFILENAME = *text*	Name of the output file to be created, if required
SURVEYLEVEL = *scalar*	Level of the survey (1, 2 or 3) to read; default 1
RECORDS = *scalar* or *variate*	Defines the records to be read within the SURVEYLEVEL; by default they are all read
ITEMS = *text*	Names of the survey items to be read
ISAVE = *text* or *pointer*	Saves the identifiers of the columns that are created

Description

CSPRO reads data from a CSPro survey data file and dictionary, specified by the FILENAME and DICTIONARY parameters. If DICTIONARY is not set, CSPRO will look for a file with the same name as FILENAME but with a .dcf extension. You can save the data in either a Genstat workbook (.gwb) or an Excel spreadsheet (.xls), by setting the OUTFILENAME option to the name

of the file to create; if OUTFILENAME is not specifed, a temporary file is used to read into Genstat and this is deleted afterwards. The SAVE parameter can save a pointer containing the structures that have been created.

CSPro surveys can have up to three levels, but most have only a single level. By default, CSPRO reads data from the first level, but you can set the SURVEYLEVEL parameter to read level 2 or 3. The RECORDS parameter specifies which records to read within the SURVEYLEVEL; by default they are all read. The ITEMS parameter can supply a text containing the names of the items to be read. You can append the character ! to the name if you want to force the item to be read as a factor, or # of you want to force it to be read as a variate. This then overrides the setting of the FACMETHOD option for that item (see below).

Setting option SUBITEM=yes allows an item to be broken down. For example, the item ID may be RRVVII (e.g. 120113), with the first two digits giving the region, the next two giving the village, and last two the individual. The sub items RR, VV and II would then also be created as separate columns region, village and individual. Dates are entered like this: e.g. YYYYMMDD with sub-items year, month and day.

By default, each record will have its own set of columns and keys, possibly with different lengths. The keys will be stored in a pointer with element numbers indicating the corresponding records (e.g. Village[1], Village[2] etc.). Alternatively, if you set MERGE=yes, the columns from the different records are merged together based on the common id columns. The merged columns will then all have the same length, with just one set of keys.

A item in a CSPro file can have one or more value sets associated with it. The value set provides mappings from values to groups which are labelled. These are more general than Genstat factors, as either series or ranges of values can be put into a single group: e.g. (1, 3 and 5) or (1 <= x <= 3) Groups can be marked as representing either a missing value or a not-applicable (NA) response. The MISSINGCODES option indicates which of these should be converted into missing values in Genstat; the default is to convert only the groups that represent missing values, and leave the non-applicable groups. By default, values that do not belong to any of the groups defined by a value set are set to missing. However, you can set FUNKNOWNGROUP=yes to create a new level of the resulting factor for these for unallocated values.

The FACMETHOD option controls how the CSPro value sets are converted to factors, using the following settings:

none	no columns are read as factors,
noranges	only columns with single entries per group are read as factors,
keepandconvertall	the original columns are included (as variates) and, in addition, a factor is created for each value set defined for a column, and
convertall	a factor is created for each value set defined for a column, but the original column is not included (so information is lost when groups with series or ranges of values are lumped into a single group).

Note, as mentioned above, the ITEMS parameter can be used to override FACMETHOD for individual survey items.

If you are saving to an Excel file, you can include the column descriptions by setting option INCLUDEEXTRA=yes. If you are saving to a GWB file, you can set option WARNONEMPTYGROUPS=yes to arrange for a dialogue to appear when the file is loaded into the Genstat client, offering to remove any factor groups that have no observations.

CSPro does not insist that each value set item must have a unique label. The DUPLICATELABELS option allows you to choose what to do with any duplicates, by selecting one of the following settings:

`combine`	combines the items into a single group,
`ignore`	ignores duplicate labels, and suppresses the warnings that would occur for duplicate factor labels if the data are saved and then reread from a GWB file,
`rename`	renames the duplicate occurrences by adding a suffix to make the labels are unique.

Setting option `FVALUESETS=yes`, creates an extra set of 7 columns (`Record`, `Item`, `ValueSet`, `From`, `To`, `Label`, `Special`) containing all the valueset information. Record, Item and Valueset are factors giving the names of the record, item and value set for each group, `From` and `To` are variates giving the ranges (or single value if `To` is set to a missing value) for each group, `Label` is a text giving the label for each group, and `Special` is a factor indicating whether the group is a special item (Missing, NA or Other).

When `CSPRO` is used within a procedure, the `SCOPE` option controls whether the structures are created locally in the procedure (default), or globally in the main program.

The options `INOPTIONS` and `OUTOPTIONS` are provided to pass extra input or output options to the `Dataload.dll`. These are only for very specialized use.

Options: `PRINT`, `FACMETHOD`, `MISSINGCODES`, `FVALUESETS`, `SUBITEMS`, `MERGE`, `FUNKNOWNGROUP`, `INCLUDEEXTRA`, `WARNONEMPTYGROUPS`, `DUPLICATELABELS`, `SCOPE`, `INOPTIONS`, `OUTOPTIONS`.

Parameters: `FILENAME`, `DICTIONARY`, `OUTFILENAME`, `SURVEYLEVEL`, `RECORDS`, `ITEMS`, `ISAVE`.

Method

The request is passed to the `Dataload.dll` library which reads the CSPro file and returns the data via a Genstat spreadsheet book.

See also

Directive: `READ`.

Procedures: `IMPORT`, `DBIMPORT`, `SPLOAD`.

Genstat Reference Manual 1 Summary sections on: Input and output, Survey analysis.

CUMDISTRIBUTION

Fits frequency distributions to accumulated counts (R.C. Butler, M.E. O'Neill, P. Brain & H. Turner).

Options

PRINT = *string tokens*	Controls printed output (model, summary, estimates, correlations, fittedvalues, monitoring); default mode, summ, esti
DISTRIBUTION = *string token*	Which distribution to use (normal, logistic, complementaryloglog, acomplementaryloglog, inversenormal, weibull, exponential); default norm
TRANSFORMATION = *string token*	Whether to use log(TIME) if DISTRIBUTION = normal, logistic, complementarylog, or acomplementarylog (log, none); default * uses log except when DISTRIBUTION = inversenormal, weibull or exponential
LAG = *string token*	Type of lag to add to TIME (none, positive, unconstrained); default none
ALLRESPOND = *string token*	If TOTUNITS is set, whether all units are constrained to respond (yes, no); default no
FORM = *string token*	Whether DATA are cumulated or differences (cumulated, differences); default cumu
LOSTUNITS = *string token*	Whether data are left-censored (yes, no); default no
SEPARATE = *string token*	Which parameters to estimate separately for each group (lag, b, m, propn, gamma); default *
POPSEPARATE = *string token*	Which parameters to estimate separately for populations in each group (b, m, lag); default *
PLOT = *string token*	Which graphs to draw (cumulative, density, trcumulative, trdensity); default cumu
MAXCYCLE = *scalar*	Number of iterations for fitting, as in RCYCLE; default 30

Parameters

DATA = *variates* or *pointers*	Specifies the accumulated counts
TIME = *variates* or *pointers*	Defines the time at which each count was recorded
GROUPS = *factors*	Factor indicating groups
INITIAL = *variates*	Initial values for all parameters
IB = *scalars* or *variates*	Initial values for *b*
IM = *scalars* or *variates*	Initial values for *m*
ILAG = *scalars* or *variates*	Initial values for *lag*
IGAMMA = *scalars* or *variates*	Initial values for *gamma*
IPROPN = *scalars* or *variates*	Initial values for proportions
STEPLENGTHS = *variates*	Steplengths for all parameters
SB = *scalars* or *variates*	Steplengths for *b*
SM = *scalars* or *variates*	Steplengths for *m*
SLAG = *scalars* or *variates*	Steplengths for *lag*
SGAMMA = *scalars* or *variates*	Steplengths for *gamma*
SPROPN = *scalars* or *variates*	Steplengths for proportions
TOTUNITS = *scalars* or *variates*	Total number
NPOPULATION = *scalars*	Number of populations (1, 2 or 3); default 1
SAVE = *pointers*	Saves the results

Description

CUMDISTRIBUTION fits frequency distributions to a variate of counts, accumulated over time. The counts are specified by the DATA parameter and the time (t) at which each count is supplied, in a variate, by the TIME parameter. Counts may be accumulated over time (option FORM=cumulated), or be the change in count from the previous time (FORM=difference). Neither the DATA or TIME variate maybe restricted, nor must they contain any missing values. The DATA values must all be non-negative integers.

The form of the cumulative density function is indicated by the DISTRIBUTION option, which has the following settings (z is a function of TIME as defined below).

DISTRIBUTION	cumulative density function
normal	NORMAL($b \times (z-m)$)
complementaryloglog	EXP(-EXP($-b \times (z-m)$))
acomplementaryloglog	1 - EXP(-EXP($b \times (z-m)$))
logistic	1 /(1 + EXP($-b \times (z-m)$)
inversenormal	NORMAL(SQRT(b/z) × (z/m - 1)) + EXP($2b/m$) × (NORMAL(- SQRT(b/z) × (1+z/m)) - 1)
weibull	1 - EXP(-($m \times z$)**b)
exponential	1 - EXP($-m \times z$)

The parameters b and m are estimated, and relate to the distribution of transformed time z as follows.

DISTRIBUTION	Parameter b	Parameter m
normal	1 / sd	mean, t50
logistic	2 × relative response rate at $z=m$	mean, t50
complementaryloglog	relative response rate at $z=m$	mode
acomplementaryloglog	(e-1) × relative response rate at $z=m$	mode
inversenormal	(mean**3) / (sd**2)	mean
weibull	shape	scale
exponential		1/mean

For some of the distributions, TIME may be logged by setting option TRANSFORMATION=log. A lag time before any units respond may be estimated by setting the option LAG=positive. You can set LAG=unconstrained to estimate a negative lag, which assumes that some units responded before TIME=0. These options give z using the following functions of TIME.

	TRANSFORM=none	TRANSFORM=log
LAG=no	*z*=TIME	*z*=LOG(TIME)
LAG=positive or unconstrained	*z*=TIME-LAG	*z*=LOG(TIME-LAG)

The available combinations of LAG and TRANSFORMATION for the various distributions are shown below.

DISTRIBUTION	TRANSFORM	Equivalent distribution	Possible settings for LAG
normal	none		none
	log	log-normal	none, positive, unconstrained
logistic	none		none
	log	log-logistic	none, positive, unconstrained
complementaryloglog	none	Gumbel, Extreme Value1	none
	log	Extreme value2	none, positive, unconstrained
acomplementaryloglog	none		none
	log	Weibull	none, positive, unconstrained
inversenormal	none		none, positive, unconstrained
weibull	none		none, positive, unconstrained
exponential	none		none, positive, unconstrained

TRANSFORMATION is set to log by default for the first four distributions, and none for the last three.

If the total number of units is known, it can be supplied by setting the TOTUNITS parameter. By default, a parameter *gamma*, the proportion of TOTUNITS that can respond, will be estimated. If option ALLRESPOND is set to yes, then *gamma* is fixed at 1 (indicating that all units will respond). If some units were lost before counting began, the number of these can be estimated by setting option LOSTUNITS=yes.

Data for several groups can be fitted together, either by setting DATA to a pointer of variates, or by setting the GROUPS parameter to a factor to identify the different groups. If DATA is set to a pointer, TIME can be set to one variate if all the DATA variates are the same length. Otherwise, it must be set to a pointer with a variate for each DATA variate. Parameters for the groups are constrained to be equal by default, but any of the parameters *b*, *m*, *lag* and *gamma* can be estimated separately between groups by setting the SEPARATE option.

The counts can be from a single population or from a mixture of up to 3 populations, as specified by the NPOPULATIONS parameter (default 1). Parameters *b*, *m* and *lag* can be estimated separately between the populations by setting the POPSEPARATE option. If this is set, the proportion (*propn*) of units in each population will also be estimated. If there are GROUPS in the data, then the proportions can be estimated separately for each group by setting SEPARATE=propn. NPOPULATIONS is the same for each group.

Initial parameter values are estimated within the procedure, but can be supplied separately using any of the parameters IB, IM, ILAG, IGAMMA and IPROPN, or in one list using the INITIAL parameter. If any parameter is to be estimated separately between GROUPS or populations, there must be one initial value for each parameter of that type to be estimated. For example, if there are two groups, and SEPARATE=m, then IM should be set to a variate of length 2. If INITIAL is set, its values will be used even if the other initial value parameters are set. The values in INITIAL must be in the order *b*, *m*, *lag*, *gamma*, *propn*, with enough values for the number of each being estimated. For *propn*, there must be 1 less than NPOPULATIONS. For example, with 2 groups and 3 populations, with SEPARATE=b,m and POPSEP=m there will be 2 initial values for *b* and 6 for *m* with two for *propn*. Steplengths for the fitting process can be supplied similarly using STEPLENGTHS or SB, SM, SLAG, SGAMMA, SPROPN. MAXCYCLE controls the maximum number of iterations, as in the RCYCLE directive.

Output is controlled by the PRINT option, with settings as in FITNONLINEAR. Parameter estimates are indexed by groups and/or population numbers, with group labels first if both populations and groups are used. If PRINT=estimates, parameters calculated from the fitted parameters (mean, sd, t50) are also printed. Option PLOT determines the form of the graphical output:

cumulative	fitted curve and cumulated counts,
density	differenced fitted curve and counts,
trcumulative	trellis version of cumulative when there are GROUPS,
trdensity	trellis version of density when there are GROUPS.

Setting PLOT=* suppresses all graphs).

Some results can be saved using RKEEP (as with FIT). Further results can be saved by setting the SAVE parameter. This creates a pointer with three sections labelled by their contents. SAVE['Data'] points to the columns used in the fitting process:

ndata	the (differenced) counts,
ntime	times for each count,
groups	grouping factor,
fitted	fitted values,
cumdata	cumulated counts,
cumfitted	cumulated fitted values,
z	transformed time variate (as above).

SAVE['CalcParams'] contains the calculated parameters and their standard errors (Mean, Sd, T50, seMean, seSd, seT50). SAVE['Viable'] contains the estimated number of viable units (Nv) for each group and, if NPOP>1, the number in each population (PopNv).

Options: PRINT, DISTRIBUTION, TRANSFORMATION, LAG, ALLRESPOND, FORM, LOSTUNITS, SEPARATE, POPSEPARATE, PLOT, MAXCYCLE.

Parameters: DATA, TIME, GROUPS, INITIAL, IB, IM, ILAG, IGAMMA, IPROPN, INITIAL, IB, IM, ILAG, IGAMMA, IPROPN, STEPLENGTHS, SB, SM, SLAG, SGAMMA, SPROPN, TOTUNITS, NPOPULATION, SAVE.

Method

This procedure extends the methods described by Brain & Butler (1988). If FORM=cumulated, the DATA vector is differenced, and if DATA is set to a pointer, the DATA variates are stacked, and

a factor created to identify the groups. The resulting data variate is then used with FITNONLINEAR. The model to be fitted is set up in a pointer to expressions formed according to the settings of the various options and parameters.

Action with RESTRICT
Because the calculations in the procedure involve differencing the counts, the TIME and DATA variates must not be restricted.

Reference
Brain, P. & Butler, R.C. (1988). Cumulative count data. *Genstat Newsletter*, **22**, 38-47.

See also
Directive: DISTRIBUTION.
Procedure: RSURVIVAL.
Genstat Reference Manual 1 Summary sections on: Repeated measurements, Survival analysis.

CVAPLOT

Plots the mean and unit scores from a canonical variates analysis (D.A. Murray).

Options

PLOT = *string tokens*	Type of plot to be drawn (meanscores, unitscores, confidenceregion); default mean, conf
GROUPS = *factor*	Group allocations in the CVA
MSCORES = *matrix*	Mean scores from the CVA; if unset these are calculated using the CVA directive
USCORES = *matrix*	Unit scores from the CVA; if unset these are calculated using the CVASCORES procedure
WSSPM = *SSPM*	Within-group sums of squares and products, means etc. for the CVA; must be supplied if the scores and groupings are not provided
CREGION = *string tokens*	Type of confidence region to be drawn (mean, population); default mean
CIPROBABILITY = *scalar*	The probability level for the confidence region; default 0.95
TAREA = *scalar*	Defines the transparency to use to shade the confidence regions; default 255 i.e. no shading

Parameters

YDIMENSION = *scalars*	Dimensions to be plotted in the y direction of each graph
XDIMENSION = *scalars*	Dimension to be plotted in the x direction
TITLE = *texts*	Title for each plot
WINDOW = *scalars*	Window for each graph; default 1
SCREEN = *string tokens*	Whether to clear the screen before plotting (clear, keep); default clea

Description

CVAPLOT plots information from a canonical variates analysis. The type of graph to be displayed is controlled by the PLOT option with settings meanscores to draw mean scores, unitscores to display the unit scores and confidenceregion to display confidence regions about the means or the tolerance region for a population. The CREGION option specifies the type of confidence region that is drawn. The setting mean will draw the confidence region about the population means, and population plots the tolerance region for the populations. By default a 95% confidence region is calculated, but this can be changed by setting the CIPROBABILITY option to the required value (between 0 and 1). You can shade the confidence regions by setting the TAREA option. This defines a transparency value (between 0 and 255) for the shaded regions, in a similar way to the TAREA option of PEN. The default value of 255 indicates that the regions are completely transparent (i.e. completely unshaded); a line is then drawn around each region.

Matrices containing the mean scores and units scores can be supplied directly, using options MSCORES and USCORES respectively, and option GROUPS can supply a factor defining the groupings of the units in the canonical variates analysis. Alternatively, you can supply a within-group SSPM and the scores will be calculated within the procedure, using the CVA directive and the CVASCORES procedure, and the groups will be accessed from within the SSPM.

The YDIMENSION and XDIMENSION parameters specify which dimensions are to be plotted in the y and x directions; by default these are dimensions 1 and 2 respectively. The WINDOW parameter indicates the window to be used for each plot (default 1), the TITLE parameter provides a title for each plot, and the SCREEN parameter indicates whether existing plots on the screen are to be kept or cleared each time (the default being to clear the screen).

Options: PLOT, GROUPS, MSCORES, USCORES, WSSPM, CREGION, CIPROBABILITY, TAREA.
Parameters: YDIMENSION, XDIMENSION, TITLE, WINDOW, SCREEN.

Method
The CVA directive and the CVASCORES procedure are used to calculate the scores if necessary. A two dimensional representation of the results of the CVA is then plotted on the current high resolution graphics device. The 95% confidence region of the group means is calculated by circles of radius

```
SQRT( EDCHISQUARE( CIPROBABILITY; 2 ) / n )
```

about the means and the tolerance region of the populations is calculated by

```
SQRT( EDCHISQUARE( CIPROBABILITY; 2 ) )
```

(see Krzanowski 1988, page 374).

Reference
Krzanowski, W.J. (1988). *Principles of Multivariate Analysis*. Oxford University Press, Oxford.

See also
Directive: CVA.
Procedures: CVASCORES, DBIPLOT.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

CVASCORES

Calculates scores for individual units in canonical variates analysis (S.A. Harding).

Option

PRINT = *string tokens*	What output to print (scores, adjustments); default scor

Parameters

WSSPM = *SSPMs*	Within-group sums of squares and products structure
LRV = *LRVs*	Loadings, roots and trace saved from CVA of the WSSPM
SCORES = *matrices*	Unit scores
ADJUSTMENTS = *matrices*	Mean Adjustments

Description

CVASCORES calculates coordinates of the individual data points projected into the canonical variate space of a canonical variates analysis. This provides data that may be plotted with the canonical variate means to provide more informative graphics.

The WSSPM parameter must be set to the within-group SSP matrix that was used as input to the CVA directive when calculating the analysis, and the LRV parameter must supply the LRV structure formed by CVA. The scores can be saved using the SCORES parameter, and the mean adjustments (as printed by CVA) can be saved using the ADJUSTMENTS parameter. The PRINT option allows the scores and adjustments to be printed, with the default to print just the scores.

Option: PRINT. Parameters: WSSP, LRV, SCORES, ADJUSTMENTS.

Method

The data matrix X is rotated using the canonical variate loadings, L, to form $Y=XL$. The columns of Y then have the mean adjustments subtracted, to match the location of the canonical variate means, which are translated such that their their weighted centroid is at the origin. As the procedure needs to compute the mean adjustments, these can also be saved as they are unavailable from CVA itself.

See also

Directive: CVA.
Procedure: CVAPLOT.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

DARROW

Adds arrows to an existing plot (D. B. Baird).

Options

`WINDOW` = *scalar*	Window number for the graphs; default 3
`COORDINATETYPE` = *string token*	Type of coordinate to use for the locations of the arrows (`frame`, `graph`); default `grap`
`YUPPER` = *scalar*	Maximum vertical coordinate in the frame; default 1
`XUPPER` = *scalar*	Maximum horizontal coordinate in the frame; default 1
`ISTYLE` = *string token*	The type of symbol at the start of the arrow (`none`, `open`, `closed`, `circle`); default `none`
`ESTYLE` = *string token*	The type of symbol at the end of the arrow (`none`, `open`, `closed`, `circle`); default `open`
`ISIZE` = *scalar*	The size of the symbol at the start of the arrow; default 1
`ESIZE` = *scalar*	The size of the symbol at the end of the arrow; default 1
`IANGLE` = *scalar*	The angle in degrees of the starting arrowhead when `ISTYLE` is `open` or `closed`; default 45
`EANGLE` = *scalar*	The angle in degrees of the ending arrowhead when `ESTYLE` is `open` or `closed`; default 45
`LAYER` = *scalar*	The plot layer for the arrows; default is a new layer above the previous plot items

Parameters

`IY` = *variates, scalars* or *factors*	The starting y-positions of the arrows
`IX` = *variates, scalars* or *factors*	The starting x-positions of the arrows
`EY` = *variates, scalars* or *factors*	The ending y-position of the arrows
`EX` = *variates, scalars* or *factors*	The ending x-position of the arrows
`COLOUR` = *variates, scalars*, *texts* or *factors*	Colour of the arrows; default `'black'`
`LINESTYLE` = *variates, scalars* or *factors*	Linestyle of the line in the arrows; default 1
`THICKNESS` = *variates, scalars* or *factors*	Thickness of the line in the arrows; default 1
`TRANSPARENCY` = *variates, scalars* or *factors*	Transparency of the arrows; default 0

Description

`DARROW` adds arrows or lines to existing plots. The coordinates defining the start and end points of the arrows are specified by the `IY`, `IX`, `EY` and `EX` parameters. The `COLOUR`, `LINESTYLE`, `THICKNESS` and `TRANSPARENCY` parameters specify the colour, linestyle, thickness and transparency (`0` = opaque - `255` = completely transparent) of each arrow. These can supply a single value, if all the arrows are to have the same attribute; otherwise, they should supply a structure of the same length as the `IY` vector.

The `WINDOW` option specifies the number of the window containing the existing plot. By default, the points defining the arrows are specified in terms of the x- and y-axes in the plot. However, you can set option `COORDINATETYPE=frame` to define the points relative to the frame. The maximum size of the frame is then defined by the `XUPPER` and `YUPPER` options.

The `ISTYLE` and `ESTYLE` options control the symbols at the start and end of the arrow, respectively. Similarly, the `ISIZE` and `ESIZE` options define the symbol sizes. The `IANGLE` and `EANGLE` options control the angle between the two sides at the start and end of the arrowheads. Setting the angles to values greater than 180 (e.g. 315) reverses the direction of the arrowheads.

The LAYER option controls which of the existing items in the plot will be overlaid by the arrows. By default, they overlay all the previous items.

Options: WINDOW, COORDINATETYPE, YUPPER, XUPPER, ISTYLE, ESTYLE, ISIZE, ESIZE, IANGLE, EANGLE, LAYER.
Parameters: IY, IX, EY, EX, COLOUR, LINESTYLE, THICKNESS, TRANSPARENCY.

Action with RESTRICT
Restrictions are ignored.

See also
Procedures: DERRORBAR, DFRTEXT, DREFERENCELINE, DTEXT.
Genstat Reference Manual 1 Summary section on: Graphics.

DAYLENGTH

Calculates daylengths at a given period of the year (R.J. Reader & K. Phelps).

Option

LATITUDE = *scalar*	Latitude at which the daylength is to be calculated, positive for northern hemisphere and negative for southern hemisphere; default 52.205 N (Wellesbourne)

Parameters

DAYNUMBER = *variate*	Days of year for which daylengths are required
DAYLENGTH = *variate*	Calculated daylengths in hours

Description

DAYLENGTH calculates a set of daylengths at a given latitude. The numbers of the days during the year for which the daylengths are required should be specified, in a variate, using the DAYNUMBER parameter. The lengths will then be stored in the variate specified by the DAYLENGTH parameter. The latitude is defined by the LATITUDE option, by default LATITUDE=52.205 which is the latitude of Wellesbourne.

Option: LATITUDE. Parameters: DAYNUMBER, DAYLENGTH.

Method

The formula by which the daylengths is calculated is given in Sellers (1965).

Action with RESTRICT

If either the DAYNUMBER or the DAYLENGTH variate is restricted, the calculations will be done only for the units not excluded by the restriction.

Reference

Sellers W.D. (1965). *Physical Climatology*. University of Chicago Press, Chicago, Illinois.

See also

Procedure: HEATUNITS.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

DBARCHART

Produces bar charts for one or two-way tables (A.R.G. McLachlan & R.C. Butler).

Options

TITLE = *text*	Title for the chart; no default
WINDOW = *scalar*	Window for the chart; default 1
KEYWINDOW = *scalar*	Window for the key, no key is produced for one-way tables; default 2
LABELS = *text*	Labels for clusters of bars; by default the labels or levels of the first classifying factor of TABLE are used
APPEND = *string token*	Whether to append bars (no, yes); default no
SCREEN = *string token*	Whether to clear screen before displaying chart (keep, clear); default clea
KEYDESCRIPTION = *text*	Title for key; default is the name of the second factor of TABLE
YSCALE = *expression structure*	Defines a transformation of the data, the expression must be a function of either Y or X, for example !e(log(X)), and should be valid for the range of the data in TABLE; default no transformation
BELOWORIGIN = *string token*	Whether to include values in TABLE less than ORIGIN (omit, include); default omit
ORIENTATION = *string token*	Direction of the plot (horizontal, vertical); default vert
BARCOVERING = *scalar*	What proportion of the space allocated along the x-axis each bar should occupy; default * gives proportion 0.8 (thus giving a gap between each bar or each group of bars)
XPOSITION = *string token*	Position of the x-axis on the y-axis (lower, origin); default lowe
OMITEMPTYLEVELS = *string token*	Whether to omit levels where there are only missing values (yes, no); default no

Parameters

TABLE = *tables*	One or two-way table of data
ORIGIN = *scalars*	Origin for y-axis; default 0
PEN = *variates* or *scalars*	Pen (or pens) to use; default is !(1 ... nlevel(last_classifying_factor))
DESCRIPTION = *texts*	Annotation for Key for two-way tables; default uses the labels or levels of the factor that is not being used as the XFACTOR
YMARKS = *variates*	Position of the tick-marks on the y-axis
XFACTOR = *factors*	X-axis factor for a 2-way TABLE; default first factor of TABLE
LOWERERRORBARS = *tables*, *variates* or *scalars*	Lower bounds of the error bars on the y-axis
UPPERERRORBARS = *tables*, *variates* or *scalars*	Upper bounds of the error bars on the y-axis
YERRORBARS = *tables*, *variates* or *scalars*	Y-axis position of any error bar symbols; by default no symbols are plotted
XERRORBARS = *tables*, *variates* or *scalars*	X-axis position of the error bars; default midpoints of

bar-chart bars

PENERRORBARS = *tables*, *variates* or *scalars*
: Pen (or pens) to use for plotting error bars; default 1

Description

DBARCHART produces bar charts for one or two-way tables. A transformation can be specified by the option YSCALE, in which case the data and y-axis are rescaled, but the y-axis is labelled on the original scale of the data. The table is specified by the TABLE parameter and the origin of the y-axis, which need not be zero, can be set with the ORIGIN parameter or with XAXIS. The PEN parameter specifies a pen, or pens, for the bars of the histogram. This can be input as a scalar if the same pen is to be used for the whole plot, or as a variate to allow the groups to be drawn in different pens; by default pens 1, 2 ... are used for the successive bars. Positions of the tick-marks on the y-axis can be specified with the YMARKS parameter or with YAXIS. Upper and lower limits for the y-axis can be defined as normal with a YAXIS statement before DBARCHART is used. Bar charts of two-way tables are by default produced with the first factor defining the groups of bars in the chart, and the second the bars within each group. The factor used for the x-axis can be specified by the XFACTOR parameter; this is ignored for one-way tables. Labelling for the key can be supplied by the DESCRIPTION parameter; if this is not set, DBARCHART uses the labels of the classifying factor that is not being used as the XFACTOR.

To plot error bars attached to the ends of the bars of the bar chart, one or both of UPPERERRORBARS and LOWERERRORBARS can be set to specify the upper and/or lower bounds of the error bars. By default, the x-axis coordinates of the error bars are the midpoints of the bar-chart bars. XERRORBARS can be used to specify different x-axis positions for the error bars. As a guide, the tickmarks on the x-axis have coordinates: 1 ... number of levels of first factor of TABLE. If errors occur with the error-bar parameters, an error message will be given and, where possible, the bar chart will still be drawn but without error bars. UPPERERRORBARS and LOWERERRORBARS must each be a table with factors that match that of TABLE or, if XERRORBARS is set, they must match the structure of XERRORBARS.

The colour, line style, line thickness, and cap width of the error bars is controlled by the definition of the pens specified by the PENERRORBARS parameter. If PENERRORBARS is not set, or there is an error with the PENERRORBARS setting, a default of PENERRORBARS=1 is used. The settings of the pen used for plotting the error bars are COLOUR (or CLINE), LINESTYLE, BARTHICKNESS, and BARCAPWIDTH, with the exception that when YERRORBARS is set, any SYMBOL and LABELS (and related) settings from the pen are used also. The YERRORBARS parameter is not necessary for the plotting of the actual error bars, but instead is used to specify a y-axis coordinate for a symbol or label to be associated with each error bar. The symbol and any label text associated with PENERRORBARS will be plotted at the (x, y) position given by (XERRORBARS, YERRORBARS). If YERRORBARS is not set, no symbols or labels are plotted regardless of the PEN settings given by PENERRORBARS.

The options of the procedure mainly control the plotting: the windows that are used for the plot (WINDOW) and for the key (KEYWINDOW), titles for the graph (TITLE) and for the key (KEYDESCRIPTION), whether the groups of bars are appended or placed side-by-side (APPEND), whether or not to clear the screen before plotting (SCREEN), and the proportion of x-axis space allocated to bars (BARCOVERING). Normally values in TABLE less than ORIGIN are omitted but, if you set BELOWORIGIN=include, bars will be drawn for these values extending below the x-axis. In that case, the x-axis will be placed at the bottom of the y-axis (XPOSITION=lower) but the axis can be placed at the origin if you set XPOSITION=origin. The ORIENTATION option controls whether the bars of the histogram are plotted vertically (the default) or horizontally. When ORIENTATION=horizontal, the horizontal axis is taken to be the y-axis, so the same XAXIS and YAXIS settings can be used however the histogram is oriented.

The default is to plot all TABLE factor levels. However, any levels that contain only missing

values can be excluded from the plot by setting option OMITEMPTYLEVELS=yes.

Unless you have set them previously by using YAXIS, the y-axis limits will be set to accommodate the error bars but with the following limitation: the axis lower limit will not be lowered below the origin (even if BELOWORIGIN=include) unless at least one of the bar-chart bars is also below the origin (this is a limitation of DHISTOGRAM). The x-axis limits are automatically set to include all of the bar-chart bars with a margin of 5% at each end of the x-axis. The amount of space at the end of the x-axis (default: 5% of x-axis data range) can be set using the MLOWER% and MUPPER% settings of XAXIS before using DBARCHART. The x-axis limits are not adjusted for the x-axis placement of error bars, so any error bars given an XERRORBARS value outside the default x-axis limits will not be plotted unless the limits are increased by using XAXIS before using DBARCHART. However, if any previously specified x-axis limits are too narrow for the plotting of the bar-chart bars, the limits will be increased to fit the bars.

Options: TITLE, WINDOW, KEYWINDOW, LABELS, APPEND, SCREEN, KEYDESCRIPTION, YSCALE, BELOWORIGIN, ORIENTATION, BARCOVERING, XPOSITION, OMITEMPTYLEVELS.
Parameters: TABLE, ORIGIN, PEN, DESCRIPTION, YMARKS, XFACTOR, LOWERERRORBARS, UPPERERRORBARS, YERRORBARS, XERRORBARS, PENERRORBARS.

Method

If YSCALE is set, the expression is used to transform TABLE and ORIGIN. Any YMARKS are also transformed to find the position of the tick-marks. TABLE is then rescaled so that the ORIGIN is zero. The y-axis is set up with the labelling on the original scale of TABLE. Two-way tables are first split into one-way tables classified by the second factor of TABLE. One sub-table is produced for each level of the first factor of TABLE. The chart is then produced with a single DHISTOGRAM statement for all sub-tables. YSCALE is imported into the program by setting Y or X as a dummy, and printing the expression into a text. A new expression is then set up using this text with the EXECUTE directive. Y (or X) is then set to ORIGIN, TABLE and YMARKS in turn before the expression is calculated. When error bars are given, the UPPERERRORBARS, LOWERERRORBARS and YERRORBARS values are scaled, if necessary,using YSCALE. Then, the error bars are plotted on top of the bar chart by using DGRAPH with the same FRAME, XAXIS and YAXIS settings as used for the bar chart. When there are bar-chart values to be plotted below the origin (BELOWORIGIN=include) and XPOSITION=lower, the bar chart is first drawn without an x-axis, then an x-axis is plotted separately at the lower end of the y-axis, and finally, by using DGRAPH, a plain line is drawn parallel to the x-axis at the origin.

See also

Directives: BARCHART, HISTOGRAM, LPHISTOGRAM.
Genstat Reference Manual 1 Summary section on: Graphics.

DBCOMMAND

Runs an SQL command on an ODBC database, PC Windows only (D.B. Baird).

Options

`WARNINGDIALOGS` = *string token*	Whether dialogs giving ODBC error and warning messages are presented (`display`, `omit`); default `disp`
`DRIVER` = *scalar*	Driver version (either 32 or 64) to use with the 64-bit version of Genstat; default 64

Parameters

`COMMAND` = *texts*	Specifies SQL commands to run on the database
`DB` = *texts*	Database connection string for each command
`GDBFILE` = *texts*	Name of GDB file to be used in specifying the database for each command
`EXIT` = *scalars*	The exit code (0=success, 1=failure) from each command

Description

`DBCOMMAND` runs a SQL command on an ODBC database. SQL commands like `CREATE TABLE`, `DROP TABLE`, `ALTER TABLE`, `INSERT INTO` and `DELETE FROM` can be used to modify the database. However, the command cannot have parameters or return data. The `EXIT` parameter can specify a scalar to save a code indicating whether the command was successful (0) or failed (1). If the command fails, an SQL error message will be printed.

The ODBC database can be specified by using the `DB` parameter to supply a text containing a database connection string. Note that any file names in the string must use `\\` rather than `/` for the directory separator: i.e. the file name

```
C:\WORK\MYDATA.MDB
```

must be specified as

```
C:\\WORK\\MYDATA.MDB
```

rather than as

```
C:/WORK/MYDATA.MDB
```

Alternatively, you can use the `GDBFILE` parameter to specify an existing GDB file that contains an ODBC query. This can be created in Genstat *for Windows* using the ODBC Data Query menu (accessible from the New option of the Spread menu on the menu bar). The DSN line in this text file can be used to connect to the same database as specified by the `DB` parameter.

You can set option `DRIVER=32` to use 32-bit ODBC drivers when you are running the 64-bit Genstat.

Options: `WARNINGDIALOGS`, `DRIVER`.
Parameters: `COMMAND`, `DB`, `GDBFILE`, `EXIT`.

Method

The SQL command is sent to the `ODBCLOAD.DLL` library which runs the command and returns an exit code and any error messages.

See also

Procedures: `DBEXPORT`, `DBIMPORT`, `DBINFORMATION`, `IMPORT`.
Genstat Reference Manual 1 Summary section on: Input and output.

DBEXPORT

Update data in an ODBC database table using Genstat data, PC Windows only (D.B. Baird).

Options

METHOD = *string token*	Type of update on table (create, insert, merge); default crea
ROWMERGEMETHOD = *string token*	For METHOD=merge, what action to take when rows do not match any in the existing table (none, matched, all); default all
COLMERGEMETHOD = *string token*	What to do with unmatched columns (add, omit); default add
OMIT = *string token*	Which rows to omit from the data for METHOD settings other than merge (none, restricted); default rest
ERRORACTION = *string token*	What to do when any non-fatal errors occur, (continue, stop); default stop
WARNINGDIALOGS = *string token*	If any errors occur, pop up warning dialogs (display, omit); default disp
GLKFILE = *text*	Name of existing Genstat ODBC Update link file (*.GLK) to use
DRIVER = *scalar*	Driver version (either 32 or 64) to use for the 64-bit version of Genstat; default 64
ODBCPATH = *text*	Path for the folder containing the executable program (Odbcload.exe) used by the 64-bit version of Genstat to export the data when DRIVER=32; default is the folder containing the Genstat executable program

Parameters

DATA = *pointer* or *text*	Pointer to a compatible set of data structures to add to the table or text with a name of an existing Genstat spreadsheet file containing data to be added
DB = *text*	Database connection string specifying the ODBC database to connect to
TABLENAME = *text*	Name of the table in the ODBC database (if METHOD is set to insert or merge, then this must already exist in the database)
COLUMNNAMES = *text*	Names of the columns in the table to be updated; if this is not provided, it will be assumed that the columns in the table have the same names as the Genstat data structures
SUBSET = *variate* or *text*	Column numbers or names of the subset of data columns (only if a pointer is used for the DATA parameter) to be added to the table; if SUBSET is not set, all columns are added to the table
MATCH = *variate*	Numbers of the columns in the table to be matched with the column in the table (the names are provided by WITH)
WITH = *text*	Names of the columns in the table to be matched with the Column; if this not provided, it is assumed that these columns have the same names as those of the Genstat data structures

Description

DBEXPORT can be used to add either a new table to an ODBC data source (METHOD=create), add rows to an existing table (METHOD=insert), or update rows in an existing table (METHOD=merge).

The form of the DB connection string can be found by saving a ODBC Query in the Genstat client in a GDB file (using the Spread > New > ODBC Query menu in Genstat *for Windows*) and then examining this file with a text editor. The second line contains the database connection string.

The data to be sent can either be specified as a pointer to a set of structures in Genstat or a text giving a Genstat spreadsheet (GSH) file. The DATA parameter need not be set if a GLKFILE is specified, as this may point to an existing GSH file. If a GLKFILE is provided, all options and parameters will be taken from this, with the exception that a different DATA set and/or TABLENAME can be provided and this will be used with the existing parameters from the GLKFILE. A GLKFILE can be created using the Spread > Export menu items and using the Save Export Link option in these menus.

The column names within the ODBC table are assumed to be the same as the Genstat identifiers, unless you specify COLUMNNAMES and WITH (for matching with MATCH).

If COLMERGEMETHOD=omit, any columns in the data not found in the database table will be omitted; otherwise new columns will be added to the existing table. The SUBSET parameter can be set to pick a subset of columns from an existing GSH file. However if DATA is set to a pointer, it would be normal to only form this to contain only the elements that you wanted updated in the table, instead of using the SUBSET parameter.

If METHOD=merge, the MATCH parameter must be set. At most only five columns can be matched. The WITH parameter may be set if the columns in the table do not have the same names as the structures used in the DATA parameter. The ROWMERGEMETHOD option controls how unmatched rows are handled in a merge: the setting none does not add unmatched rows, the setting matched only adds a row if another with the same matching criteria already existing in the table, and all adds in all unmatched rows into the table.

If the WARNINGDIALOGS option is set to display, message boxes will pop up on the windows desktop detailing any errors; the setting omit suppresses the warning messages. The Genstat server will wait until the user clicks OK on these, so this will halt any processing, and is better not used in batch jobs. If option ERRORACTION=stop, any warnings (such as not being able to add missing values into a column or not being able to add rows with duplicate ID's) will cause the update to stop; otherwise all valid data will be added to the table, unless a fatal error occurs.

The ODBCPATH option specifies the path for the folder containing the executable program (Odbcload.exe) used by the 64-bit version of Genstat to export the data when option DRIVER=32. In the 16th Edition, the executable should already be installed the folder containing the Genstat executable program, which is the default setting. So this option should not need to be set. There is more information about using 32-bit ODBC drivers with 64-bit Genstat on the VSN website www.vsni.co.uk.

(Note: DBEXPORT replaces the procedure %ODBCUPDATE from earlier editions of Genstat.)

Options: METHOD, ROWMERGEMETHOD, COLMERGEMETHOD, OMIT, ERRORACTION, WARNINGDIALOGS, GLKFILE, DRIVER, ODBCPATH.
Parameters: DATA, DB, TABLENAME, COLUMNNAMES, SUBSET, MATCH, WITH.

Method

The structures in DATA are saved to a GSH file using FSPREADSHEET. A GLK file is built using the supplied parameters or an existing GLK file, and then this is passed to the ODBCLOAD.DLL library to be processed.

Action with `RESTRICT`
Restrictions on the structures are obeyed if `OMIT=restricted`, otherwise they are ignored. If the restrictions on the structures are not consistent, a fault will occur.

See also
Procedure: `DBCOMMAND`, `DBIMPORT`, `DBINFORMATION`, `EXPORT`.
Genstat Reference Manual 1 Summary section on: Input and output.

DBIMPORT

Loads data from an ODBC database, PC Windows only (D.B. Baird).

Options

PRINT = *string token*	What information to print (catalogue); default cata
OUTTYPE = *string token*	Whether to form a Genstat command file or spreadsheet file as output (GEN, GSH, GWB); default GWB
METHOD = *string token*	Whether to load data into the Genstat server after creating the file, or merely to create the file, or to run a command with no output (create, load, command); default load
IMETHOD = *string token*	Whether to read the column names from the first row of data, or to use default column names (read, supply, none, default); default read
ENDSTATEMENT = *string token*	Ending statement to use in a GEN output file (RETURN, ENDBREAK); default RETURN
WARNINGDIALOGS = *string token*	Whether dialogs giving ODBC error and warning messages are presented (display, omit); default disp
DRIVER = *scalar*	Driver version (either 32 or 64) to use for the 64-bit version of Genstat; default 64
ODBCPATH = *text*	Path for the folder containing the executable program (Odbcload.exe) used by the 64-bit version of Genstat to load the data when DRIVER=32; default is the folder containing the Genstat executable program
NROWSFETCH = *scalar*	Number of rows to fetch per driver transaction; default 40

Parameters

DB = *text*	Database connection string
SQL = *text*	SQL Query string to run against the ODBC database
GDBFILE = *text*	Name of GDB file to be used in reading from ODBC database
OUTFILE = *text*	Output file to be created; if this is not provided a temporary file will be created, and then deleted if the data is loaded
COLUMNS = *text*	Names and/or type codes for the columns read (the type of column can be forced by ending the column name, if supplied, with the code ! for a factor, # for a variate, and $ for a text)
ISAVE = *pointer*	Name of a pointer to save the column identifiers
NROWSALLOCATE = *scalars*	Specifies how many rows to allow space for, in the initial allocation of memory, before the data are read; default 1000

Description

This procedure runs an SQL command against an ODBC database and returns the data as a set of Genstat structures. The COLUMNS parameter can be used to set the names and types of the structures or to receive back a pointer to the structures created. You can force the type of column by ending the column name with the code ! for a factor, # for a variate, and $ for a text. For example

```
COLUMN=!T('Trt!','ID$','Rank#')
```

will create a factor called Trt, a text called ID and a variate called Rank. If only the type code is provided, the columns will not be renamed, but the new types will set, e.g.

```
COLUMN=!T('!','$','#')
```

will force the first three columns to be of type factor, variate and text respectively. A column name ending with an underscore (_) will also be converted to a factor in Genstat.

Either an existing GDB file is used which contains an ODBC query, or the texts supplied by the DB and SQL parameters are used to specify the ODBC query. The GDB file can be created using the Spread > New > ODBC Data Query menu. The DSN line in this text file can be used to connect to the same database as specified by the DB parameter with ad hoc queries specified with the SQL parameter.

Note that any file names in the DB connection string will need to use \\ rather than / for the directory separator, i.e. the file name C:\WORK\MYDATA.MDB would need to be given in Genstat as 'C:\\WORK\\MYDATA.MDB' rather than as C:/WORK/MYDATA.MDB.

The NROWSFETCH option allows you to specify the number of rows to fetch in each driver transaction (default 40). Fetching several at once saves time, but requires more memory. You can also save time by using the NROWSALLOCATE parameter to pre-allocate space in memory. Currently memory for 1000 rows is allocated initially, and this extended by 1000 rows whenever it is exhausted. Setting ROWSALLOCATED make this more efficient and more likely to succeed when the transfer uses more than half the available RAM (as the extension is then likely to fail).

The ODBCPATH option specifies the path for the folder containing the executable program (Odbcload.exe) used by the 64-bit version of Genstat to export the data when option DRIVER=32. In the 16th Edition, the executable should already be installed the folder containing the Genstat executable program, which is the default setting. So this option should not need to be set. There is more information about using 32-bit ODBC drivers with 64-bit Genstat on the VSN website www.vsni.co.uk.

(Note: DBIMPORT replaces the procedure ODBCLOAD from earlier editions of Genstat.)

Options: PRINT, OUTTYPE, METHOD, IMETHOD, ENDSTATEMENT, WARNINGDIALOGS, DRIVER, ODBCPATH, NROWSFETCH.
Parameters: DB, SQL, GDBFILE, OUTFILE, COLUMNS, ISAVE. NROWSALLOCATE.

Method

The SQL query is sent to the ODBCLOAD.DLL library which runs the query and saves the results in a temporary GWB file. This is then loaded using the SPLOAD directive.

Action with RESTRICT

Restrictions are not applicable to any of the parameters.

See also

Procedure: DBCOMMAND, DBEXPORT, DBINFORMATION, IMPORT.
Genstat Reference Manual 1 Summary section on: Input and output.

DBINFORMATION

Loads information on the tables and columns in an ODBC database, PC Windows only (D.B. Baird).

Options

PRINT = *string token*	What to print (information); default info
INFORMATION = *string token*	What information to read from the database (tables, columns); default tabl
DRIVER = *scalar*	Driver version (either 32 or 64) to use with the 64-bit version of Genstat; default 64

Parameters

DB = *texts*	Database connection string
GDBFILE = *texts*	GDB file specifying an ODBC query
ISAVE = *pointers*	Specifies pointers to save the information

Description

This procedure allows you to print or save information about the tables and columns in an ODBC database. The ISAVE parameter can specify a pointer to save the information, and the INFORMATION option controls what is saved. If INFORMATION=tables, a single text structure ISAVE['Table'] is saved. Alternatively, if INFORMATION=columns, three text structures named ISAVE['Table'], ISAVE['Column'] and ISAVE['Type'] are saved. These contain the table names, the column names and the column type (either Numeric, Text, Date, Time, Binary or Unknown), respectively. By default the information is printed, but you can set option PRINT=* to suppress this.

The ODBC query can be specified by using the DB parameter to supply a text containing a database connection string. Note that any file names in the string must use \\ rather than / for the directory separator: i.e. the file name

```
C:\WORK\MYDATA.MDB
```

should be specified as

```
C:\\WORK\\MYDATA.MDB
```

rather than as

```
C:/WORK/MYDATA.MDB
```

Alternatively, you can use the GDBFILE parameter to specify an existing GDB file that contains an ODBC query. This can be created in Genstat *for Windows* using the ODBC Data Query menu (accessible from the New option of the Spread menu on the menu bar). The DSN line in this text file can be used to connect to the same database as specified by the DB parameter.

You can set option DRIVER=32 to use 32-bit ODBC drivers when you are running the 64-bit Genstat; there is more information about this on the VSN website: www.vsni.co.uk.

Options: PRINT, INFORMATION, DRIVER.
Parameters: DB, GDBFILE, ISAVE.

Method

An SQL query to get the information is sent to the ODBCLOAD.DLL library which runs the query and saves the results in a temporary GWB file. This is then loaded using the SPLOAD directive.

Action with RESTRICT

Restrictions are not applicable to any of the parameters.

See also
Procedures: DBCOMMAND, DBEXPORT, DBIMPORT, IMPORT.
Genstat Reference Manual 1 Summary section on: Input and output.

DBIPLOT

Plots a biplot from an analysis by PCP, CVA or PCO (A.I. Glaser).

Options

PLOT = *string tokens*	Additional features for the plot (convexhull, means); default * i.e. none
METHOD = *string token*	Type of axes to plot (predictive, interpolative); default pred
HORIZONTAL = *identifier*	Which axis to make horizontal; default * i.e. none
PREDICTIONS = *matrix*	Saves predicted values
GROUPS = *factor*	Factor defining groupings of individuals for a PCP biplot; default * i.e. none
LMINDIVIDUALS = *string tokens*	How to label the individuals (labels, none, numbers, unitlabels); default labe if LINDIVIDUALS is set, otherwise unit
LMVARIABLES = *string tokens*	How to label the variables (identifiers, labels, none, numbers); default labe if LVARIABLES is set, otherwise iden
LINDIVIDUALS = *texts*	Labels for individuals (i.e. scores)
LVARIABLES = *texts*	Labels for variables (i.e. biplot axes)
MULTIPLIER = *scalar*	Value to multiply vector loadings; default * i.e. determined automatically
TITLE = *text*	Title for the plot; if this is unset, an appropriate title is formed automatically
WINDOW = *scalar*	Which graphical window to use; default 1 when there are groups, otherwise 3
KEYWINDOW = *scalar*	Which graphical window to use for the key when there are groupings of individuals (0 for none); default 2
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
SIZEMULTIPLIER = *scalar*	Multiplier used in the calculation of the size in which to draw symbols and labels; default 1
SAVE = *pointer*	Supplies results from an ordination analysis by PCP, CVA or PCO; default uses the most recent analysis

Parameters

VARIABLE = *identifiers*	Axis variables
DISPLAY = *string tokens*	Whether to show, hide or omit each axis (show, hide, omit); default show
COLOUR = *texts* or *scalars*	Colour to use to plot each axis

Description

DBIPLOT plots biplots displaying the results from a principal components, canonical variates or principal coordinates analysis, performed by the PCP, CVA or PCO directives. By default DBIPLOT uses the results from the most recent PCP, CVA or PCO, but you can display results from an earlier analysis by saving the information with the SAVE parameter of PCP, CVA or PCO, and then providing this to DBIPLOT using its own SAVE parameter.

Following the approach of Gower & Hand (1996), the biplot can be viewed as a multivariate analogue of the scatterplot. The information is plotted on the plane defined by the first two principal axes of the analysis (i.e. the first two principal components for a PCP, or the first two

canonical variates for a CVA). The default title of the biplot contains the percentage of variance explained by the first and second dimension combined, whilst the title of the x- and y-axis shows the amount of variation explained by the first and second dimension individually (you can specify your own title using the TITLE option). The scores from the analysis are plotted, to show the positions of the individual observations. More importantly, the plot contains an oblique "axis" for each variable (its *biplot axis*) that allows you to see how each individual's projection into this plane relates to its value for the variable concerned. The type of axis to be displayed will depend on how you want to use the plot. The possibilities, selected by the setting of the METHOD option, are as follows:

predictive	plots predictive axes (default),
interpolative	plots interpolative axes.

Predictive axes show the values of the variables that are predicted by the projection into 2-dimensions that is defined for each point by the analysis; essentially this is done by taking an orthogonal projection of the point onto each the biplot axis. Interpolative axes show the values of the variables that would lead to a point being placed at the position of the selected point on the graph. So here the point is being predicted by the variables, rather than the variables by the point. This is done by taking the sum of a set of vectors, one in the direction of each variable, with lengths equal to the values of the variables for that point.

The axes are defined from the loadings from the analysis. With a PCP analysis (or a PCO analysis based on a data matrix), the directions of the axes are given by loadings calculated in the analysis (but the positions of the scale points on the axes differ between the two types of axis). For a CVA analysis, the loadings define the interpolative axes for the biplots, and their inverses define the predictive axes. However, no loadings are available for PCO analyses based a dissimilarity matrices, and so no axes can be plotted. For further explanation, and details of the underlying mathematics, see Gower & Hand (1996).

Arrows are plotted on the axes to represent their loadings (or inverse loadings); the loadings show the approximate contribution of each variable in the first two dimensions. If the loadings are all close to the origin, they are multiplied by a scalar to make them easier to read. By default, the multiplier is calculated automatically, but you can supply a specific value by using the MULTIPLIER option. To save the automatic value, you can set MULTIPLIER to a scalar containing a missing value.

In general, each axis will be at an angle to the traditional x-axis. However, you can arrange for one of the biplot axes to be in the direction of the x-axis, by setting the HORIZONTAL option to the identifier of its variate. It should be noted that this operation is purely cosmetic and, if HORIZONTAL is not set, then the direction of the x-axis will represent the direction of maximum variance.

By default all the axes are plotted, each in a colour chosen automatically by DBIPLOT. However, there are parameters to allow you to modify this for any axis. The VARIABLE parameter specifies the axis to change (using its identifier). The DISPLAY parameter indicates whether the axis is to be shown, hidden or omitted altogether. (The Graphics Viewer of Genstat *for Windows* allows you to toggle displayed items to become hidden, or hidden items to become displayed.) The COLOUR parameter defines the colour to be used, by supplying either a single-valued text with the name of the colour or a scalar containing the RGB value for the colour (see the PEN directive for details).

The scores from PCP analyses are plotted to identify the position of each individual as a red circle, unless you use the GROUPS option to define groupings of the individuals (the groups are then plotted in different colours). With a CVA analysis, groupings are automatically defined from the groups in the analysis itself.

Hotpoints are defined at the point for each of individual to allow you to view the values corresponding to that individual on the axes. In the Graphics viewer in Genstat *for Windows*, you can click on the hotpoint symbol and then click on any score to see how that point is represented

on each of the axes. In addition, whatever axes are defined, you can use the PREDICTIONS option to save a matrix with the predicted values of the individuals for all the variables.

The PLOT option allows you to illustrate other aspects of the scores.

convexhull	draws a convex hull around the points (or the points in each group if groupings have been defined).
means	plots the group means for a CVA, or the group means for a PCP (if the GROUPS option is set), or the overall mean for a PCO biplot. (In other situations the centroid is the origin, which is where all the oblique axes cross, so it would clutter up an already congested plot.)

The types of label for the scores and loadings can be set using the LMINDIVIDUALS and LMVARIABLES parameters respectively, by selecting one of the following settings:

identifiers	uses the identifiers of the variables (LMVARIABLES only),
labels	expects labels to be supplied (in a text) using the LINDIVIDUALS or LVARIABLES parameter,
none	gives no labels,
numbers	uses the row or column numbers of the scores and variables, and
unitlabels	unit labels of the data variates or row labels of the data matrix, if present, otherwise the unit numbers (LMINDIVIDUALS only).

If LINDIVIDUALS is set, the default for LMINDIVIDUALS is defined to be labels; otherwise the default is unitlabels. Similarly, the default for LMVARIABLES is labels if LVARIABLES is set; otherwise it is defined to be identifiers.

The WINDOW and KEYWINDOW options specify the windows to use for the plot and its key, respectively, in the usual way. The SCREEN option controls whether the graphical display is cleared before the biplot is plotted.

The SIZEMULTIPLIER option allows you to modify the sizes of the symbols and labels in the plot. The default of 0.75 works well under most circumstances, but you might want to specify a smaller value to prevent overlapping, when there are large numbers of points or axes to be displayed.

Options: PLOT, METHOD, HORIZONTAL, PREDICTIONS, GROUPS, LMINDIVIDUALS, LMVARIABLES, LINDIVIDUALS, LVARIABLES, MULTIPLIER, TITLE, WINDOW, KEYWINDOW, SCREEN, SIZEMULTIPLIER, SAVE.
Parameters: VARIABLES, DISPLAY, COLOUR.

Method

The plots in DBIPLOT are explained in Gower & Hand (1996); see Chapter 2 for principal components, and Chapter 5 for canonical variates.

Reference

Gower, J.C. & Hand, D.J. (1996). *Biplots*. Chapman & Hall, London.

See also

Directives: CVA, PCO, PCP.
Procedures: BIPLOT, CABIPLOT, CRBIPLOT, CRTRIPLOT, CVAPLOT, GGEBIPLOT.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

DCIRCULAR

Plots circular data (P.W. Goedhart & R.W. Payne).

Options

PLOT = *string tokens*	Information to be plotted (counts, kerneldensity, lines, mean, rose); default coun, mean, rose
TITLE = *text*	Title for the graph; default * i.e. none
SEGMENT = *scalar*	Width of sectors (in degrees) into which to group an ANGLES variates before plotting; default 20
MSEGMENT = *scalar*	Defines the centre (in degrees) of the sectors; default 0
BANDWIDTH = *scalar*	Bandwidth to use for the kernel density estimate; if this is unset, the value h_0 suggested by Fisher (1993, page 26) is used
NGRID = *scalar*	Defines the number of grid points for the kernel density estimate; default 180
WINDOW = *scalar*	Window for the graph; default 3
SCREEN = *string token*	Whether to clear screen before displaying the graph (keep, clear); default clea

Parameters

ANGLES = *factors* or *variates*	Directional observations to be plotted
GRID = *variates*	Saves the grid (in degrees) on which the kernel density is estimated
DENSITY = *variates*	Saves the kernel density estimate
SAVEBANDWIDTH = *scalar*	Saves the calculated bandwidth h_0 when BANDWIDTH is unset

Description

DCIRCULAR plots data values that consist of directional observations recorded as angles between 0 and 360 degrees. The data values are supplied by the ANGLES parameter, in either a variate or a factor. With a variate, the observations are grouped for plotting into sectors of width specified (in degrees) by the SEGMENT option, with centres defined by the MSEGMENT option. The sectors are centred at MSEGMENT, MSEGMENT+SEGMENT, MSEGMENT+2*SEGMENT, and so on. The default value for SEGMENT and MSEGMENT is 20 and 0 respectively. If ANGLES is set to a factor, its levels define the midpoints of the sectors and these must be in clockwise order.

The graph contains a circle with marks at every 10 degrees, and labels at 0, 90, 180 and 270 degrees. The representations of the observations are determined by the settings supplied for the PLOT option as follows

counts	plots counts of the number of observations in each sector.
kerneldensity	plots estimates of the probability distribution of the data, using a quartic kernel function with bandwidth specified by the BANDWIDTH option. If BANDWIDTH is unset, a default is calculated based on the estimated concentration of the data (this is the value h_0 suggested by Fisher, 1993, page 26). The kernel is calculated on a grid of values with number of values defined by the NGRID option.
lines	plots lines in each direction with lengths proportional to the number of observations in that direction.
mean	plots the mean vector (see Fisher 1993, page 31).
rose	plots a "rose" diagram in which the observations in each sector are represented as a triangle with apex at the centre

of the circle and area proportional to the number of observations there.

By default `PLOT=counts,mean,rose`.

The options `TITLE`, `WINDOW` and `SCREEN` allow you to define a title for the plot, specify which window to use, and indicate whether or not to clear the screen beforehand. Parameters `GRID`, `DENSITY` and `SAVEBANDWIDTH` can be used to save the grid (in degrees), kernel estimate and bandwidth h_0. The latter is saved only when `BANDWIDTH` is unset.

Options: `PLOT`, `TITLE`, `SEGMENT`, `MSEGMENT`, `BANDWIDTH`, `NGRID`, `WINDOW`, `SCREEN`.
Parameters: `ANGLES`, `GRID`, `DENSITY`, `SAVEBANDWIDTH`.

Method

`DCIRCULAR` uses Genstat's standard graphics and calculation commands. The underlying methodology is described by Fisher (1993).

Action with `RESTRICT`

If `ANGLES` is restricted, only the unrestricted units are plotted.

Reference

Fisher, N.I. (1993). *Statistical Analysis of Circular Data*. Cambridge University Press, Cambridge.

See also

Procedures: `CASSOCIATION`, `CCOMPARE`, `CDESCRIBE`, `DYPOLAR`, `RCIRCULAR`, `WINDROSE`.
Genstat Reference Manual 1 Summary section on: Graphics.

DCLUSTERLABELS

Labels clusters in a single-page dendrogram plotted by DDENDROGRAM (R.W. Payne).

Options

WINDOW = *scalar*	Window containing the dendrogram; default 1
UNITS = *variate* or *text*	Names used for the units in the clusters supplied by CLUSTER
PEN = *scalar*	Pen to use to plot the labels; default 1

Parameters

CLUSTER = *variates* or *texts*	Specifies clusters to be labelled
LABEL = *texts*	Specifies the label to be plotted where each cluster is formed
YSAVE = *scalars*	Saves the y-coordinate where each label is plotted
XSAVE = *scalars*	Saves the x-coordinate where each label is plotted

Description

DCLUSTERLABELS can be used to plot labels by the positions, where clusters are formed in a dendrogram previously plotted by DDENDROGRAM.

The WINDOW option specifies the window containing the dendrogram; default 1. The PEN option specifies the pen to use for the plot; default 1.

The clusters are specified by the CLUSTER parameter. By default, each one is specified in a variate containing the numbers of the units in that cluster. (These numbers are the row or column positions of those units in the similarity matrix used by HCLUSTER.) You can form clusters like these using the HFCLUSTERS procedure. Alternatively, you can use textual labels or other numbers to identify the contents of the clusters, by supplying these in a text or a variate using the UNITS option. The contents of UNITS must be in the same order as the rows and columns of the similarity matrix used by HCLUSTER.

The label for each cluster is specified, in a single-valued text, by the LABEL parameter. The YSAVE and XSAVE parameters can save the y- and x-coordinate where each label is plotted, in scalars. You might want to adjust these, and use them to plot the labels on a new dendrogram, if some of the clusters are formed too close together.

Options: WINDOW, UNITS, PEN.
Parameters: CLUSTER, LABEL, YSAVE, XSAVE.

See also

Directive: HCLUSTER.
Procedures: DDENDROGRAM, HFCLUSTERS, HPCLUSTERS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

DCOLOURS

Forms a band of graduated colours for graphics (P.W. Goedhart).

Options

METHOD = *string token*	Type of colour band required (spectral, blackbody, linear); default line
PLOT = *string token*	What to plot (testgraph); default *

Parameters

START = *scalar* or *text*	Start value for the colour band; default * gives an appropriate default for the METHOD concerned
END = *scalar*, *text* or *variate*	End value(s) for the colour band; default * gives an appropriate default for the METHOD concerned
GAMMA = *scalar* or *variate*	The gamma-correction exponent(s) for the colour band; default 1
NCOLOURS = *scalar* or *variate*	Number(s) of colours in the colour band; default 20
RGB = *variates*	Saves the RGB colour values of each colour band
RED = *variates*	Saves the red component of the RGB colour values
GREEN = *variates*	Saves the green component of the RGB colour values
BLUE = *variates*	Saves the blue component of the RGB colour values
TITLE = *text*	General title for each test graph; default forms an informative title automatically
WINDOW = *scalar*	Window number for each test graph; default 1
SCREEN = *string token*	Whether to clear the screen before plotting each test graph or to continue plotting on the old screen (clear, keep); default clea

Description

Procedure DCOLOURS creates a colour of graduated colours by interpolating between start and end colour values. You can save the RGB colours of the band, in a variate, using the RGB parameter. Alternatively, you can save the red, green and blue components of the colours using the RED, GREEN and BLUE parameters (again in variates).

A test graph displaying the colour band can be requested by setting option PLOT=testgraph. The WINDOW parameter supplies the window number for the plot (default 1). The TITLE parameter can supply a title for the test graph; if this is not set, a suitable title is generated automatically. You can set parameter SCREEN=keep to plot the test graph on an existing screen; by default the screen is cleared first.

The METHOD option provides a choice of three different types of colour band. The default, METHOD=linear, forms the colours by interpolating between start and end RGB values. The start value is specified by the START parameter, as either a scalar defining an RGB colour value, or a text containing the name of one of the pre-defined Genstat colours (see the PEN directive for the available names, or search for "Graphics Colours" in the on-line help). You can set the END parameter to a single scalar or text (giving either the RGB value or the name of the colour) to define the band as a single sequence of colours. Alternatively, you can define a variate or a text with several values to form the band from several sequences of colours. At each END colour, DCOLOURS then begins a new sequence running from that colour to the next END colour. The default values for START and END are 'white' and 'black'.

Setting METHOD=spectral forms an approximate rainbow spectrum for wavelengths between 380 nm and 780 nm. There can now be only a single sequence of colours. The START and END parameters specify the start and end wavelengths, as scalars, with default values of 380 and 780.

The final setting, METHOD=blackbody, forms colours of hot objects with temperatures

between 500 K and 11000 K. Again, only a single sequence of colours is allowed. The START and END parameters specify the start and end temperatures, as scalars, with default values of 500 and 11000.

The NCOLOURS parameter specifies the number of colours in each sequence of colours, as a scalar for the spectral or blackbody methods, or as either a scalar or a variate for the linear method; the default is 20.

The red, green and blue values in each sequence are assumed by default to vary linearly with wavelength, temperature or red/green/blue components. Alternatively, you can use the GAMMA parameter to specify the power for a power transformation (default 1). It must be set to a scalar for the spectral or blackbody methods, and to either a scalar or a variate for the linear method. Its values must lie in the interval [0.25, 4].

The number of values specified by each set of END, GAMMA and NCOLOURS parameters can be different. However, the number of values in the setting of the END parameter determines the number of colour sequences in the band, and the values in the GAMMA setting and NCOLOURS setting are recycled as required.

Options: METHOD, PLOT.
Parameters: START, END, GAMMA, NCOLOURS, RGB, RED, GREEN, BLUE, TITLE, WINDOW, SCREEN.

Method
For a single linear colour band the red component is calculated as follows:

```
VARIATE   [VALUES=1...#NCOLOURS] count
CALCULATE count = (count - 1) / (NCOLOURS - 1)
&         red = RED(START) + (RED(END) - RED(START)) * count
&         red = INTEGER(255 * (red/255)**GAMMA)
```

Spectral and blackbody colours can be found at www.midnightkite.com/color.html which links to Fortran code for spectral colours at www.physics.sfasu.edu/astro/color/spectra.html, and for blackbody colours at www.physics.sfasu.edu/astro/color/blackbody.html.

Action with RESTRICT
Restrictions are not allowed.

See also
Directive: PEN.
Procedure: GETRGB.
Genstat Reference Manual 1 Summary section on: Graphics.

DCOMPOSITIONAL

Plots 3-part compositional data within a barycentric triangle (S.J. Clark).

Options

PRINT = *text*	What to print (proportions); default *
VERTEXLABELS = *text*	Labels for the vertices of the triangle; default * uses the names of the corresponding variates given in the DATA pointer
TITLE = *text*	Title for the barycentric triangle; default * (i.e. no title)
PERPENDICULARS = *text*	Whether to draw perpendiculars from each vertex to its opposite side (yes, no); default no
WINDOW = *number*	Which high-resolution graphics window to use; default 3
SCREEN = *string token*	Whether to clear the graphics screen before plotting (clear, keep); default clea

Parameters

DATA = *pointers*	Contains variates which form the three-part compositions
SCALE = *scalars*	Scale factor for adjusting size of triangle to represent a fourth category; default 1
SAVECOORDINATES = *pointers*	Saves the two-dimensional x- and y-coordinates into the first and second elements of the pointer, respectively
PEN = *scalars* or *variates* or *factors*	Pen number to draw points within the barycentric triangle; default 1

Description

DCOMPOSITIONAL plots three-part compositional data within a barycentric triangle (ternary diagram). Four-part compositional data can be represented when there is only one experimental unit.

Compositional data consist of vectors of proportions (one for each experimental unit). Each vector contains a set of *d* non-negative elements, each element representing a proportion of some whole, with the sum of the elements constrained to unity (Aitchison 1986). A composition with *d* elements per vector is termed a *d*-part composition. DCOMPOSITIONAL produces a graphical display of three-part compositions within an equilateral triangle with unit height (termed a barycentric triangle or ternary diagram), using high-resolution graphics. The three parts of the compositions are input using the DATA parameter as a pointer containing three separate variates. The first, second and third variates in the pointer should correspond, respectively, to the parts required to be represented at the top, bottom left and bottom right vertices of the triangle. DCOMPOSITIONAL also allows for data to be input on an original scale, in which case they will be converted to proportions of the totals. Variates representing the two-dimensional x- and y-coordinates can be saved (in a pointer) using the SAVECOORDINATES parameter. The PEN parameter may be used to specify the pen to use to plot the points within the barycentric triangle; the default setting is pen 1 (for which the initial defaults are METHOD=point and SYMBOLS=1).

The SCALE parameter should be set only when there is a single experimental unit (i.e. the length of each variate in the DATA pointer is one) and is provided to aid representation of a four-part composition. Its value, which should equal the sum of three of the parts divided by the total of all four parts, is used to scale the overall size of the triangle; by default SCALE is 1. The three parts in the numerator should be input using the DATA parameter, and the point plotted within the scaled triangle therefore represents the relative proportions of the three parts amongst themselves. This option is most useful when plotting several triangles in different windows of

the same frame.

The proportions can be printed by setting option PRINT=proportions. Labels for the vertices of the triangle can be specified by setting the VERTEXLABELS option to a text structure containing exactly three values: the first, second and third values should correspond, respectively, to the labels required at the top, bottom left and bottom right vertices of the triangle. By default the vertices will be labelled by the names of the corresponding variates given in the DATA pointer. The perpendiculars from each vertex to its opposite side will be drawn (using LINESTYLE=2) if the PERPENDICULARS option is set to yes; otherwise these lines are omitted. The graphical display can be controlled as usual using the TITLE, WINDOW and SCREEN options. By default triangles are produced in window 3, have no title, and are drawn on a new screen.

Options: PRINT, VERTEXLABELS, TITLE, PERPENDICULARS, WINDOW, SCREEN.
Parameters: DATA, SCALE, SAVECOORDINATES, PEN.

Method
The percentages of the totals are computed within the procedure (when proportions are input this will therefore have no effect), and standard graphics commands are used to produce high-quality graphical output.

Action with RESTRICT
If the variates input by the DATA parameter are restricted, only the selected points are plotted.

Reference
Aitchison, J. (1986) *The statistical Analysis of Compositional Data*. Chapman & Hall, London.

See also
Directive: DPIE.
Genstat Reference Manual 1 Summary section on: Graphics.

DCORRELATION

Plots a correlation matrix (A.I. Glaser).

Options

PLOT = *string tokens*	Type of plot (together, separate); default sepa
SHOW = *string tokens*	What features to include on the plots (axes, diagonal); default axes
NCOLOURS = *scalar*	Number of distinct colour to use from 0 to −1 or 1; default 20
COLOURS = *text* or *variate*	Text or variate with three values, defining the colours to use for correlations of −1, 0 and 1; default * chooses the colours automatically
WEIGHTS = *variate*	Provides weights for the units of the variates; default * assumes that they all have weight one

Parameters

PVARIATES = *pointers* or *symmetric matrices*	Pointer to either the first (P-) set or the only set of variates to be correlated, or symmetric matrix containing the correlations themselves
QVARIATES = *pointers*	Pointer to the second (Q-) set of variates to be correlated
PROWS = *scalars*	Specifies the number of rows corresponding the first (P-) set of variates in a correlation matrix supplied by PVARIATES, when this contains two sets
TITLE = *text*	Title for the plot

Description

DCORRELATION provides a graphical representation of a correlation matrix, which can show the correlation within a dataset, as well as the correlation within and between two different datasets. Each element of the correlation matrix is represented by a shaded rectangle indicating the value at that location, using a different colour or shading density. This type of display is often used before a canonical correlation analysis to see if there are any significant correlations within and between the datasets to be analysed; see the CANCORRELATION procedure for details.

The PVARIATES parameter can supply a symmetric matrix containing correlations that have already been calculated (e.g. using the FCORRELATION procedure). If the matrix involves two sets of variates (as in a canonical correlation analysis), you should arrange for them to be specified in set order i.e. all the first set, and then all the second set. You should then specify the number of variates in the first set using the PROWS parameter.

Alternatively, you can set PVARIATES to a pointer containing the variates themselves. You can then use the QVARIATES parameter to supply a pointer with a second set of variates.

The WEIGHTS option can provide a variate of weights for the units of the variates; by default these are all assumed to have weight one.

The PLOT option selects the type of plot, with settings:

together	to plot the correlation matrix as one symmetric matrix, with a dashed black line to show the boundaries between two datasets (if supplied), and
separate	to plot the correlation matrix in three separate components with the within dataset correlations at the top of the window, and the between-dataset correlations underneath. When there are more variates in the second (Q-) set than the first (P-) set, the separate plot will display the

transpose of the between-dataset correlations.

The default is PLOT=separate, unless there is only one set of variates when it defaults to 'together'.

The SHOW option controls whether some features are included on the plots:

axes	includes axes, and
diagonal	includes the diagonal of the correlation matrix.

The default is SHOW=axes.

There is also a key containing a strip of colours showing how the colours in the plot represent the different correlations. The NCOLOURS option specifies the number of distinct colours to use as the correlations decrease from 0 to −1 or increase from 0 to 1. This can vary from 2 upwards, with a default of 20. The COLOURS option allows you to control the range of colours that are used. It should be set to a text or variate with three values: the first value defines the colour to use for correlations of −1, the second value gives the colour for correlations of 0, and the third gives the colour for correlations of 1. (See PEN for details of how colours are defined.) The default colours, if COLOURS is unset, range from dark blue for values close to −1 to dark red for values close to 1.

The TITLE parameter supplies a main title for each plot.

Options: PLOT, SHOW, NCOLOURS, COLOURS, WEIGHTS.
Parameters: PVARIATES, QVARIATES, PROWS, TITLE.

Method

The plots in DCORRELATION are produced using DBITMAP.

See also

Directive: CORRELATE.
Procedures: FCORRELATION, PARTIALCORRELATIONS, PRCORRELATION.
Genstat Reference Manual 1 Summary sections on: Graphics, Multivariate and cluster analysis.

DCOVARIOGRAM

Plots models fitted to 2-dimensional auto- and cross-variograms (D.A. Murray).

Options

PLOT = *string token*	Controls how to display the plotted variograms (separate, scattermatrix); default scat
ESTIMATES = *pointer*	Pointer containing model estimates saved from MCOVARIOGRAM

Parameter

COVARIOGRAM = *pointer*	Pointer to supply the semi-variances, distances and associated information as saved from FCOVARIOGRAM

Description

DCOVARIOGRAM plots 2-dimensional auto- and cross-variograms using data generated by FCOVARIOGRAM. DCOVARIOGRAM can also be used to display the fitted model for isotropic models using estimates generated from MCOVARIOGRAM. The data should be supplied in a pointer that has been saved using the COVARIOGRAM option from FCOVARIOGRAM. This pointer provides the auto-variograms, cross-variograms and associated information required for the plots. Its elements contain:

1 a matrix with columns of variograms and cross-variograms and rows indexed by lags within directions;
2 a variate of counts at the lags in each direction;
3 distances of the lags in each direction;
4 horizontal angles;
5 vertical angles;
6 variances;
7 distance classes;
8 method;
9 pointer containing identifiers of the DATA variates;
10 number of dimensions.

The ESTIMATES option can be used to plot an isotropic fitted model of coregionalization where the estimates are taken directly from MCOVARIOGRAM. Graphical output is controlled using the PLOT option. The setting separate produces each auto- and cross-variogram on a separate plot. Alternatively, they can be combined onto a single scatter matrix using the scattermatrix setting.

Options: PLOT, ESTIMATES.
Parameter: COVARIOGRAM.

Method

DCOVARIOGRAM uses Genstat's standard graphics and calculation commands. See MCOVARIOGRAM directive for details on the models and estimates used for plotting the model of coregionalization.

Reference

Webster, R. & Oliver, M.A. (2001). *Geostatistics for Environmental Scientists*. Wiley, Chichester.

See also
Directives: FCOVARIOGRAM, MCOVARIOGRAM, COKRIGE.
Procedure: DVARIOGRAM.
Genstat Reference Manual 1 Summary sections on: Spatial statistics, Graphics.

DDEEXPORT

Sends data or commands to a Dynamic Data Exchange (DDE) server, PC Windows only (D.B. Baird).

Options

SERVER = *text*	Name of DDE Server; default Excel
TOPIC = *text*	Name of DDE Topic
ITEM = *text*	Name of DDE Item; default R1C1
OUTFILE = *text*	Name of Excel or Quattro Pro file
SHEETNAME = *text*	Name of Excel or Quattro Pro sheet within the file
COLUMN = *text* or *scalar*	Name or number or column of the first cell to write to, either as a text (e.g. 'A', 'AN') or a number (e.g. 1, 40)
ROW = *text* or *scalar*	Number of the row of the first cell to write to, i.e. '2', or 2
LABELLING = *text*	What labels to write to the DDE server (rows, columns) default rows, colu
METHOD = *string token*	Whether the DATA parameter specifies a text which is send as a series of commands to the DDE server or data values (data, commands); default data

Parameter

DATA = *identifiers*	The data structures to be written to the DDE server, or a text containing the commands to be sent to the server

Description

The data to be written via DDE is specified by the DDE triplet: server, topic and item. For convenience this has been broken down to the components OUTFILE, SHEETNAME, COLUMN and ROW for the two common spreadsheets Excel and Quattro Pro for Windows. For Excel/QPW also only the first cell need be provided and Genstat will automatically work out the correct item range given the size of the DATA structures passed to DDEEXPORT.

When a command is being sent to a DDE server (METHOD=command), the DATA must be a text, and only the SERVER and TOPIC options need to be set.

The LABELLING option allows you to only send the required aspects of the data to the DDE server.

The TOPIC for Excel has the format '[<FileName>]<SheetName>' e.g. '[D:/Work/Data.XLS]Data Summary', and the ITEM has the numerical format R<n1>C<n2>:R<n3>C<n4> e.g. 'R2C3:R25C5' or the format '<Column letter><rowno>:<Column letter><rowno>' e.g. 'C2:E25'.

The TOPIC for QPW has the format 'FileName' e.g. 'D:/Work/Data.WB3', and the ITEM has the format of 'Sheet:<Column letter><rowno>..<Column letter><rowno>' e.g. 'B:C2..E25'.

The use of DDEEXPORT is illustrated in the following examples. To write three variates to a QPW file in the first sheet in cells B2..D8:

```
DDEEXPORT [SERVER=QPW; TOPIC='C:\\WORKBOOK.QPW';\
  ITEM='A:B2..D8'] X,Y,Z
```

To write a matrix to Excel in the cells starting at D5:

```
DDEXPORT \
  [topic='[D:\\\\DATA\\\\RESULTS.XLS]Sheet2';item='D5'] VC
```

To send a set of DDE commands to Excel:

```
TEXT CMDS; !T('[OPEN(""C:\\\\TRIAL\\\\DATA.XLS"")]', \
  '[WORKBOOK.INSERT(1)]', \
```

```
  '[SELECT(""A4..D8"")]')
DDEXPORT [METHOD=command] CMDS
```

Excel DDE commands are a subset of the Excel 4 macro language. The format of the commands are [Function(arg1,arg1,...)]. text strings in the arguments are quoted with double quotes (e.g. "A1"). The following is a subset of Excel commands which may be useful:

`[APP.RESTORE()]`	Restore the Excel window
`[APP.MINIMIZE()]`	Minimize the Excel window
`[APP.ACTIVATE()]`	Make Excel the application with focus
`[OPEN("filename")]`	Open a workbook in Excel
`[WORKBOOK.INSERT(1)]`	Insert an new workbook
`[WORKBOOK.SELECT("sheetname")]`	Make the named sheet the current sheet
`[WORKBOOK.DELETE()]`	Delete the current sheet
`[SELECT("object")]`	Select the cells/column/rows specified in object
`[SORT(1,"R1C1",1)]`	Sort the selected cells using key in specified cell
`[SAVE()]`	Save the current workbook
`[SAVE.AS("filename",1)]`	Save the current workwork as a new file
`[CLOSE(1)]`	Close and save the current workwork (0 = close but don't save)

Complete details are available in the Windows help file Macrofun.hlp available on the Microsoft Internet site.

(Note: `DDEEXPORT` replaces the procedure `%DDE` from earlier editions of Genstat.)

Options: `SERVER`, `TOPIC`, `ITEM`, `OUTFILE`, `SHEETNAME`, `COLUMN`, `ROW`, `LABELLING`, `METHOD`.
Parameter: `DATA`.

Action with `RESTRICT`

Restrictions on the structures are ignored and all data will be sent to the DDE server. However, if the restrictions on the structures are not consistent, a fault will occur.

See also

Procedure: `DDEIMPORT`.
Genstat Reference Manual 1 Summary section on: Input and output.

DDEIMPORT

Get data from a Dynamic Data Exchange (DDE) server, PC Windows only (D.B. Baird).

Options

PRINT = *string token*	Controls whether a catalogue of the structures read from the DDE server is printed (catalogue); default cata
SHOW = *string token*	Whether to display the spreadsheet within the windows interface (yes, no); default no
IMETHOD = *string token*	How identifiers are specified for the columns (read, supply, none); default read if COLUMNS is unset, supply otherwise

Parameters

SERVER = *text*	Name of DDE Server (ignored if GDEFILE is set); default Excel
TOPIC = *text*	Name of DDE Topic
ITEM = *text*	Name of DDE Item
GDEFILE = *text*	Name of a previously saved Genstat DDE link file
COLUMNS = *text*	Names and/or type codes for the columns read (the type of column can be forced by ending the column name, if supplied, with the code ! for a variate, # for a variate, and $ for a text)
ISAVE = *pointer*	Name of a pointer to save the column identifiers

Description

A DDE server is another windows program (e.g. Microsoft Excel) that will supply data on a DDE request. The data to be read via DDE is specified by the DDE triplet: server, topic and item. Alternatively the name of an existing GDE file containing these can be specified using the GDEFILE parameter. GDE files can be created by opening a spreadsheet with the Spread > New > from DDE Link menu item, and then saving the resulting spreadsheet with the file type Genstat DDE Link (*.GDE).

The TOPIC for Excel has the format '[<FileName>]<SheetName>' e.g. '[D:/Work/Data.XLS]Data Summary', and the ITEM has the numerical format R<n1>C<n2>:R<n3>C<n4> e.g. 'R2C3:R25C5' or the format '<Column letter><rowno>:<Column letter><rowno>' e.g. 'C2:E25'.

The TOPIC for QPW has the format 'FileName' e.g. 'D:/Work/Data.WB3', and the ITEM has the format of 'Sheet:<Column letter><rowno>..<Column letter><rowno>' e.g. 'B:C2..E25'.

The COLUMNS parameter can be used to set the names of the structures. It can also be used to force the type of column by ending the column name with the code ! for a variate, # for a variate, and $ for a text. For example

```
COLUMN=!T('Trt!','ID$','Rank#')
```

will create a factor called Trt, a text called ID and a variate called Rank. If only the type code is provided, the columns will not be renamed, but the new types will set, e.g.

```
COLUMN=!T('!','$','#')
```

will force the first three columns to be of type factor, variate and text respectively.

(Note: DDEIMPORT replaces the procedure DDELOAD from earlier editions of Genstat.)

Options: PRINT, SHOW, IMETHOD.
Parameters: SERVER, TOPIC, ITEM, GDEFILE, COLUMNS, ISAVE.

Method
The DDE server is queried with the TOPIC and ITEM, and any data received from the DDE server is sent to a temporary GSH file which is read in with the SPLOAD directive.

Action with RESTRICT
Restrictions are not applicable to any of the parameters.

See also
Procedure: DDEEXPORT.
Genstat Reference Manual 1 Summary section on: Input and output.

DDENDROGRAM

Draws dendrograms with control over structure and style (P.G.N. Digby).

Options

STYLE = *string token*	Style to use for the links of the dendrogram (average, centroid, lower, full); default aver
ORDERING = *string tokens*	How to define the order of the units for the dendrogram (given, ziggurat, size, first); default zigg, size, firs
REVERSE = *string token*	Whether to reverse the order of the units in the dendrogram (no, yes); default no
ORIENTATION = *string token*	Specifies the orientation of a dendrogram produced by high-resolution graphics (north, south, east, west); default west
METHOD = *string token*	Method used to represent the scale on which the amalgamations have been made: settings other than the default are relevant only for data not generated by HCLUSTER or HDISPLAY (similarities, percentages, distances); default simi
SCREEN = *string token*	Setting to use for the SCREEN option of DGRAPH (clear, keep); default clea
CHANGE = *string token*	If a dendrogram-save structure from a previous DDENDROGRAM is used as the DATA parameter then this option specifies the area of the process where the first changes occur: see the description of the SAVE parameter (order, dendrogram, display); default orde
GRAPHICS = *string token*	Form of graphics to be used (lineprinter, highresolution); default high
DSIMILARITY = *string token*	Whether to display an axis for the similarities in high-resolution graphics (no, yes); default no
LOWSIMILARITY = *scalar*	Lower value to be used for the axis showing the similarities; default * i.e. determined from the data
NPAGES = *scalar*	Number of pages to use for a high-resolution plot; default 1
PAGEINFORMATION = *string tokens*	Controls what to include in a multi-page plot (similarity, title, pagenumber); default simi, titl, page
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the current setting

Parameters

DATA = *matrices* or *pointers*	Data defining each dendrogram in the form of either a matrix saved using the AMALGAMATIONS parameter of HCLUSTER (methods other than single linkage), or a matrix from the TREE parameter of HDISPLAY, or a SAVE structure from a previous use of DDENDROGRAM
PERMUTATION = *variates*	Specify or save permutations of the units for drawing each dendrogram, according to ORDERING option
LABELS = *variates* or *texts*	Supply labels to use for the units of each dendrogram; these should be in the natural order of the units, not in a

	permuted order
TITLE = *texts*	Titles for the dendrograms
WINDOW = *scalars*	Window to use for each dendrogram (window 1 if unset); if this is set to zero the dendrogram is not drawn, but results can still be saved using the PERMUTATION, ZIGGURAT and SAVE parameters
PENS = *scalars*, *variates*, *strings* or *texts*	Scalar or string specifying the graphics pen or symbol in which to draw each (high-resolution or line-printer) dendrogram; alternatively use of a variate or text allows the structure of each dendrogram to be highlighted by drawing different links with different graphics pens or symbols
ZIGGURAT = *variates*	Save the "ziggurat-degree" of the links in each dendrogram
SAVE = *pointers*	Save the information required to plot a dendrogram, for use as input for the DATA parameter in a subsequent call to DDENDROGRAM

Description

DDENDROGRAM draws dendrograms using line-printer or high-resolution graphics, as indicated by the GRAPHICS option. Dendrograms can be drawn in many ways, often with apparently quite different results, as illustrated by Digby (1985). Considerable control is allowed over the way in which the dendrogram is formed; in particular the order of the units and the style used for drawing the links of the dendrogram can be varied.

The information defining the dendrogram is given by the DATA parameter. This should be a matrix containing the amalgamations information from hierarchical cluster analysis (from the AMALGAMATIONS parameter of HCLUSTER) or a matrix containing the minimum spanning tree information (from the TREE parameter of the HDISPLAY directive); alternatively a SAVE structure from a previous DDENDROGRAM can be used as input. However, the amalgamations matrix from HCLUSTER is unusable if the clustering has been produced by single linkage, so the minimum spanning tree information, which is equivalent, should be used as input instead.

The PERMUTATION parameter can be supplied with a variate, either to specify a permutation of the rows of the dendrogram or to save the permutation generated by DDENDROGRAM, as indicated by the ORDERING option. Setting ORDERING=given takes the ordering defined by the PERMUTATION variate. The other settings of ORDERING define partial orderings of the units, and are used in conjunction with each other to obtain the full ordering: ziggurat (Critchley 1983) is associated with ultrametric distances amongst the units; size specifies that when 2 groups merge the smaller is always placed before the larger in the order; first specifies that when 2 groups merge the group containing the lowest numbered unit is always placed before the other in the order. The orders given by settings ziggurat and size are not completely specified and recourse may be made to the other of these settings or to first. If ORDERING is not set to given then a list of settings may be specified in which case the first in the list is used, the second is used to satisfy indeterminacies in the order given by the first setting in the list, and so on. The default is the list of settings: ziggurat, size, first. The REVERSE option allows the ordering thus obtained to be reversed.

The LABELS parameter can be given a variate or a text to supply labels for the rows of the dendrogram. Labelling can be suppressed altogether by using a text containing only spaces.

The STYLE option controls the style to use in forming the links of the dendrogram: its setting indicates where the line representing each new cluster should be placed. Assuming that the dendrogram has the units on the left-hand side, the settings can be described as follows:

`average` (the default) the new line is midway between the old lines; `centroid` the new line is placed at the mid-point of all the units in the group it represents; `lower` the new line is a continuation of the lower of the two old lines (comparable with dendrograms from `HCLUSTER`); `full` the new line is a continuation of the upper or lower of the two old lines, so that each vertical line spans all the units in the group it represents.

The `ORIENTATION` option is relevant only to high-resolution graphics, when it controls the orientation of the dendrogram: for example the setting `north` results in a "hanging dendrogram" with the units across the top. The default setting is `west`, which gives a dendrogram with the units on the left-hand side; this is also how `DDENDROGRAM` draws dendrograms on the line-printer.

The `METHOD` option indicates the scale on which the amalgamations have been made. This option need be set only if the data have been obtained from a source other than `HCLUSTER` or `HDISPLAY`.

The `TITLE` parameter specifies a title for each dendrogram. For high-resolution graphics, the `WINDOW` parameter defines the graphics window to use for each plot. With line-printer graphics, two "windows" are available: window 1 has a width of 101 characters, window 2 a width of 61 characters. If `WINDOW` is not set, window 1 is used. If it is set to zero, the dendrogram is not drawn but results can still be saved using the `PERMUTATION`, `ZIGGURAT` and `SAVE` parameters; however, if the `SAVE` structure is used later as input to `DDENDROGRAM`, the `CHANGE` option must not be set to `display` as the dendrogram stage will not have been completed.

The `LOWSIMILARITY` option allows the lower value of the axis showing the similarities (or percentage similarities or distances, according to the setting of the `METHOD` option) to be set e.g. to zero. Otherwise, this is determined automatically from the minimum value in the data. By default the axis is not plotted, but this can be changed by setting option `DSIMILARITY=yes`.

The `NPAGES` option allows the display to be split over several pages in a high-resolution plot. The `PAGEINFORMATION` option then controls what information is shown on the pages:

`similarity`	includes the similarity axis on pages 2 onwards when `DSIMILARITY=yes` (otherwise it appears only on page 1),
`title`	includes the `TITLE` on pages 2 onwards, and
`pagenumber`	includes page numbers.

As in other graphics commands, the `SCREEN` option controls whether to clear the high-resolution graphics screen before plotting (default `clear`), and the `ENDACTION` option controls whether Genstat pauses or continues after completing the plot.

For high-resolution graphics, the `PENS` parameter can be set to a scalar to define the pen to use to draw the dendrogram. Alternatively, a variate can be specified to highlight the structure of the dendrogram by drawing different links with different pens; the links are taken in the same order as the rows of the `AMALGAMATIONS` matrix from `HCLUSTER` or in increasing order of the links of the minimum spanning tree. `DDENDROGRAM` will use pen 1 if the `PENS` parameter is not set. Any pens used by `DDENDROGRAM` will be set to `METHOD=line`, `SYMBOLS=0`, `JOIN=given`. If a scalar is supplied or `PENS` is not set, the pen used will also have `LINESTYLE` set to 1. If a variate is used, appropriate settings of `COLOUR` and `LINESTYLE` should set (using the `PEN` directive) prior to calling `DDENDROGRAM`. Similarly, with line-printer graphics, the `PENS` parameter can be set either to a string or to a text, according to whether the links are to be drawn with the same or different symbols; if the parameter is unset, the plus symbol (+) is used for all the links.

The `ZIGGURAT` parameter can be used to save the "ziggurat-degree" (Critchley 1983) of each link. This could then be used to form the setting of the `PENS` parameter for a later dendrogram, in order to display particular aspects of the clustering more clearly.

The `SAVE` parameter can be used to save the various structures that control the drawing of a dendrogram in order to save computing time when drawing a similar dendrogram. The `SAVE` structure should then be used as the setting of the `DATA` parameter, and the `CHANGE` option used

to indicate the stage at which to start changing aspects of the previous dendrogram. The various stages (in order) involve the following options and parameters:

order	ORDERING and PERMUTATION;
dendrogram	STYLE and METHOD;
display	REVERSE, ORIENTATION, SCREEN, LABELS, TITLE, WINDOW, PENS, DSIMILARITY and LOWSIMILARITY.

Options: STYLE, ORDERING, REVERSE, ORIENTATION, METHOD, SCREEN, CHANGE, GRAPHICS, DSIMILARITY, LOWSIMILARITY, NPAGES, PAGEINFORMATION, ENDACTION.
Parameters: DATA, PERMUTATION, LABELS, TITLE, WINDOW, PENS, ZIGGURAT, SAVE.

Method

Dendrograms are constructed and drawn in four separate stages: firstly the amalgamations information is used to construct information on group sizes; secondly a permutation of the units is formed, if required, according to several possible ordering schemes; thirdly graphical information on each of the links of the dendrogram is formed; lastly this graphical information is used to display the dendrogram, subject to requirements over orientation, pens, etc. Separate procedures are used for each stage (for details see the source code of DDENDROGRAM, obtainable via LIBEXAMPLE). A preliminary stage is also needed to construct the amalgamations from information on a minimum spanning tree. Communication amongst the subsidiary procedures is obtained using a pointer, which the user may keep using the SAVE parameter. The algorithms used by the first three subsidiary procedures are similar to those described by Digby (1984a, 1984b).

Action with RESTRICT

If any of the options or parameters are restricted unpredictable results may occur: none of the options or parameters should be restricted.

References

Critchley, F. (1983). Ziggurats and dendrograms. *Report No. 43*. Department of Statistics, University of Warwick.

Digby, P.G.N. (1984a). Drawing pretty dendrograms. *Genstat Newsletter*, **14**, 18-26.

Digby, P.G.N. (1984b). Dendrograms and ziggurats. *Genstat Newsletter*, **14**, 14-18.

Digby, P.G.N. (1985). Graphical displays for classification. *PACT Journal of the European Study Group on Physical, Chemical and Mathematical Techniques Applied to Archaeology*.

See also

Directives: HCLUSTER, HDISPLAY.
Procedure: DCLUSTERLABELS.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

DDESIGN

Plots the plan of an experimental design (K.E. Bicknell & R.W. Payne).

Options

`Y` = *variate*	Specifies the *y* position of the plots in standard coordinates 1 ... number of rows of plots in the experiment (taking 1 as the top row of the window)
`X` = *variate*	Specifies the *x*-coordinate of the plots in standard coordinates 1 ... number of columns of experimental plots
`TITLE` = *text*	Title for the plan
`WINDOW` = *scalar*	Window number for the plan; default 3
`KEYWINDOW` = *scalar*	Window number for the key; default 0
`SCREEN` = *string token*	Whether to clear the screen before plotting (`clear`, `keep`); default `clea`
`KEYDESCRIPTION` = *text*	Overall description for the key; default `*`
`ENDACTION` = *string token*	Action to be taken after completing the plot (`continue`, `pause`); default `*` uses the setting from the last `DEVICE` statement
`CHARACTERS` = *scalar*	Sets a limit on the length of each factor label; default `*` i.e. none
`SIZE` = *scalar*	Provides a multiplier by which to scale the sizes of the factor labels on the plan

Parameters

`FACTOR` = *factors*	Factors to be listed on the plan and to define the layout (the procedure determines the style of line to divide each pair of plots in the design from the grid pen of the first factor in the list with which they have different levels); default `*` forms the list from first the factors specified by a preceding `BLOCKSTRUCTURE` statement, and then those specified by a preceding `TREATMENTSTRUCTURE` statement
`PEN` = *scalars*	Pen to be used to write the levels of each factor on the plan (if `PEN=0` the levels of that factor are not included); default 1 if the `FACTOR` parameter is specified, otherwise 0 for block factors and 1 for treatment factors
`PENGRID` = *scalars*	Pens to be used to draw the boundaries between the plots in the design (according to the first `FACTOR` with which they have different levels but ignoring factors with `PENGRID=0`); default 1,2...
`LABELS` = *texts*	Labels to be used for each factor if its own levels or labels are inappropriate

Description

`DDESIGN` uses high-resolution graphics to produce a plan of an experimental design. The plots in the design are assumed to be arranged on a rectangular grid. The rows of the plots are assumed to run from 1 (at the top of the graph) upwards and are specified by a variate supplied by the `Y` option. The columns (again running from 1 upwards) specified by a variate supplied by the `X` option. If either `Y` or `X` is not specified, `DDESIGN` will generate values automatically according to the factors in the design.

The TITLE, WINDOW, KEYWINDOW, SCREEN, KEYDESCRIPTION and ENDACTION options operate as usual in high-resolution graphics. The CHARACTERS option allows a limit to be set on the length of each factor label when written on the plan, and the SIZE option allows the size of the plotted factor labels to be scaled (using the SIZE parameter of the PEN directive).

The factors involved in the experiment can be listed using the FACTOR parameter. If this is omitted DDESIGN forms the list firstly from the factors in the previous BLOCKSTRUCTURE statement (or a "units" factor if there was none), and then from the factors (if any) in the previous TREATMENTSTRUCTURE statement.

These factors are then used to draw the plan and to label the plots in the design. The PEN parameter allows the levels or labels of the factors to be drawn using different pens (and thus, for example, in different colours). If the pen for any factor is defined as zero, its levels/labels are not included. However, it can still be used to determine the lines drawn to delimit the plots. For these lines, DDESIGN considers each pair of adjacent plots and checks through the list of factors to find the first one for which they have different levels. It then uses the grid pen (defined by the PENGRID parameter) to draw the dividing line. If the grid pen of any factor is zero, it is ignored.

This makes it very easy to achieve the usual style of plan in which stronger lines are used for example to indicate the boundaries between different blocks than between the plots within blocks. For example, the parameter settings to draw a randomized block design with a single treatment factor Treat in this way would be

```
FACTOR=Block,Plots,Treat; PEN=1; PENGRID=1,2,0
```

if all the factors are to have their levels listed within the plots, or

```
FACTOR=Block,Plots,Treat; PEN=0,0,1; PENGRID=1,2,0
```

if only Treat is to be listed. Note that, as each pair of plots will have different levels of either Block or Plot (or both), the PENGRID specified here for Treat is irrelevant.

If a plot has no neighbour in some direction, DDESIGN will check the next but one plot; if this too is not used in the design, the grid pen of the first FACTOR is used to mark the boundary.

The final parameter, LABELS, allows alternative labels to be specified for each factor if the existing ones are inappropriate.

Options: Y, X, TITLE, WINDOW, KEYWINDOW, SCREEN, KEYDESCRIPTION, ENDACTION, CHARACTERS, SIZE.
Parameters: FACTOR, PEN, PENGRID, LABELS.

Method

DDESIGN makes use only of standard Genstat facilities for manipulation and plotting.

Action with RESTRICT

If any of the factors or X or Y is restricted, only the unrestricted plots are displayed.

See also

Procedures: ADSPREADSHEET, PDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

DECIMALS

Sets the number of decimals for a structure, using its round-off (A. Keen).

Options

SETATTRIBUTE = *string token*	Attributes to be redefined for STRUCTURE (decimals); default deci
SIGNIFICANTFIGURES = *scalar*	Required number of significant figures; default takes the system default, which can be modified by SET

Parameters

STRUCTURE = *identifiers*	Numerical structure for which the number of decimals is to be set
DECIMALS = *scalars*	To save the number of decimals to use for all the values of each structure
ROUND = *scalars*	To save the round-off provided by using DECIMALS decimal places
VDECIMALS = *structures*	To save numbers of decimals for every value of each structure
VROUND = *structures*	To save the round-off for every value of each structure

Description

The number of decimals that Genstat uses as a default, when printing a numerical structure, is calculated as the number required to display the mean of the absolute values of the numbers in the structure to a standard number of significant figures (see the PRINT directive). Usually the standard number of significant figures is four, but this "system default" can be changed using the SIGNIFICANTFIGURES option of the SET directive. The default method allows output to be generated automatically with reasonable accuracy. However, it may be preferable to use fewer decimals if the numbers can be represented exactly with three or fewer significant figures. For example it may be preferable to use two decimal places rather than four for a variate containing the values 0.1 and 0.21 (i.e. to print 0.10 and 0.21, rather than 0.1000 and 0.2100).

The DECIMALS procedure operates similarly to the standard Genstat default, except that the number of decimal places is decreased if the final decimal position would contain the digit zero for every value of the structure. It also differs in that it has its own SIGNIFICANTFIGURES option to change the required number of significant figures from the system default.

The numerical structure for which the number of decimals is to be determined must be supplied using the STRUCTURE parameter. The DECIMALS parameter can save the appropriate number of decimal places (as a scalar), and parameter ROUND can save the maximum round-off over the values of the structure (see Method). By default DECIMALS modifies the declaration of the STRUCTURE so that this becomes its default number of decimal places for subsequent printing (see the DECIMALS parameter of SCALAR, VARIATE, TABLE, MATRIX and SYMMETRICMATRIX). However, you can set option SETTATTRIBUTE=* if you want the default number of decimals to remain unchanged.

DECIMALS can also calculate a separate number of decimal places for each of the values of the STRUCTURE. This can be saved (in a structure of the same type as the STRUCTURE) using the VDECIMALS parameter, and the round-off for each value can similarly be saved using the VROUNDOFF parameter.

Options: SETATTRIBUTE, SIGNIFICANTFIGURES.
Parameters: STRUCTURE, DECIMALS, ROUND, VDECIMALS, VROUND.

Method
The round-off value of a number equals 10^k with k a negative or positive integer or zero. The round-off value of a number equals d if the number after dividing by d is an integer but after dividing by $10 \times d$ is not. If the round-off value is such that the number of significant digits is greater than 4, the round-off value is increased correspondingly. For example, the round-off value of 880 equals 10, that of 0.2300 equals 0.01 and of 9999.11 equals 1. The round-off value of a structure is the minimum of the round-off values of all the elements of the structure, subject to the restriction that the number of significant digits does not exceed 4 for any of the values of the structure.

The number of decimals of a structure is calculated from the round-off value of the structure as $-\log_{10}$(round-off value), with minimum value zero. So in the above examples the number of decimals equals 0 for 880, 2 for 0.2300 and 0 for 9999.11.

Action with `RESTRICT`
Restrictions are ignored.

See also
Directives: `PRINT`, `SET`.
Procedure: `MINFIELDWIDTH`.
Genstat Reference Manual 1 Summary sections on: Input and output, Data structures.

†DELLIPSE

Draws a 2-dimensional scatter plot with confidence, prediction and/or equal-frequency ellipses superimposed (V.M. Cave).

Options

PLOT = *string tokens*	What type of ellipse to plot (confidence, prediction, equalfrequency); default conf
PROBABILITY = *scalar* or *variate*	Probability level(s) for the ellipse(s); default 0.95
NPOINTS = *scalar*	Number of points used to draw the ellipses; default 1000
DISPLAY = *string token*	Whether to include the data points on the graph (show, hide); default show
PAXES = *string token*	Whether to plot the principal axes on the graph (no, yes); default no
TFILL = *scalar*	Transparency used to fill the area inside the ellipses, on a scale of 0 (opaque) to 255 (completely transparent); default 255
USEPENS = *string token*	Whether to use the current pen definitions for drawing the ellipses, drawing the principal axes and plotting the data (no, yes); default no
CMATCH = *string token*	When USEPENS=yes and groups are to be plotted, indicates whether the colours for the ellipses and principal axes are matched to the corresponding group, or to the colours defined by the pens for the different ellipse types and principal axes (group, pen); default group
WINDOW = *scalar*	Window to use for the graph(s); default 1
KEYWINDOW = *scalar*	Window to use for the key; by default the key is drawn on the right, in window 255
KEYDESCRIPTION = *text*	Overall title for the key; default * i.e. none
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clear

Parameters

Y = *variates* or *pointers*	Vertical coordinates (i.e. variable to plot on the y-axis)
X = *variates* or *pointers*	Horizontal coordinates (i.e. variable to plot on the x-axis)
GROUPS = *factors*	Defines groupings of the data points
DESCRIPTION = *texts*	Labels for the groups; default generates the labels automatically
TITLE = *text*	Title for the plot; default * i.e. none
YTITLE = *text*	Title for the y-axis; by default a title is generated automatically
XTITLE = *text*	Title for the x-axis; by default a title is generated automatically

Description

The DELLIPSE procedure produces a 2-dimensional scatter plot with confidence, prediction and/or equal-frequency ellipses superimposed, using high-resolution graphics.

The parameters Y and X supply variates or pointers containing the y- and x-coordinates to be plotted in the scatter plot, respectively. The DISPLAY option allows you to control whether the

data points are shown or hidden in the initial graph displayed by the Graphics Viewer. By default, they are displayed (i.e. DISPLAY=show).

The PROBABILITY option specifies the probability level for the ellipse, or ellipses, that are superimposed onto the scatter plot; default 0.95. You can supply either a single probability in a scalar, or several in a variate. When a variate is supplied, an ellipse is drawn at each value.

The PLOT option specifies the types of ellipses to be plotted. These are:

confidence	confidence ellipse (the default),
prediction	prediction ellipse,
equalfrequency	equal-frequency.

Assuming a bivariate normal distribution, for PROBABILITY level p it is expected that $(100\times p)$% of confidence ellipses will contain the true bivariate mean, $(100\times p)$% of prediction ellipses will contain a future bivariate observation, and $(100\times p)$% of the observed data will be enclosed within the equal-frequency ellipse. When PROBABILITY specifies several probability levels, a separate graph is drawn for each selected ellipse type, unless over-written by setting the option SCREEN=keep.

When both X and Y supply variates, the GROUPS option can be used to specify a factor to partition the data into different groups, so that separate ellipses are superimposed for each group. Alternatively, groups of data can be plotted by supplying pointers for X and/or Y. Here, the elements of the pointer(s) define separate groups. For example, if both X and Y are pointers of length 2, then separate ellipses are drawn for the groups of data (X[1],Y[1]) and (X[2],Y[2]). Similarly, if X supplies a pointer of length 2 but Y supplies a variate, then separate ellipses are drawn for (X[1],Y) and (X[2],Y). The number of groups must not exceed 25. By default, different colours will be used for plotting the units belonging to the different groups and their ellipses. Options USEPENS and CMATCH can be used to change the colours (see below).

Principal axes (also known as major and minor axes) can be added to the plot by setting the option PAXES to yes. By default, they are not plotted (PAXES=no). For the principal axes to be drawn at right-angles, the x- and y-axes must be identically scaled. This can be controlled through the FRAME directive, by setting the SCALING parameter to xyequal for the WINDOW in which the graph is to be plotted.

By default, the pen attributes are defined automatically within the procedure. However, you can set option USEPENS=yes to request that the current pen definitions are used. The following pens are used to draw the different elements in the plot:

pen=253	confidence ellipse,
pen=254	prediction ellipse,
pen=255	equal-frequency ellipse,
pen=256	principal axes,
pen=1...*N*	data (where *N* is the number of groups to be plotted).

When USEPENS=yes and groups are to be plotted (i.e. GROUPS is supplied or X and/or Y supplies a pointer), the CMATCH option is used to control the colour of the ellipses and principal axes drawn for each group. The default setting, CMATCH=group matches the colours of the ellipses and principal axes to those used for plotting the data from each group (i.e. the colours used to draw the ellipses and principal axes are controlled by pens 1...*N*, where *N* is the number of groups). Alternatively, CMATCH=pen draws the ellipses for all groups using the colour defined by the pen of the corresponding ellipse type (i.e. pen 253, 254 or 255), and the principal axes using the colour defined by pen 256. When USEPENS=yes, but there are no separate groups to be plotted, the colours used for drawing the different ellipse types and the principal axes are taken from pens 253, 254, 255 and 256, and the setting of CMATCH is ignored.

When the pen attributes are defined by the procedure (i.e. when USEPENS=no), the TFILL option can be used to fill the area inside the ellipses with transparent colours. It supplies a scalar, ranging from 0 (opaque) to 255 (completely transparent). By default, the ellipses are not filled, i.e. TFILL=255. (Note, when filled ellipses are wanted, TFILL=200 usually provides a good

level of transparency.) When using your own pen definitions (i.e. when USEPENS=yes), you can fill the ellipses by setting the METHOD parameter of the PEN directive to fill for the appropriate pen number. You can also control the transparency of the fill by using the TAREA parameter.

The number of points used to draw the ellipses is specified by the NPOINTS option (default 1000). Increasing NPOINTS will improve the accuracy (and visual smoothness) of the resulting ellipses, but it also increases the computational burden. However, it may be necessary sometimes to capture the curvature accurately around the vertices, especially when FILL=yes.

The TITLE, YTITLE and XTITLE parameters can supply an overall title, a y-axis title and a x-axis title for the plot, respectively. If no y- or x-axis titles are supplied, titles are generated automatically. To omit the axis title, a blank string can be supplied, i.e. YTITLE=' '. By default, no overall title is displayed.

The WINDOW option defines the window to plot the graph (default 1). Similarly the KEYWINDOW option defines the window to use for the key. By default, the key is drawn in window 255, which is defined to be on the right of the scatter plot. Setting KEYWINDOW=0 suppresses the key. Default labels are constructed automatically for the different groups of data (i.e., when GROUPS is supplied, or X and/or Y supplies a pointer). However, you can use the DESCRIPTION parameter to supply as a text structure containing *N* labels of your own for the key. The KEYDESCRIPTION option can supply an overall description for the key. By default, no title is displayed for the key.

Options: PLOT, PROBABILITY, NPOINTS, DISPLAY, PAXES, TFILL, USEPENS, CMATCH, WINDOW, KEYWINDOW, KEYDESCRIPTION, SCREEN
Parameters: Y, X, GROUPS, DESCRIPTION, TITLE, YTITLE, XTITLE

Method

DELLIPSE uses the methods described in Sokal & Rohlf (1995).

Action with RESTRICT

When X and Y supply both variates, DELLIPSE takes account of restrictions on Y, X and GROUPS. However, if more than one are restricted, they must be restricted in the same way. Furthermore, their unrestricted lengths must be the same.

When X and/or Y supplies a pointer, any restrictions are ignored.

Reference

Sokal, R.R., & Rohlf, J.F. (1995). Chapter 15: Correlation. In *Biometry: The Principles and Practice of Statistics in Biological Research*, Third edition, 555-608. New York: W.H. Freeman & Company.

See also

Directives: DGRAPH, CORRELATE, PEN, FRAME, PCP.
Procedures: DCORRELATION, FCORRELATION.
Functions: CORRELATION.
Genstat Reference Manual 1 Summary section on: Graphics.

DEMC

Performs Bayesian computing using the Differential Evolution Markov Chain algorithm (W. van den Berg & R.W. Payne).

Options

PRINT = *string token*	What to print (results, monitoring, scatterplot, histogram); default resu, moni, scat, hist
CALCULATION = *expression*	Calculation(s) of logposterior, involving explanatory or pointer variate; if unset, this is calculated by the procedure specified by the PROCEDURE option
LOGPOSTERIOR = *scalar*	Identifier of scalar holding log-posterior within CALCULATION (must be set if CALCULATION is set)
MULTIPLE = *scalar*	Number of populations is number of parameters times MULTIPLE; default 3
UNIFORMLIMIT = *scalar*	Uniform random numbers are drawn from (-UNIFORMLIMIT, UNIFORMLIMIT) and added to candidate parameter sets; default 0.00001
DATA = *identifiers*	Data structures used in CALCULATION or by PROCEDURE
NGENERATIONS = *scalar*	Maximum number of iterations; default 1000
STEP1 = *scalar* or *variate*	Generations for which gamma is set to 1; default 0
FRACTIONBURNIN = *scalar*	Fraction of iterations used for burn-in; default 0.5
GRVARIANCE = *scalar* or *variate*	Variance to generate populations from initial values of the parameters; default 0.1
PERCENTAGES = *variate*	Percentages for which quantiles has to be calculated; default !(2.5, 25, 50, 75, 97.5)
PROCEDURE = *identifier*	Identifier of procedure to calculate LOGPOSTERIOR if CALCULATION is unset; default _DEMCLOGPOSTERIOR
SEED = *scalar*	Seed for the random numbers; default 0
NWINDOWS = *scalar*	Number of histograms and scatterplots per screen when plotting estimates and logposterior from all iterations
SDLOGPOSTERIOR = *scalar*	Saves the s.d. for LOGPOSTERIOR
QUANTILESLOGPOSTERIOR = *variate*	Saves quantiles for LOGPOSTERIOR
RHATLOGPOSTERIOR = *scalar*	Saves the convergence criterion for LOGPOSTERIOR
ALLLOGPOSTERIOR = *variate*	Saves the parameter estimates for LOGPOSTERIOR from all the iterations
IPOPULATIONS = *pointers*	Pointer to supply initial populations of the parameters and the corresponding log-posteriors
FPOPULATIONS = *pointers*	Pointer to save final populations of the parameters and the corresponding log-posteriors

Parameters

PARAMETER = *scalars*	Parameters to estimate
INITIAL = *scalars*	Initial values of the parameters; must be set unless IPOPULATIONS is set
SD = *scalars*	Standard errors of the estimates
QUANTILES = *variates*	Saves the quantiles for each parameter
RHAT = *scalars*	Convergence criteria
ALLESTIMATES = *variates*	Saves the parameter estimates from all the iterations

Description

DEMC uses the *Differential Evolution Markov Chain* algorithm of Ter Braak (2006) to do Bayesian computations by Markov chain Monte Carlo. The logarithm of the posterior density for each set of parameters can be calculated either by a list of expressions supplied by the CALCULATION option, or by a (user-defined) procedure whose name is specified by the PROCEDURE option (with default name _DEMCLOGPOSTERIOR). The names of the parameters and their initial values are specified by the PARAMETER and INITIAL parameters, respectively. Data structures containing information that is needed to calculate the log-posterior are supplied by the DATA option. Also, if you are using the CALCULATION option, you must define the identifier of the log-posterior (as used to store the results of the calculations) using the LOGPOSTERIOR option.

The number of populations of parameters to be generated is defined as the number of parameters multiplied by the value supplied by the MULTIPLE option (default 3). The Normal variance used to generate the initial population from the initial values is specified by the GRVARIANCE option. You can set this to a scalar to use the same variance for each parameter, or to a variate to define different variances for the parameters; by default GRVARIANCE=0.1. The fraction of the data used for burn-in is specified by the FRACTIONBURNIN option (default 0.5).

The NGENERATIONS option defines the number of generations to form from the populations, and the FRACTIONBURNIN option defines the proportion of these that are for burn-in. (The distributions of the parameters are determined only from the generations that are produced after burn-in is complete.) The SEED option defines a seed for the random numbers that are used within DEMC. The default value 0 continues from the previous random-number generation or (if none) initializes the seed automatically. Options UNIFORMLIMIT and STEP1, which control how the new populations are formed, are explained in the Method section.

Once the generations are complete, the identifiers defined by PARAMETER are defined as scalars containing the means of the parameters over the populations generated after burn-in. Standard deviations and convergence criteria for the parameters can be saved, in scalars, using the SD and RHAT parameters. If RHAT is greater than 1.1, say, for any parameter, the number of generations should be increased. The QUANTILES parameter allows to save a variate for each PARAMETER, containing quantiles at percentages specified by the PERCENTAGES option (by default 2.5, 25, 50, 75, 97.5). To study the parameter distributions in more detail, you can also use the ALLESTIMATES parameter to save variates containing all the values generated after burn-in for each PARAMETER. The LOGPOSTERIOR, SDLOGPOSTERIOR, RHATLOGPOSTERIOR, QUANTILESLOGPOSTERIOR and ALLLOGPOSTERIOR allow the equivalent information to be saved for the log-posterior.

The final populations and corresponding log-posteriors can be saved, in a pointer, by the FPOPULATIONS option. You can then restart DEMC from the current position, and run some more generations, by using this pointer as the setting of the IPOPULATIONS option. FPOPULATIONS[1...N] have number of units equal to the number of parameters d, while FPOPULATIONS[N1] has number of units equal to N, where N = MULTIPLE × d. This can cause problems if you try to save FPOPULATIONS[] using procedure EXPORT.

Options: PRINT, CALCULATION, LOGPOSTERIOR, MULTIPLE, UNIFORMLIMIT, DATA, NGENERATIONS, STEP1, FRACTIONBURNIN, GRVARIANCE, PERCENTAGES, PROCEDURE, SEED, NWINDOWS, SDLOGPOSTERIOR, QUANTILESLOGPOSTERIOR, RHATLOGPOSTERIOR, ALLLOGPOSTERIOR, IPOPULATIONS, FPOPULATIONS.

Parameters: PARAMETER, INITIAL, SD, QUANTILES, RHAT, ALLESTIMATES.

Method

DEMC uses the DE-MC algorithm of Ter Braak (2006) to perform Markov chain Monte Carlo (MCMC); see Congdon (2001, 2003), Gelman et al. (2004) or Lee (2003). The DE-MC algorithm combines the genetic algorithm called Differential Evolution (DE) with MCMC. The values of the INITIAL parameter are used to generate *n* parameter sets, by generating *d* independent Normal deviates with means INITIAL and variance GRVARIANCE. Here, *d* is the number of parameters, and *n* is *d* multiplied by the value of the MULTIPLE option.

For each parameter set *i* (*i*=1...*n*), the algorithm selects two other parameter sets at random, and calculates the differences between their parameter values, multiplied by a parameter γ and a random number taken from the uniform distribution on (-UNIFORMLIMIT, UNIFORMLIMIT); γ generally takes the value $2.38/\sqrt{(2\times d)}$, but the STEP1 option allows you to define generations in which γ takes the value 1 (by default there are none). These differences are then added to the parameter values in set *i* to form a new candidate set of values. The candidate set replaces set *i* if its log-posterior likelihood is greater than the log-posterior likelihood of set *i* + the logarithm of a random number from the uniform distribution on (0,1); see Ter Braak 2006).

References

Congdon, P. (2001). *Bayesian Statistical Modelling*. Wiley, Chichester, England

Congdon, P. (2003). *Applied Bayesian Modelling*. Wiley, Chichester, England.

Gelman, A., Carlin, J.B., Stern, H.S. & D.B. Rubin (2004). *Bayesian Data Analysis, 2nd Edition*. Chapman & Hall, London.

Lee, P.M. (2003). *Bayesian Statistics an Introduction, 3rd Edition*. Arnold, London.

Ter Braak, C.J.F. (2006) A Markov chain Monte Carlo version of the genetic algorithm Differential Evolution: easy Bayesian computing for real parameter spaces. *Statistics & Computing*, **16**, in press.

See also

Procedure: BGXGENSTAT.

Genstat Reference Manual 1 Summary section on: Bayesian methods.

DERRORBAR

Adds error bars to a graph (R.W. Payne).

Options

ORIENTATION = *string token*	Direction of the line (horizontal, vertical); default vert
BARCAPWIDTH = *scalars*	Width of the cap drawn at the ends of the error bar; default 1
WINDOW = *scalar*	Window in which to draw the bar; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2

Parameters

BARLENGTH = *scalars*	Lengths of the bars
Y = *identifiers*	Vertical coordinates for the midpoints of the bars
X = *identifiers*	Horizontal coordinates for the midpoints of the bars
PEN = *scalars*	Pen to use for each bar
LABEL = *texts*	Text to plot alongside each bar
YLPOSITION = *string tokens*	Position of each label in the y-direction (above, below, centre, center); default belo
XLPOSITION = *string tokens*	Position of each label in the x-direction (centre, center, left, right); default righ
PENLABEL = *scalars*	Pen to use for each label
DESCRIPTION = *texts*	Annotation for the key

Description

The DERRORBAR procedure plots error bars on a graph. The window containing the graph is specified by the WINDOW option (default 1). The ORIENTATION option controls whether the bars are horizontal (i.e. parallel to the x-axis) or vertical (i.e. parallel to the y-axis).

The BARLENGTH parameter defines the length of each bar, on the y-axis for a vertical line, or the x-axis for a horizontal line. The positions of their midpoints are specified by the Y and X parameters. If these are not set, a vertical bar will be plotted just inside the left-hand side of the window, and a horizontal bar will be plotted at the bottom of the window. The PEN parameter can specify the pen to use for each bar. If this is not set, pen 255 is used as a default, having first been defined to draw continuous black lines. The BARCAPWIDTH option specifies the size of the "caps" drawn at the ends of the bars.

The LABEL parameter allows you to plot a label alongside each bar. Its position is specified by the YLPOSITION and XLPOSITION parameters. The pen to use can be specified by the PENLABEL parameter. If this is not set, pen 256 is used as a default, having first been defined to omit any symbol and use the colour black.

The DESCRIPTION parameter can supply annotation to add to the key for each bar. The window for the key is specified by the KEYWINDOW option (default 2).

Options: ORIENTATION, BARCAPWIDTH, WINDOW, KEYWINDOW.
Parameters: BARLENGTH, Y, X, PEN, LABEL, YLPOSITION, XLPOSITION, PENLABEL, DESCRIPTION.

See also

Procedures: DARROW, DTEXT, DFRTEXT, DREFERENCELINE.
Genstat Reference Manual 1 Summary section on: Graphics.

DESCRIBE

Saves and/or prints summary statistics for variates (R.C. Butler & D.A. Murray).

Options

PRINT = *string token*	Controls whether or not the summaries are printed (summaries); default summ
SELECTION = *string tokens*	Selects the statistics to be produced (nval, nobs, nmv, mean, median, min, max, range, q1, q3, sd, sem, var, sevar, %cv, sum, ss, uss, skew, seskew, kurtosis, sekurtosis, all); default mean, min, max, nobs, nmv, medi, q1, q3
GROUPS = *factor*	Allows groups to be defined, so that summaries are produced for each group in turn

Parameters

DATA = *variates*	Data to summarize
SUMMARIES = *variates* or *pointers*	To save summaries for each DATA variate, in a variate if GROUPS is unset, or in a pointer to a set of variates (one for each group) if groups have been specified; will be redefined if necessary

Description

DESCRIBE calculates up to 22 different summary statistics for values stored in a variate. The statistics may be saved, or printed, or both. The statistics to be calculated are indicated by the SELECTION option; the available settings are:

nval	number of values
nobs	number of non-missing values
nmv	number of missing values
mean	arithmetice mean
median	median
min	minimum
max	maximum
range	range (max-min)
q1	lower quartile
q3	upper quartile
sd	standard deviation
sem	standard error of mean
var	variance
sevar	standard error of variance
%cv	coefficient of variation
sum	total of values
ss	corrected sum of squares
uss	uncorrected sum of squares
skew	skewness (see Method)
seskew	standard error of skewness
kurtosis	kurtosis (see Method)
sekurtosis	s.e. of kurtosis
all	all 22 summaries

by default the mean, min, max, nobs, nmv, median and both quartiles are calculated.

Printing is controlled by the PRINT option. The statistics are printed by default, so to suppress printing you need to put PRINT=*.

The GROUPS option allows groups of observations to be defined. Summaries are then given for each group.

The SUMMARIES parameter allows the statistics to be saved in a variate, or a pointer to a set of variates if there are groups. These need not be declared in advance. The units of the variate(s) are labelled by the corresponding strings from the settings (in capital letters) of the SELECTION option, to simplify the subsequent access of any individual statistic. For example, the minimum value can be copied from a SUMMARIES variate v into a scalar m by

```
CALCULATE m = v$['MIN']
```

Options: PRINT, SELECTION, GROUPS. Parameters: DATA, SUMMARIES.

Method

The statistics are calculated in a variate which is then restricted to print only those that were required, and to obtain the unit numbers of those to be copied into the SUMMARIES variate.

SE Variance is calculated as

$\sqrt{((N\,(M_4 - 4\,M_1\,M_3 + 6\,M_1\,M_1\,M_2 - 3\,M_1^4)/(N\text{-}1) - (N\,(M_2 - M_1\,M_1)/(N\text{-}1))^2)/N)}$

Skewness is calculated as $(M_3 - 3\,M_1\,M_2 + 2\,M_1^3) / (M_2 - M_1\,M_1)^{3/2}$

SE Skewness is calculated as $\sqrt{(\{6N\times(N-1)\}/\{(N-2)\times(N+1)\times(N+3)\})}$

Kurtosis is calculated as $(M_4 - 4\,M_1\,M_3 + 6\,M_1^2\,M_2 - 3\,M_1^4)/(M_2 - M_1\,M_1)^2 - 3$

SE Kurtosis is calculated as $\sqrt{(\{24N(N-1)^2\}/\{(N-2)(N-3)(N+5)(N+3)\})}$

where $M_i = \sum x^i / N$

and N = NOBSERVATIONS(DATA)

Action with RESTRICT

The statistics are calculated for the restricted set of units from each DATA variate. Any existing restrictions are not affected by the procedure.

See also

Directive: TABULATE.

Procedures: CDESCRIBE, PTDESCRIBE, TABMODE, VSUMMARY.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

DESIGN

Helps to select and generate effective experimental designs (R.W. Payne, M.F. Franklin & A.E. Ainsley).

Option

STATEMENT = *text*	Saves a command to recreate the design

No parameters

Description

DESIGN is a procedure which can be used interactively to form experimental designs of various types. The process involves answering questions, posed by Genstat, first to select the particular type of design, then to give details such as names of factors, numbers of treatments, and so on. A range of subsidiary procedures may be called, depending on the type of design selected. If you wish to avoid some of the question-and-answer process, the subsidiary procedures can also be called directly. They all have options and parameters which provide an alternative way of supplying the information otherwise obtained by the various questions and, provided you supply all the required information, they can also be used in batch. The STATEMENT option of DESIGN allows you to save a Genstat text structure containing a command to use the relevant subsidiary procedure, and setting all the options and parameters required to recreate the design.

There are 18 types of design.

Orthogonal hierarchical designs – designs such as randomized blocks, split-plots, split-split-plots, &c.

Complete factorial designs (with interactions confounded with blocks) – these are available for treatments that all have the same number of levels k, where k is a prime number or a power of a prime number. The design, constructed by procedure AGFRACTION, will be a minimum-aberration design. (To explain this, we first define the resolution of a design as the largest integer r such that no interaction term with r factors is confounded with blocks. The aberration of the design is the number of interaction terms with r+1 factors that are confounded. A minimum aberration design is defined as a design with the smallest aberration out of the designs with the highest available resolution. So, essentially this selects the best design by minimizing the number of interactions with the minimum number of factors that are confounded.)

Fractional factorial designs (with blocking if required) – these are formed by AGFACTORIAL by taking one block of a minimum-aberration factorial design. If required, the resulting fractional factorial can be further dividing into its own blocks.

Factorial designs from a repertoire (with blocking) – these have several treatment factors and a single blocking factor (giving strata for blocks and plots within blocks). The blocks are too small to contain a complete replicate of the treatment combinations and so various interaction are confounded with blocks. (See procedure AGDESIGN.)

Fractional factorial designs from a repertoire (with blocking) – again there are several treatment factors but the design does not contain every treatment combination and so some interactions are aliased; there can also be a blocking factor and some interactions will then be confounded with blocks. (See procedure AGDESIGN.)

Lattice designs – designs for a single treatment factor with number of levels that is the square of some integer k. The design has replicates, each containing k blocks of k plots, and different treatment contrasts can be confounded with blocks in each replicate. (See procedure AGSQLATTICE.)

Lattice squares – these are similar to lattices except that the blocking structure with the replicates has rows crossed with columns; again different treatment contrasts can be confounded with the rows and columns in each replicate. (See procedure AGSQLATTICE.)

Latin squares – designs are available for any number of treatments (subject to workspace limitations) also, where feasible, more than one orthogonal treatment factor can be generated to form Graeco-Latin squares etc. (see procedure AGLATIN).

Latin squares balanced for carry-over effects – these are relevant when the same plots or subjects are treated during several successive time periods, and there is interest both in the direct effect of a treatment during the period in which it is applied and its carry-over (or "residual") effect during later periods (see procedure AGCROSSOVERLATIN).

Semi-Latin squares – $n \times n$ Latin squares whose individual plots are split into k sub-plots to cater for a treatment factor with $n \times k$ levels; three types are available Trojan squares, interleaving Latin squares and inflated Latin squares (see procedure AGSEMILATIN).

Complete and quasi-complete Latin squares – Latin squares designed to guard against interference between plots; a complete Latin square is a Latin square in which each ordered pair of treatments appears exactly once within the rows of the square, and exactly once within the columns; a quasi-complete Latin has similar properties, but here each unordered pair occurs exactly twice within the rows, and exactly twice within the columns (see procedure AGQLATIN).

Alpha designs – these again have a single treatment factor but there is no constraint on the number of levels; the blocking structure has replicates and blocks within replicates. Further details are given in the description of the procedure AFALPHA or by (Patterson & Williams 1976).

Cyclic designs – these are designs with a single blocking factor which defines blocks that are too small to contain every treatment. Usually there is a single treatment factor, but you can also generate the cyclic superimposed designs of Hall & Williams (1973) in which there are two treatment factors and the treatment structure fits only the main effects. An alternative refinement (Davis & Hall 1969) has a crossed blocking structure generally taken to represent subjects*time. Details of the cyclic process by which the treatment levels are generated can be found in the description of the procedure AFCYCLIC.

Balanced-incomplete-block designs – designs where the experimental units are grouped into blocks such that every pair of treatments occurs in an equal number of blocks. All comparisons between treatments are thus made with equal accuracy, so the design is balanced and, in particular, can be analysed by ANOVA. Further details are given in the description of procedure AGBIB.

Neighbour-balanced designs – designs that allow an adjustments to be made for the effect that a treatment may have on adjacent plots. Further details are given in the description of procedure AGNEIGHBOUR.

Central composite designs – used to study multi-dimensional response surfaces; see procedure AGCENTRALCOMPOSITE.

Box-Behnken designs – used to study multi-dimensional response surfaces; see procedure AGBOXBEHNKEN.

Plackett Burman (main effect) designs – for estimating main effects of factors with two levels, using a minimum number of experimental units (Plackett & Burman 1946). Further details are given in the description of procedure AGMAINEFFECT.

Loop designs – for use e.g. in time-course microarray experiments; see procedure AGLOOP.

Reference-level designs – for use e.g. in two-colour microarray experiments; see procedure AGREFERENCE.

You will be asked to provide a seed to be used to randomize the design and then given the opportunity to print a plan. If the design can be analysed by ANOVA, the procedures will define appropriate block and treatment formulae and then ask if you want to see the skeleton analysis-of-variance table (containing just source of variation, degrees of freedom and efficiency factors). Whether or not you choose to print any of this information, at the end of the whole process all the block and treatment factors necessary to define the design will be available – and they will

have the identifiers that you have supplied in response to the various questions asked by the procedures.

Option: STATEMENT. Parameters: none.

Method

The QUESTION procedure is used to find out what design is required. DESIGN then calls either AGHIERARCHICAL (for an orthogonal hierarchical design), AGFACTORIAL (for minimum-aberration complete or fractional factorial designs), AGDESIGN (for a factorial design), AGFRACTION (for a fractional factorial design), AGLATIN (for mutually orthogonal Latin squares), AGCROSSOVERLATIN (for Latin squares balanced for carry-over effects), AGSEMILATIN (for a semi-Latin square), AGQLATIN (for complete and quasi-complete Latin squares), AGSQLATTICE (for a square lattice or lattice square design), AGALPHA (for an alpha design), AGCYCLIC (for a cyclic design), AGNEIGHBOUR (for a neighbour-balanced design), AGCENTRALCOMPOSITE (for a central composite design), AGBOXBEHNKEN (for a Box-Behnken design), AGMAINEFFECT (for a Plackett Burman main effect design), AGLOOP (for a loop design) or AGREFERENCE (for a reference-level design). The designs are generated using GENERATE and the other standard Genstat directives for calculation and manipulation. Some of the information needed to specify the designs is stored in backing-store files, and much of this was adapted from the standard designs of the program DSIGNX (Franklin & Mann 1986).

References

Davis, A.W. & Hall, W.B. (1969). Cyclic change-over designs. *Biometrika*, **56**, 283-293.

Franklin, M.F. & Mann, A.D. (1986). *DSIGNX a Program for the Construction of Randomized Experimental Plans*. Scottish Agricultural Statistics Service, Edinburgh (revised edition).

Hall, W.B. & Williams, E.R. (1973). Cyclic superimposed designs. *Biometrika*, **60**, 47-53.

Patterson, H.D. & Williams E.R. (1976). A new class of resolvable incomplete block designs. *Biometrika*, **63**, 83-92.

Plackett, R.L. & Burman, J.P. (1946). The design of optimum factorial experiments. *Biometrika*, **33**, 305-325 & 328-332.

See also

Genstat Reference Manual 1 Summary section on: Design of experiments.

DFOURIER

Performs a harmonic analysis of a univariate time series (G. Tunnicliffe Wilson & R.P. Littlejohn).

Options

PRINT = *string tokens*	Controls printed output (accumulated, means, tsm); default *
PLOT = *string tokens*	What to plot (periodogram, harmonics, means, residuals, cumulative, range); default peri, harm, mean, resid
MODELTYPE = *string token*	What harmonic regression model to fit (none, best, full); default none
GROUPS = *factor*	Groups for plot of means
ORDER = *variate*	Order for time series model; default !(1,0,0)
COLOURS = *text* or *variate*	Colour for each level of GROUPS
FACSHORTCYCLE = *factor*	Factor giving levels of the short cycle
NCOMPONENTS = *scalar*	Number of nested cycles, must be 0, 1, or 2; default 0
SHORTCYCLE = *scalar*	Length of the short cycle; default 24
LONGCYCLE = *scalar*	Length of the long cycle; default 365.225
LABSHORTCYCLE = *text*	Label for the short cycle; default 'daily'
LABLONGCYCLE = *text*	Label for the long cycle; default 'annual'
NHSHORTCYCLE = *scalar*	Number of harmonics for the short cycle; default 5
NHLONGCYCLE = *scalar*	Number of harmonics for the long cycle; default 3
RANGE = *variate*	Variate with two values, defining the frequency range within [0,0.5] to draw a portion of the periodogram

Parameters

DATA = *variates*	Time series
PERIODOGRAM = *variates*	Saves the periodogram of DATA
FREQUENCY = *variates*	Saves the frequencies at which the periodogram is calculated
MEANS = *tables*	Saves the table of means of the fitted model for each value of FACSHORTCYCLE by each level of GROUPS
RESIDUALS = *variates*	Saves the residuals from the fitted model
FITTEDVALUES = *variates*	Saves the fitted values from the model

Description

DFOURIER performs a harmonic analysis for a univariate time series which is supplied, in a variate, by the DATA parameter. In its basic form, it can produce 3 pages of graphs to study the series. These graphs are all controlled by the PLOT option. Setting PLOT=periodogram produces a page of graphs showing the time series, its periodogram and its log periodogram. The frequencies for the periodogram are calculated internally, and noted in the output. These can be saved, in a variate, by the FREQUENCY parameters, and the PERIODOGRAM parameter can save the periodogram. The cumulative setting of PLOT plots the cumulative periodogram (on a separate page), and the range setting plots the periodogram over the range specified by the RANGE option (this must be a value within [0,0.5]).

Other graphs are useful if you anticipate that the series will show some specific components. The number of these components is specified by the NCOMPONENTS option, and may be either 0 (no components, the default), 1 (a "short" cycle) or 2 (a "short" and a "long" cycle). The lengths of the long and short cycles are specified by the LONGCYCLE and SHORTCYCLE options, respectively. The defaults 365.225 and 24, correspond to hourly measurement of annual and

daily cycles. The LABLONGCYCLE and LABSHORTCYCLE options supply labels for these cycles for the plots, with defaults of 'annual' and 'daily' respectively.

The components are particularly useful for analysing meterological time series (such as air temperatures) measured hourly over several years, where you want to describe how the diurnal pattern varies throughout the year. A single (non-sinusoidal) periodic component with period p (e.g. $p = 24$ for hourly observations) produces a main spike in the periodogram at the frequency $f = 1/p$, followed by a series of diminishing spikes at integer multiples of f known as *harmonics*.

When there are two periodic components with interacting rhythms, signals are observed in the periodogram not only at harmonics of each frequency, but at integer differences of the lower frequency from the higher. Thus, if hourly and annual frequencies are denoted by f_d and f_a, spikes may be observed in the periodogram at

$$f_{da} = n \times f_d + m \times f_a,$$

where n is a non-negative integer, and m is an integer, which must be positive when n is zero.

These spikes generated by the interaction are generally hard to discern in an ordinary graph of the periodogram. The harmonic setting of PLOT produces a trellis plot that zooms in on a narrow range of about $n \times f_d$, for integer values of n ranging from 1 up to a value defined by the NHSHORTCYCLE option. This can be set to either 5 (default), 7 or 8, producing respectively a 3 × 2, 4 × 2 or 3 × 3 array of graphs. The NHLONGCYCLE option specifies the number of vertical lines to be drawn, within each graph, at positions corresponding to differences due to the long cycle. This can be set to 0, 1, 2 or 3 (default). It should be set to 0 if there is only one periodicity in the sampling protocol. The y-axes of the plots are scaled individually to a suitable order of magnitude, which is denoted in each graph. The frequency range for each panel is

$$n \times f_d +/- 5.1 \times f_a.$$

The MODELTYPE option allows a harmonic regression analysis to be conducted on DATA. The setting full fits sine and cosine terms for each frequency indicated in the harmonics graph. Alternatively, the setting best fits the full model and then drops terms that are non-significant at the 5% level. This does not guarantee that all terms remaining in the model are necessarily significant at the 5% level. In practice, however, dropping these additional terms will usually make little difference to the fitted model or residual variance. The accumulated setting of the PRINT option prints the accumulated analysis of deviance table from the fit.

With the tsm setting of the PRINT option, the model fitted as above is then used as the TRANSFERFUNCTION in a time series analysis of DATA. The TSM is defined by the ORDER option; by default this is set to a first-order autoregression (i.e. ORDER=!(1,0,0)). Note that this may take a long time to fit if there are many missing values in the data.

The fitted values and residuals from the final model (tsm is fitted after best, which is fitted after full) can be saved by the FITTEDVALUES and RESIDUALS parameters. The residuals setting of PLOT produces time-series plots of the residuals, from the BJIDENTIFY procedure.

DFOURIER forms tables of means of the fitted values classified by the the short cycle component and another factor, specified by the GROUPS option. You can supply the short cycle factor using the FACSHORTCYCLE option; this must have SHORTCYCLE levels or a fault will be generated. If FACSHORTCYCLE is unset, the required factor will be internally generated with levels 1...SHORTCYCLE. The factor GROUPS may, for example, be month or season. The SHORTCYCLE factor should be nested within GROUPS to provide meaningful output, but no checks are carried out on this.

You can plot the means using the means setting of the PLOT option. The points in each group are plotted in different colours, and you can supply these using the COLOURS option. If COLOURS is unset, the colours are set by default. If GROUPS has 4 levels, it is assumed they correspond to season, and pens 1 to 4 are defined to be red, gold, blue and green, corresponding to summer, autumn, winter and spring. If GROUPS has 12 levels, it is assumed that they correspond to months, and pens 1 to 12 are given decreasing intensities within the seasonal shades in clusters of three. Thus pens 1 to 3 are given crimson, red and salmon for the summer

months. Note that this is tuned to a southern hemisphere calendar.

Options: PRINT, PLOT, MODELTYPE, GROUPS, ORDER, COLOURS, FACSHORTCYCLE, NCOMPONENTS, SHORTCYCLE, LONGCYCLE, LABSHORTCYCLE, LABLONGCYCLE, NHSHORTCYCLE, NHLONGCYCLE, RANGE.
Parameters: DATA, PERIODOGRAM, FREQUENCY, MEANS, RESIDUALS, FITTEDVALUES.

Action with RESTRICT
There must not be any restrictions.

See also
Directive: FOURIER.
Procedures: MCROSSPECTRUM, PERIODTEST, PREWHITEN, REPPERIODOGRAM, SMOOTHSPECTRUM.
Genstat Reference Manual 1 Summary section on: Time series.

DFRTEXT

Adds text to a graphics frame (W. van den Berg).

No options

Parameters

Y = *variates* or *scalars*	Vertical coordinates in the frame
X = *variates* or *scalars*	Horizontal coordinates in the frame
TEXT = *texts*	Text to plot
PEN = *scalars*, *variates* or *factor*s	Pens to use; default 1
YUPPER = *scalars*	Maximum vertical coordinate in the frame; default 1
XUPPER = *scalars*	Maximum horizontal coordinate in the frame; default 1

Description

The DFRTEXT procedure provides a convenient way of putting textual annotation or description onto a graphics frame, similarly to the way in which the DTEXT procedure can add text to a plot inside the frame. The text to plot is specified by the TEXT parameter. This can be either a single string, or a Genstat text structure containing several lines of text. The Y and X parameters specify the coordinates where the text is to be plotted, with scalars for a single string or line, or with variates for several lines. These coordinates relate to the "standard device coordinates" used to define windows within the frame (see the FRAME directive), and must lie between 0 and the values supplied by the YUPPER and XUPPER parameters. The PEN parameter specifies the pen or pens to use (default 1).

Options: none.
Parameters: Y, X, TEXT, PEN, YUPPER, XUPPER.

Method

The FRAME directive is used to define a window, without margins, filling the whole screen, with maximum values in the x- and y-dimensions defined by the XUPPER and YUPPER parameters, respectively. The x- and y-axes in that window are defined to go from 0 to XUPPER and YUPPER, using the XAXIS and YAXIS directives. The DGRAPH directive is then used to plot the text.

Action with RESTRICT

DFRTEXT takes account of restrictions on any set of Y, X and TEXT parameters.

See also

Procedures: DTEXT, DARROW, DERRORBAR, DREFERENCELINE.
Genstat Reference Manual 1 Summary section on: Graphics.

DFUNCTION

Plots a function (R.W. Payne).

Options

FUNCTION = *expression*	Function to plot
TITLE = *text*	Title for the plot; default shows the function
COLOUR = *text* or *scalar*	Colour of the function curve; default 'green'
WINDOW = *scalar*	Which graphics window to use; default 3
ELEVATION = *scalar*	Elevation of the viewpoint for the surface that is plotted when there are two arguments; default 25 (degrees)
AZIMUTH = *scalar*	Rotation about the horizontal plane for the viewpoint of a surface plot; default 225 (degrees)
DISTANCE = *scalar*	Distance of the viewpoint of a surface plot from the centre of the grid on the base plane; default * gives a distance of 100 times the maximum of the x-range and the y-range
ZSCALE = *scalar*	defines the scaling of the z-axis relative to the horizontal (x-y) axes in a surface plot; default 1
SCREEN = *string token*	Whether to clear the screen before plotting (clear, keep, resize); default clea

Parameters

ARGUMENT = *scalars*	Arguments of the function
LOWER = *scalars*	Lower values of the arguments for the plot
UPPER = *scalars*	Upper values of the arguments for the plot
STEP = *scalars*	Steps at which to evaluate the function

Description

DFUNCTION plots a function using high-resolution graphics. The function is specified, as a Genstat expression, using the FUNCTION option.

The arguments of the function (against which the function is to be plotted) are specified by the ARGUMENT parameter. There can be either one or two arguments, and they must each be either a scalar or an undeclared data structure. With one argument a curve is plotted using DGRAPH, while with two arguments a surface is plotted using DSURFACE. The LOWER and UPPER parameters specify the lower and upper values of the arguments to show in the plot. The STEP parameter specifies the step between the values of each ARGUMENT at which the FUNCTION is evaluated in the plot. The default generates 1001 values of a single argument, or 101 values of each argument if there are two.

The TITLE option can supply a title for the plot. If this is not specified, a title is constructed automatically to show the FUNCTION expression. The COLOUR option specifies the colour to use for the function curve or suface. The default is to plot it in green. The WINDOW option specifies which graphics window to use. The default is to use window 3, which should fill the whole frame. The SCREEN option controls whether the screen is cleared before plotting, or whether is is kept, or kept with resizing, from previous plots. By default the screen is cleared.

Options ELEVATION, DISTANCE, AZIMUTH and ZSCALE are relevant only when a surface is plotted (i.e. when there are two arguments), and all operate as in the DSURFACE directive. The ELEVATION, DISTANCE and AZIMUTH options specify the position of the viewpoint, and the ZSCALE option specifies a scaling factor for the z-axis that is used for the function, versus the x- and y-axes that are used for the arguments. See DSURFACE for further details.

Options: FUNCTION, TITLE, COLOUR, WINDOW, ELEVATION, AZIMUTH, DISTANCE, ZSCALE, SCREEN.
Parameters: ARGUMENT, LOWER, UPPER, STEP.

See also
Directives: CALCULATE, EXPRESSION.
Procedure: DTEXT.
Genstat Reference Manual 1 Summary section on: Graphics.

DHSCATTERGRAM

Plots an h-scattergram (D.A. Murray).

Options

LAGCLASS = *scalar* or *variate*	The lag classes to be displayed in the plots; default all lag classes
ARRANGEMENT = *text*	Specifies whether to display the plots individually or with multiple plots on the same page (single, multiple); default mult

Parameters

DATA = *variates*	Observations as a variate
LAGPOINTS = *pointers*	Lag classes, indexes to observations and directions for plotting

Description

DHSCATTERGRAM plots an h-scattergram of all values of z(x) against z(x+h) within a lag class. H-scattergrams are useful for identifying outliers that can skew the average semivariance variance within a lag class. The plot displays a 1 to 1 reference line and the closer the points lie to this line the stronger the correlation and the smaller the semivariance.

The observations should be supplied using the DATA parameter within a variate. The data for the lag classes can be taken directly from the FVARIOGRAM directive. The parameter LAGPOINTS corresponds to the parameter with the same name in FVARIOGRAM. The elements of LAGPOINTS contain:

1. variate of lag classes
2. variate of indices for z(x)
3. variate of indices for z(x+h)
4. factor of directions

By default an h-scattergram is produced for every lag class. However, you can select a subset of these by supplying the lag numbers in either a scalar or variate using the LAGCLASS option. The ARRANGEMENT option controls whether the plots are each drawn on separate pages or as a multiple plot in a 4 by 4 or 9 by 9 arrangement.

Options: LAGCLASS, ARRANGEMENT.
Parameters: DATA, LAGPOINTS.

Action with RESTRICT

If the data variates are restricted, only the units not excluded by the restriction will be used.

Reference

Webster, R. & Oliver, M.A. (2007). *Geostatistics for Environmental Scientists, 2nd edition*. Wiley, Chichester.

See also

Directives: FVARIOGRAM, KRIGE.
Genstat Reference Manual 1 Summary sections on: Spatial statistics, Graphics.

DHELP

Provides information about Genstat graphics (S.A. Harding).

No options

Parameter

TOPIC = *string tokens*	Lists the required graphics topics (current, possible); default poss

Description

DHELP provides information about the Genstat high-resolution graphics. It has a single parameter called TOPIC, which supplies a list of strings indicating the topics about which you want information. The setting current gives details about the current settings of the graphics frames, windows, axes and pens, including the negatively numbered pens used as initial defaults. The other setting possible indicates the available frames, windows, axes and pens, and lists the available graphics devices (indicating which one is currently selected).

Options: none.
Parameter: TOPIC.

Method

The information is obtained using the DKEEP directive, and the SAVE parameters of the FRAME, PEN, XAXIS, YAXIS and ZAXIS directives.

See also

Directive: HELP.
Genstat Reference Manual 1 Summary section on: Graphics.

DIALLEL

Analyses full and half diallel tables with parents (J.F. Potter).

Options

PRINT = *string tokens*	Controls printed output (data, vrwr, regression, aov, means, griffingaov); default data, vrwr, regr, aov, mean
LABELS = *text*	Labels for rowcols, one text value for each, column *j* has the same label as row *j*, so each value of LABELS is the label for a pair of parents, applying to a rowcol; default 1...*N*, where *N* is the dimension of each diallel table
METHOD = *string token*	Whether to perform full or half diallel analysis (half, full); default full

Parameter

DATA = *matrices*	Each matrix contains the data for one block in the analysis, half diallel tables are presented as square matrices with the upper triangles and leading diagonals containing the values of interest, the matrices must be of the same size

Description

DIALLEL performs analysis of variance of full diallel tables (Hayman 1954) and half diallels (Jones 1965). Work on variance and covariance relationships is also performed (Jinks 1954). The data are specified by the DATA parameter, in a square matrix for every block in the analyses. Half diallel tables are presented as square matrices with the upper triangle and leading diagonal containing the values of interest. The PRINT option controls printed output:

data	data values,
vrwr	variances and covariances of rowcols,
regression	regression of the variances on the covariances,
aov	analysis of variance table,
means	means,
griffingaov	analysis of variance defined by Griffing (1956), which provides estimates of general combining ability (GCA) and specific combining ability (SCA).

The LABELS option can give a text to be used for labelling rowcols (called arrays in the literature). The METHOD option specifies whether analysis of full or half diallels is required.

Options: PRINT, LABELS, METHOD.
Parameter: DATA.

Method

DIALLEL performs analysis of variance of full diallel tables, according to the method of Hayman (1954), and half diallels, according to the method of Jones (1965). A diallel table is a representation of the results of crossing a set of male and female homozygous parents in all possible combinations, including male:female reciprocation in full diallels. DIALLEL expects parent values (selfs) to be present as the leading diagonal of the table (whether a full or half matrix).

The analysis of variance estimates the following genetic components of variation.

a: variation between mean effects of each parental line. Genetically this provides a test of additive variation, but also detects dominance if asymmetry present, i.e. if alleles at any

one locus are not equally frequent (Hayman 1954).

b: variation caused by dominance at some of the loci. This term splits into:

b1: if significant this shows that dominance is largely uni-directional;

b2: estimates the asymmetry mentioned in *a*;

b3: signifies that some dominance is peculiar to individual crosses; If the symmetry condition is met, *b1* and *b3* together give a test of dominance equivalent to *b*.

c: variation between average maternal effects of each parental line.

d: variation in the reciprocal differences not attributable to *c*.

t: total variation.

Components *c* and *d* are reciprocal effects not available in half diallels. In the absence of replication, the *d* term should be used as the error term for testing components *a* to *c* in the full diallel. In the Griffing analysis, *a* corresponds to GCA, and *b* corresponds to SCA.

DIALLEL can also analyse over any number of blocks, in which case block effects are also estimated, and block interactions with the above components can then be used as estimates of error to test the significance of the components.

Variances of rowcols (*Vr*) are compared with the covariance of the rowcols (*Wr*) with the corresponding concurrent parents, using the method of Jinks (1954). This entails the regression of *Wr* on *Vr*, which gives measures of adequacy of the model, average dominance, and the distribution of dominant and recessive genes. The analysis of diallel tables is more fully described by Mather and Jinks (1971).

Many other diallel methods exist, DIALLEL representing quite a complex one, but one which makes fairly limiting assumptions, e.g. only a reference population in Hardy-Weinberg equilibrium with respect to individual loci and linkage equilibrium with respect to all pairs of loci can legitimately be used to estimate the genetic variance components. This means a large population reproducing by panmixia without selection. This and other difficulties such as the need for distinction between ancestral and descendant reference populations are discussed by Wright (1985).

Action with RESTRICT

Restrictions are ignored for text LABELS and are not relevant for DATA, which is of type matrix.

References

Griffing, B. (1956). Concept of general and specific combining ability in relation to diallel crossing system. *Aust. J. Biol.*, **9**, 463-493.

Hayman, B.I. (1954). The Analysis of Variance of Diallel Tables. *Biometrics*, **10**, 235-244.

Jones, R.M. (1965). Analysis of Variance of the Half Diallel Table. *Heredity*, **20**, 117-121.

Jinks, J.L. (1954). The Analysis of Continuous Variation in a Diallel Cross of Nicotiana rustica Varieties. *Genetics*, **39**, 767-788.

Mather, K. & Jinks, J.L. (1971). *Biometrical Genetics*, 249-284. Chapman & Hall Ltd.

Wright, A.J. (1985). Diallel Designs, Analyses, and Reference Populations. *Heredity*, **54**, 307-311.

See also

Procedure: FDIALLEL.

Genstat Reference Manual 1 Summary sections on: Analysis of variance, REML analysis of linear mixed models.

DILUTION

Calculates Most Probable Numbers from dilution series data (M.S. Ridout & S.J. Welham).

Options

PRINT = *string tokens*	Output required (estimates, fitted); default esti, fitt
%LIMITS = *scalar*	Percentage points for confidence limits; default 95
RMETHOD = *string token*	Which type of residuals to form (deviance, Pearson); default devi
MAXCYCLE = *scalar*	Maximum number of iterations allowed for the Newton-Raphson algorithm to converge; default 10
TOLERANCE = *scalar*	Defines the convergence criterion; default 0.0005

Parameters

POSITIVE = *variates*	Number of positive subsamples at each dilution
NSAMPLE = *variates*	Total number of subsamples tested at each dilution
VOLUME = *variates*	Volume of original sample present in each dilution
FITTED = *variates*	To store the fitted values
RESIDUAL = *variates*	To store the residuals, as specified by option RMETHOD
MPN = *scalars*	To store the maximum likelihood estimate of Most Probable Number
UPPER = *scalars*	To store the upper confidence limit for MPN
LOWER = *scalars*	To store the lower confidence limit for MPN
DEVIANCE = *scalars*	To store the residual deviance
PEARSONCHISQUARE = *scalars*	To store Pearson's chi-square statistic
DF = *scalars*	To store the degrees of freedom for goodness-of-fit tests (zero if no test is available)

Description

A dilution series experiment seeks to estimate the number of organisms in a sample. This is done by preparing successive dilutions of the original sample (usually with a constant dilution factor at each stage), and then testing for the presence/absence of organisms in several subsamples at each dilution. Under certain assumptions, discussed, for example, by Cochran (1950), it is then possible to estimate, by maximum likelihood, the number of organisms in the original sample. In the context of dilution series data, the maximum likelihood estimator is usually known as the Most Probable Number (MPN) of organisms.

DILUTION calculates the MPN estimator, together with likelihood-based confidence limits for the number of organisms.

The number of positive subsamples at each dilution (i.e. the number of subsamples which show the presence of organisms) must be specified in a variate using the parameter POSITIVE. The total number of subsamples used at each dilution, and the volume of the original sample used at each dilution, must be supplied in variates using parameters NSAMPLE and VOLUME.

Output is controlled by the PRINT option. The estimate setting produces the MPN estimate and associated confidence limits, together with the deviance and Pearson's chi-square statistic. The fitted setting gives observed and fitted values with residuals. All this information is produced by default. The range of the confidence limits can be set by option %LIMIT, the default being 95% limits, and the type of residuals produced (deviance or Pearson) is controlled by the RMETHOD option.

Both the MPN estimator and the confidence limits are calculated iteratively. Option MAXCYCLE sets the maximum number of iterations allowed in each case, the default being 10. Option TOLERANCE specifies the convergence criterion for the MPN estimator; the estimation

process is considered to have converged when the absolute value of the derivative of the log-likelihood is less than TOLERANCE. The default value of TOLERANCE is 0.0005. The iterative calculation of the confidence limits is considered to have converged when the log-likelihood takes the correct value to 2 decimal places.

All the information generated can be saved using parameters of the procedure: MPN saves the estimate; UPPER and LOWER save the upper and lower confidence limits; DEVIANCE, PEARSONCHISQUARE and DF save the goodness of fit statistics and the degrees of freedom; and the fitted values and residuals are saved by FITTED and RESIDUAL.

Options: PRINT, %LIMITS, RMETHOD, MAXCYCLE, TOLERANCE.
Parameters: POSITIVE, NSAMPLE, VOLUME, FITTED, RESIDUAL, MPN, UPPER, LOWER, DEVIANCE, PEARSONCHISQUARE, DF.

Method
The Newton-Raphson algorithm is used to find both the MPN and the appropriate confidence limits.

Action with RESTRICT
If any of POSITIVE, NSAMPLE or VOLUME are restricted (these restrictions must be compatible), then only the restricted set of units will be used.

Reference
Cochran, W.G. (1950). Estimation of bacterial densities by means of the 'most probable number'. *Biometrics*, **6**, 105-116.

See also
Procedures: PROBITANALYSIS, WADLEY.
Genstat Reference Manual 1 Summary section on: Regression analysis.

DIRECTORY

Prints or saves a list of files and/or subdirectories with names matching a specified mask (D.B. Baird).

Options

PRINT = *string tokens*	What to print (filenames, subdirectories); default file
SAVEPATH = *string token*	Whether to include the path in FILENAMES (yes, no); default no
MASKTYPE = *string token*	The type of mask specified by MASK (file, directory); default file

Parameters

MASK = *texts*	Mask identifying the files that are to be included in the each listing, if no directory path is included, the current working directory is searched; default '*.*'
FILENAMES = *texts*	Saves the list of files that match each mask
SUBDIRECTORIES = *texts*	Saves the list of subdirectories that match each mask

Description

DIRECTORY obtains lists of files or subdirectories matching a specified template. The file and subdirectory lists can be saved in the texts supplied by the FILENAMES and SUBDIRECTORIES parameters respectively. The SAVEPATH option controls whether the filenames also include the path.

Printed output is controlled by the PRINT option, with settings:

filenames	to print the list of files, and
subdirectories	to print the subdirectories.

The template is specified in a text with a single value, using the MASK parameter. It uses the standard conventions of the DOS DIR command but, for convenience, you can put / instead of \, as in other Genstat file specifications. So, for example, you can list the files with a suffix .GEN in the folder (or directory) c:\genprog by putting either

```
DIRECTORY 'c:\\genprog\\*.gen'
```

or

```
DIRECTORY 'c:/genprog/*.gen'
```

(The doubling of each symbol \ within the string is required as usual to tell Genstat that it should not be treated as the continuation symbol.) If no directory is specified in the template (e.g. MASK='*.gen'), the current working directory is searched.

If the MASKTYPE option is set to directory then the MASK parameter is interpreted as being a directory, and all the files or subdirectories within this directory are returned. For example the following program prints and saves all the files within the subdirectories of the current working directory

```
GET [WORKINGDIRECTORY=CDir]
DIRECTORY [PRINT=*;MASKTYPE=directory] CDir; SUBDIRECTORIES=Subs
FOR [INDEX=i] SD=#Subs
    CONCAT [NEWTEXT=SDir] CDir,'/',SD
    DIRECTORY [MASKTYPE=directory] SDir; FILE=Files[i]
ENDFOR
```

Options: PRINT, SAVEPATH, MASKTYPE.
Parameters: MASK, FILENAMES, SUBDIRECTORIES.

Method
The file list is obtained by a call to `SUSPEND` with a `DIR` command.

See also
Directive: `SUSPEND`.
Procedure: `%CD`.

DISCRIMINATE

Performs discriminant analysis (L.H. Schmitt & P.G.N. Digby).

Options

PRINT = *string tokens*	Printed output from the analysis (counts, lrv, tests, ccorrelations, icorrelations, correlations, adjustments, means, gdistances, scores, distances, newgroups, table, validation); default coun
NROOTS = *scalar*	The number of dimensions to be used for printed and saved output, and used in calculating the distances and the allocation of units; default is to use the full dimensionality
REALLOCATE = *string token*	Whether units from the training set are to be reallocated to groups (no, yes); default no
PLOT = *string tokens*	Features for the plots (means, mlabels, scores, polygons, confidencecircle); default mean, scor, poly (Note: * suppresses plotting)
VALIDATIONMETHOD = *string token*	Validation method to use to calculate error rates (bootstrap, crossvalidation, jackknife); default cros
NSIMULATIONS = *variate*	Number of bootstraps or cross-validation sets to use for selection and for validation; default !(10,50)
NCROSSVALIDATIONGROUPS = *scalar*	Number of groups for cross-validation, default 10
SEED = *scalar*	Seed for random number generation; default 0
YROOT = *scalars*	Specifies roots for plotting on y-axes
XROOT = *scalars*	Specifies roots for plotting on x-axes
TITLE = *strings*	Titles for plots
WINDOW = *scalars*	Windows for plots
SCREEN = *string tokens*	Action before each plot (keep, clear); default clea

Parameters

DATA = *pointers*	Each pointer contains a set of variates to be analysed
GROUPS = *factors*	Define groupings for the units in each training set, or missing values for the units to be allocated
NEWGROUPS = *factors*	Saves allocations (and reallocations)
ALLOCATION = *factors*	Saves allocations to groups including those not present in the training set
MEANS = *matrices* or *pointers*	Saves scores for group means
SCORES = *matrices* or *pointers*	Saves scores for units
DISTANCES = *matrices*	Saves unit to group-mean squared distances
LRV = *LRVs*	Saves the LRVs from the canonical variates analyses
ADJUSTMENTS = *matrices*	Saves adjustments to the canonical variates analyses
GDISTANCES = *symmetric matrices*	Saves the distances between groups
CCORRELATIONS = *matrices*	Saves canonical correlation coefficients
ICORRELATIONS = *symmetric matrices*	Saves within-group correlation matrices of the input variates

CORRELATIONS = matrices	Saves within-group correlations between the input and canonical variates

Description

DISCRIMINATE performs discriminant analysis (see, for example, Mardia, Kent & Bibby 1979).

The input for the procedure is given by a pointer and a factor, specified by the DATA and GROUPS parameters, respectively. The pointer contains a set of variates defining the attributes of the units. Any unit with a missing value in any of the variates is excluded from the analysis. Units can also be excluded from the analysis by restricting the factor or variates; any such restrictions must be consistent (the rules here are exactly as used by the FSSPM directive). The factor specifies the pre-defined groupings of the units from which the allocation is derived (the "training set"); the units to be allocated by the analysis have missing factor values.

Printed output is controlled by the option PRINT with settings:

counts	tables of the number of units in each group with a complete set of observations;
lrv	canonical variate loadings, latent roots and trace;
tests	chi-square tests (as given by CVA);
ccorrelations	canonical correlation coefficients (see Klecka 1980);
icorrelations	within-group correlation matrix of the input variates;
correlations	within-group correlations between the input and canonical variates;
adjustments	adjustments required to the canonical variate scores;
means	canonical variate scores for the group means;
gdistances	inter-group distances (as given by CVA);
scores	canonical variate scores for the units;
distances	Mahalanobis squared distances between the units and the group means;
newgroups	initial grouping and the allocation of units to groups;
table	tables of counts of allocations; and
validation	estimated error rates (see the VALIDATION option below).

The NROOTS option specifies how many dimensions are printed and retained for the latent roots and vectors, and for the scores of the means and units. The distances of the units from the group means, and thus the allocation of units, are also formed from the scores in the number of dimensions specified by NROOTS. By default, the results are for the full dimensionality, i.e. the smaller of the number of variates and one less than the number of groups.

The REALLOCATE option specifies whether the units in the training set are to be reallocated to groups by the procedure. If the default setting no is used then their group values, either printed or saved, will be missing.

The VALIDATIONMETHOD option specifies the validation method, with settings for cross-validation, jackknife and bootstrap. Cross-validation works by randomly splitting the units into a number of groups specified by the NCROSSVALIDATIONGROUPS option (default 10). It then omits each of the groups, in turn, and predicts how the the omitted units are allocated to the discrimination groups. Jackknifing leaves the units out one at a time, and uses the rest of the data to predict the group of the omitted unit. The bootstrap method works by drawing a bootstrap sample of units (a random sample of units with replacement of the same size as the original sample), and predicting the units that are not present in the random sample. The resulting bootstrap error rate is then calculated as a weighted average of the error rate of the omitted observations and the predictive error rate of the bootstrap sample. The weights used are 0.632 and 0.368 respectively, and so this is known as the *632 rule*.

The NSIMULATIONS option sets the number of simulations for cross-validation or bootstrapping. It should be set to a variate with two values: the first value defines the number

of simulations to use during selection (default 10), and the second sets the number to use in the estimation of the error rates (default 50).

The SEED option provides the seed for the random numbers used for the randomizations during in the simulations. The default value of 0 continues an existing sequence of random numbers, if none have been used in the current Genstat job, it initializes the seed automatically using the computer clock.

The PLOT option provides for group means, labels for group means, unit scores, group polygons enclosing units, and 95% confidence circles around group means. The YROOT and XROOT options specify the roots for the axes. The TITLE, WINDOW and SCREEN options allow further control of the plots. More than one plot can be output by having a list of scalars for YROOT. In this case, the values of XROOT, TITLE, WINDOW and SCREEN are cycled in parallel. A rug-like plot is drawn if only one root is extracted or if YROOT is set to a missing value.

Results from the analysis can be saved using the parameters NEWGROUPS, ALLOCATION, MEANS, SCORES, DISTANCES, LRV, ADJUSTMENTS, GDISTANCES, CCORRELATIONS, ICORRELATIONS and CORRELATIONS. The structures specified for these parameters need not be declared in advance. The default is to save MEANS and SCORES in matrices. However, if you declare either as a pointer, it will instead store the results as a data matrix (i.e. a pointer of variates corresponding to the columns of the matrix). The results correspond to p dimensions, where p is the smaller of either the number of variates, or the number of groups minus one.

Options: PRINT, NROOTS, REALLOCATE PLOT, VALIDATIONMETHOD, NSIMULATIONS, NCROSSVALIDATIONGROUPS, SEED, YROOT, XROOT, TITLE, WINDOW, SCREEN.
Parameters: DATA, GROUPS, NEWGROUPS, ALLOCATION, MEANS, SCORES, DISTANCES, LRV, ADJUSTMENTS, GDISTANCES, CCORRELATIONS, ICORRELATIONS, CORRELATIONS.

Method

A canonical variates analysis (CVA) is used to obtain the scores for the group means and the LRV containing the loadings (L), roots and trace; the analysis excludes units omitted by RESTRICT, or that have missing values in the data variates or the GROUPS factor. Scores are then calculated for all the units (i.e. ignoring any restrictions or missing values), using the formula

$$(XL) - (JA)$$

where X is a matrix containing the full set of units-by-variables data, J is a column vector of one's, and A is a row vector of adjustments required to place the scores for the units onto the same scale as those for the group means.

Mahalanobis squared distances between the units and the group means are calculated from the canonical variate scores. Each unit is then allocated to the group for which it has the smallest Mahalanobis squared distance to the group mean.

There are two internal procedures _DISAXSCALE and _DISENCLOSE.

Action with RESTRICT

The input variates and factor may be restricted. The restrictions must be identical. The canonical variates analysis is based only on the units not excluded by the restriction and having non-missing values for all data variates. Scores are calculated for all the units with a complete set of non-missing values, however these are based only on the non-excluded units: i.e. the adjustments for the canonical variate scores are calculated from the non-excluded units, and the loadings used to calculate the scores are those from the canonical variates analysis. If there is a restriction in place, the count setting of the PRINT option will produce two parallel tables, one with the number of units in the training set and another with the number of units if the data were not restricted. The table setting of the PRINT option will produce two tables, one using only those units present in the training set and another for those units excluded by the restriction.

If the restriction results in levels of the GROUPS factor being unrepresented in the training set,

the group centroids for these levels are estimated from the scores of the units that were excluded and the levels will be included in the GDISTANCE symmetric matrix. The DISTANCES parameter will include the distances to all the centroids, including those levels not in the training set. The ALLOCATION parameter will allocate to the nearest centroid even if it was not in the training set (as distinct from the NEWGROUPS factor).

For levels and units in the training set, plotted means are marked with symbol 1 (×) and the units with symbol 3 (+). Means for levels and units excluded by the restriction are plotted with symbols 19 and 20 respectively. Units with a missing GROUPS value are plotted with symbol 18 if not in the excluded set otherwise symbol 21 is used. Polygons are not drawn around groups excluded from the training set by a restriction.

References

Klecka, W.R. (1980). *Discriminant Analysis (Quantitative Applications in the Social Sciences)*. Sage Publishing, Newbury Park, California.

Mardia, K.V., Kent, J.T. & Bibby, J.M. (1979). *Multivariate Analysis*. Academic Press, London.

See also

Directive: CVA.

Procedures: CVAPLOT, DBIPLOT, QDISCRIMINATE, SDISCRIMINATE.

Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

DKALMAN

Plots vector time series (A.I. Glaser).

Options

TIMEPOINTS = *variate*	X-coordinates for the graphs; default uses the integers 1, 2...
TITLE = *texts*	Overall title for the graphs
YTITLE = *texts*	Titles for the y-axes; default * forms titles automatically from the identifiers or labels of the y-variables
XTITLE = *texts*	Title for the x-axis in each set of graphs; default * uses the identifier of TIMEPOINTS (if set)
NROWS = *scalar*	Specifies the number of rows of graphs to appear on the graphics screen; default * takes the number of y-variables
NCOLUMNS = *scalar*	Specifies the number of columns of graphs to appear on the graphics screen; default 1

Parameter

SAVE = *pointers*	Save structure from KALMAN with information about the analysis; default plots information from the most recent KALMAN analysis

Description

DKALMAN plots results from an analysis by the KALMAN procedure. By default this will be from the most recent analysis, but you can use the SAVE parameter to supply results from an earlier analysis (saved using the SAVE parameter of KALMAN).

The TIMEPOINTS option supplies the time points. If this is not set (or if there are at most only two unique values), DKALMAN uses the integers 1 ... *n*, where *n* is the number of time points in the analysis.

You can use the TITLE option to supply a title for the graphs. If TITLE is not set, no title is displayed.

The YTITLE option supplies a title for the y-axes; this must be set either to a text with a value for each y-variable, or one with a single value (which will then be used for all of them). You can set YTITLE='' to stop a title appearing on the y-axes. If YTITLE is not set, DKALMAN forms the titles automatically. If the Y parameter of KALMAN was set to a matrix, DKALMAN uses the column labels if available, or otherwise the column numbers. If Y was set to a pointer of variates, DKALMAN uses the identifiers of the variates if these exist outside the pointer. For example if the pointer contains variates Loss and Profit, then those identifiers will be used. If the variates have no identifiers of their own, but exist only as suffixed identifiers (e.g. Income[2010] and Income[2011] or Income['Dollar'] and Income['Euro']), then it uses the pointer suffixes (e.g. 2010 and 2011) or, if available, the labels (e.g. 'Dollar' and 'Euro').

The XTITLE option supplies a title for the x-axes; this must be set to a text with a single value. If XTITLE is not set, DKALMAN uses the identifier of the TIMEPOINTS option (if specified).

By default, the graphs are plotted in a single column, but this can be altered by using NROWS and NCOLUMNS options to specify the required number of rows and columns respectively. The graphs will be spread over several screens if the values supplied for NROWS and NCOLUMNS, are too small to include all the graphs on a single screen.

Options: TIMEPOINTS, TITLE, YTITLE, XTITLE, NROWS, NCOLUMNS.
Parameter: SAVE.

Action with `RESTRICT`
`DATA` variates must not be restricted.

See also
Procedure: `KALMAN`.
Genstat Reference Manual 1 Summary sections on: Time series, Graphics.

DKEY

Adds a key to a graph (D.B. Baird & V.M. Cave).

Options

`WINDOW` = *scalar*	Window in which to draw the key; default 2
`NCOLUMNS` = *scalar*	Number of columns forming the grid in which the key is displayed; default * (i.e. set automatically)
`NROWS` = *scalar*	Number of rows forming the grid in which the key is displayed; default * (i.e. set automatically)
`TITLE` = *text*	Title for the key
`PENTITLE` = *scalar*	Pen used to write the title of the key; default is that set for the window in which the key is plotted
`PENLABELS` = *variate*	Pens to use to plot the labels; default is to plot the labels using the settings of `LFONT`, `LSIZE` and `LCOLOUR`
`TPOSITION` = *string*	Position of the title (`inside`, `outside`, `left`, `centre`, `center`, `right`); default `cent`, `outs`
`ORDER` = *string*	Order in which to fill the key's row by column grid (`rows`, `columns`); default `rows`
`LSIZE` = *scalar*	Relative size of the labels; default 1
`LFONT` = *scalar* or *text*	Font to use for the labels; default 1
`LCOLOUR` = *scalar* or *text*	Colour used to write the labels; default `'black'`
`XLOFFSET` = *scalar* or *variate*	Offset in the x-direction between the items (i.e. symbols/lines) and labels in the key; default 0
`COLSPACING` = *string*	Column spacing (`equal`, `unequal`); default `equa`
`ROWGAP` = *scalar*	Multiplier for gaps between rows; default 1
`COLGAP` = *scalar*	Multiplier for gaps between columns; default 1
`BORDER` = *string*	Border around the key (`fit`, `given`, `none`); default `fit`
`CBORDER` = *string*	Colour for the border around the key; default `'black'`

Parameters

`DESCRIPTIONS` = *texts*	Labels for the key
`PEN` = *variates*	Pens to use for the items in the key; default uses the integers 1, 2 ...
`METHOD` = *texts*	Method for plotting the items in the key (`fill`, `point`, `line`, `both`, `none`); default is to use the method defined for the corresponding `PEN`
`SYMBOL` = *variates*, *scalars*, *factors* or *texts*	Symbols to be drawn in the key; default is to use those specified by `PEN`
`COLOUR` = *variates*, *scalars*, *factors* or *texts*	Colours of lines, or of filled areas when `METHOD='fill'`; default is to use those specified by `PEN`
`CSYMBOL` = *variates*, *scalars*, *factors* or *texts*	Colours of symbols; default is to use those specified by `PEN`
`CFILL` = *variates*, *scalars*, *factors* or *texts*	Colours used to fill hollow symbols; default is to use those specified by `PEN`
`SIZEMULTIPLIER` = *variates*, *scalars* or *factors*	Relative sizes of symbols and filled area; default is to

use those specified by PEN

LINESTYLE = *variates*, *scalars* or *factors*
: Numbers or names of the linestyles to use; default is to use those specified by PEN

THICKNESS = *variates*, *scalars* or *factors*
: Thicknesses of the lines; default is to use those specified by PEN

TRANSPARENCY = *variates*, *scalars* or *factors*
: Transparencies of the filled areas when METHOD='fill'; default is to use those specified by PEN

Description

The DKEY procedure provides a more flexible way of providing a key for a plot, than the standard facilities provided by the ordinary plotting commands. The standard keys can be suppressed by setting the option KEYWINDOW in those commands to zero.

The labels to appear in the key must be supplied as a text structure by the DESCRIPTIONS parameter. The number of labels defines the number of items *n* to appear in the key. The appearance of the labels (size, font and colour) can be controlled either by the PENLABELS option by or the LSIZE, LFONT and LCOLOUR options. PENLABELS can supply a variate, with *n* values, to define the pens to use for the labels.

If PENLABELS is not set, the labels are all written in the same style, using the settings of the LSIZE, LFONT and LCOLOUR options. The LSIZE option modifies the size of the labels, by specifying a value by which the default size is to be multiplied; default 1. The LFONT option specifies the font to use for the labels. This can be set either to a text containing the name of a font family, or to a scalar containing an integer between 1 and 25. The default is to use the *default graphics font* (i.e. the default font set on the Fonts tab of the Options menu in the Graphics Viewer). The LCOLOUR option specifies the colour for the labels. This can be set either to a text containing the name of one of Genstat's pre-defined colours, or to a scalar containing a number defining a colour using the RGB system. The default is 'black'.

The METHOD parameter supplies a text defining the types of item to be plotted in the key. The text can contain a single string if all the items are to be displayed in the same way, or a string for each item if they are to be displayed differently. The possible strings are

'point'	for points,
'line'	for lines,
'both'	for points and lines,
'fill'	for filled rectangles, and
'none'	to prevent an item from being plotted.

The default is to use the method defined for the corresponding PEN.

The appearance of the items (symbol type, colour, size, linestyle, line thickness and transparency) can be controlled by specifying the pens to be used to plot them by the PEN parameter. The default is to use pens 1 ... *n*.

Alternatively, you can set the appearance of the items explicitly, by using the parameters SYMBOL, COLOUR, CSYMBOL, CFILL, SIZEMULTIPLIER, LINESTYLE, THICKNESS and TRANSPARENCY. (These override the settings from PEN.) For each of these parameters, you can supply either a single value or a structure with *n* values (one for each item).

The SYMBOL parameter defines the symbols for items that are displayed as points (or as both points and lines). It can be set either to a text containing names of Genstat's pre-defined symbols (see PEN for details), or to a scalar or variate containing integers between −4 and 22, or to factor with at most 22 levels.

The COLOUR, CSYMBOL and CFILL parameters specify the colours to be used for the items.

The COLOUR parameter defines the colours of lines and filled areas. The CSYMBOL parameter defines the colours used for symbols. The CFILL parameter defines the colours used for filling areas inside hollow symbols. They can be set either to a text containing the name of one of Genstat's pre-defined colours (see PEN for details), or to a scalar or variate containing numbers defining colours using the RGB system, or to a factor. The transparency of a filled area can be set using the TRANSPARENCY parameter. This can be set either to a scalar or variate containing values between 0 (opaque) and 255 (completely transparent), or to factor with at most 255 levels.

The SIZEMULTIPLIER parameter can modify the size of symbols and filled areas, by specifying a value by which the default size is to be multiplied. Either a scalar, variate or factor can be supplied. The LINESTYLE parameter defines what sort of line is drawn (for example a solid, dotted or dashed line). This can be set either to a text containing the names of Genstat's pre-defined linestyles (see PEN for details), or to a scalar or variate containing integers between 1 and 11, or to factor with at most 11 levels. The THICKNESS parameter can modify the thickness of lines, by specifying a value by which the standard thickness is to be multiplied. Either a scalar, variate or factor may be supplied.

The WINDOW, NCOLUMNS, NROWS, ORDER, XLOFFSET, COLSPACING, ROWGAP, COLGAP, BORDER and CBORDER options control the layout of the key. The WINDOW option specifies the window in which the key is drawn; default 2. The number of rows and columns, forming the grid in which the key is arranged, can be set by the NROWS and NCOLUMNS options, respectively. If these are not set, an appropriate grid is constructed automatically. The order in which the items fill the grid is determined by the ORDER option. The default, ORDER=rows, fills the grid row by row. Alternatively ORDER=columns fills the grid column by column. The COLSPACING option specifies whether or not the columns of the grid are equally spaced (equal and unequal, respectively); default equal. The ROWGAP and COLGAP options control the sizes of the gaps between rows and columns, respectively. The distance between the items and labels can be adjusted by the XLOFFSET option. Each label in the grid can be individually offset by suppling a variate with *n* values. When a single value is supplied, a common offset is applied to all labels in the grid. The BORDER option controls the border drawn around the key. The default, BORDER=fit, draws a border fitted to the key. When BORDER=given, the border frames the window (and the key is drawn so that it occupies the entire window). Finally, if BORDER=none, no border is drawn. The CBORDER option specifies the colour for the border, when one is drawn around the key; default 'black'.

The TITLE option can provide a title for the key. The pen for the title can be set by the PENTITLE option. The default is to use the pen defined for the window in which the key is plotted. The TPOSITION parameter specifies the position of the title: either inside or outside the border with left, right or centre justification. The default is to centre the title outside the border of the key.

Options: WINDOW, NCOLUMNS, NROWS, TITLE, PENTITLE, PENLABELS, TPOSITION, ORDER, LSIZE, LFONT, LCOLOUR, XLOFFSET, COLSPACING, ROWGAP, COLGAP, BORDER, CBORDER.
Parameters: DESCRIPTIONS, PEN, METHOD, SYMBOL, COLOUR, CSYMBOL, CFILL, SIZEMULTIPLIER, LINESTYLE, THICKNESS, TRANSPARENCY.

Action with RESTRICT

DKEY takes account of restrictions on DESCRIPTIONS, PEN, PENLABELS and XLOFFSET. However, the parameters METHOD, SYMBOL, COLOUR, CSYMBOL, CFILL, SIZEMULTIPLIER, LINESTYLE, THICKNESS and TRANSPARENCY must not be restricted.

See also

Directive: PEN

Procedures: DTEXT, DFRTEXT.
Commands for: Graphics.

DKSTPLOT

Produces diagnostic plots for space-time clustering (D.A. Murray).

Options

`PLOT` = *string token*	Whether to produce plots separately or in composite (`separate`, `combined`); default `comb`
`DZERO` = *string token*	Whether to produce a DZERO plot (`yes`, `no`); default `no`

Parameters

`Y` = *variates*	Vertical coordinates of the spatial point patterns
`X` = *variates*	Horizontal coordinates of the spatial point patterns
`KS` = *variates*	Estimates of spatial K function
`KT` = *variates*	Estimates of temporal K function
`KST` = *matrices*	Estimates of space-time K function
`KSE` = *matrices*	Estimates of standard errors of space-time K function

Description

For data that consist of locations and times of events within a specified spatial region and time-period, it is often of interest to examine whether events that are relatively close in space are also relatively close in time. Data that have events both close in space and time are said to exhibit space-time clustering. `DKSTPLOT` produces three diagnostic plots for space-time clustering. The first plot is a map of the spatial point pattern. The second is a contour or perspective plot of the difference between the space-time K function and product of the spatial and temporal K functions.

$$D^{(s,t)} = K^{st(s,t)} - K^{s(s)} \times K^{t(t)}$$

This gives information on the scale and nature of the dependence between spatial and temporal components. Alternatively, by setting the option `DZERO=yes` the contour plot will be drawn by scaling $D^{(s,t)}$ by the product of the spatial and temporal K functions.

$$D0^{(s,t)} = D^{(s,t)} / K^{s(s)} \times K^{t(t)}$$

This represents the proportional increase attributable to space-time interaction. The third plot is of the standardized residuals given by

$$(K^{st(s,t)} - K^{s(s)} \times K^{t(t)}) / SE(K^{st(s,t)})$$

The data required by the procedure are the coordinates of a spatial point pattern (specified by the parameters `X` and `Y`). The estimates for the spatial and temporal K functions are supplied using the `KS` and `KT` parameters. The space-time K function estimates and associated standard errors are supplied using the `KST` and `KSE` parameters.

The `PLOT` option controls whether to display the plots on one graph or to produce a separate graph for each plot.

Options: `PLOT`, `DZERO`.
Parameters: `Y`, `X`, `KS`, `KT`, `KST`, `KSE`.

Method

The procedure `DPTMAP` is called draw the map of the spatial point process. The estimates for the K functions and associated standard errors are calculated using the procedures `KSTHAT` and `KSTSE`.

Action with RESTRICT

The variates `X` and `Y` may be restricted. The K function estimates cannot be restricted.

References

Diggle, P.J., Chetwynd, A.G., Haggkvist, R. & Morris, S.E. (1995). Second-order analysis of space-time clustering. *Statistical Methods in Medical Research*, **4**, 124-136.

See also

Procedures: `KSTHAT`, `KSTMCTEST`, `KSTSE`.

Genstat Reference Manual 1 Summary sections on: Graphics, Spatial statistics.

DMADENSITY

Plots the empirical CDF or PDF (kernel smoothed) by groups (D.B. Baird).

Options

PLOT = *string tokens*	What to plot (cdf, pdf, histogram); default cdf, pdf
TRANSFORMATION = *string token*	Whether to transform the data to log base 2 (log2, none); default none
BANDWIDTH = *scalar*	Bandwidth to use in kernel density estimates for PDF
ARRANGEMENT = *string token*	Whether to use trellis or single plots (single, trellis); default trel
WINDOW = *scalar*	Window number for the graphs; default 3
KEYWINDOW = *scalar*	Window number for the key; default 0 i.e. none
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameters

DATA = *variates* or *pointers*	Data coordinates
GROUPS = *factors* or *texts*	Groups

Description

DMADENSITY plot the empirical cumulative probability density function (CDF) or probability density function (PDF) as estimated by kernel smoothing. The bandwidth for the smoothing must be specified by the BANDWIDTH option. By default the data values are untransformed, but you can set option TRANSFORMATION=log2 to transform them to logarithms base 2.

The data are specified by the DATA parameter. The data can be in a single variate. The GROUPS parameter can then supply a factor defining groups, within which the CDF and PDF are to be displayed. Alternatively, the data can be in a pointer to a set of variates. The GROUPS can be omitted, or it can supply a text defining a label for each group. By default, the plots for the groups are displayed in a trellis arrangement, but you can set option ARRANGEMENT=single to display them separately, in single plots.

The WINDOW option specifies the window to use (by default 3), and the KEYWINDOW option can specify a window for a key (by default there is none). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

Options: PLOT, TRANSFORMATION, BANDWIDTH, ARRANGEMENT, WINDOW, KEYWINDOW, DEVICE, GRAPHICSFILE.
Parameters: DATA, GROUPS.

Action with RESTRICT

The DATA variate(s) can be restricted to use just a subset of the data values.

See also

Procedures: DPROBABILITY, KERNELDENSITY, PTKERNEL2D, PTK3D.
Genstat Reference Manual 1 Summary sections on: Graphics, Microarray data.

DMASS

Plots discrete data like mass spectra, discrete probability functions (J.W. McNicol).

Options

X = *variate*	Positions on the x-axis at which to plot the lines; default uses 1, 2 ...
TITLE = *text*	Title for the graph; default * i.e. none
WINDOW = *scalar*	Window for the graph; default 3
YTITLE = *texts*	Title for the y-axis
XTITLE = *texts*	Title for the x-axis
YMARKS = *scalars* or *variates*	Distance between each tick mark on y-axis (scalar) or positions of the marks (variate)
XMARKS = *scalars* or *variates*	Distance between each tick mark on x-axis (scalar) or positions of the marks (variate)
YMPOSITION = *string tokens*	Position of the tick marks across the y-axis (left, right, centre); default left
XMPOSITION = *string tokens*	Position of the tick marks across the x-axis (above, below, centre); default * i.e. none
YLABELS = *texts*	Labels at each mark on y-axis
XLABELS = *texts*	Labels at each mark on x-axis
PENAXES = *scalar*	Pen to be used for axes and their titles; default 1
PENTITLE = *scalar*	Pen to use for the title; default 1
LINETHICKNESS = *scalar*	Thickness for the vertical lines representing the mass heights; default 1
SCREEN = *string token*	Whether to clear screen before displaying the graph (keep, clear); default clea

Parameters

Y = *variates*	Heights for the masses
LINECOLOUR = *texts* or *scalars*	Colours for the vertical lines representing mass heights; default * sets suitable colours automatically

Description

DMASS produces plots appropriate for ordered discrete data such as mass spectra, discrete probability functions or principal component weights. The Y parameter specifies one or more variates, each of which defines the heights of a set of vertical lines. By default the lines are plotted at equal unit spacing along the x-axis (i.e. at positions 1, 2 and so on), but other positions can be specified using the X option. The X and Y variates must all have equal lengths. So, for example, if a particular line is absent for one of the spectra, the variate must contain a zero value. The LINECOLOUR parameter defines the colour to be used for each set. By default, the standard colours are used in the same order as for pens 2, 3... (see PEN).

As usual, options TITLE, WINDOW and SCREEN allow you to define a title for the plot, specify which window to use, and indicate whether or not to clear the screen beforehand. Likewise, options YTITLE, XTITLE, YMARKS, XMARKS, YMPOSITION, XMPOSITION, YLABELS and XLABELS define titles, tick marks and labelling of the axes, similarly to the XAXIS and YAXIS directives. The pens to use for the title and for the axes can be defined by the PENTITLE and PENAXES options, and the LINETHICKNESS option controls the thickness of the lines used to plot the masses.

Options: X, TITLE, WINDOW, YTITLE, XTITLE, YMARKS, XMARKS, YMPOSITION, XMPOSITION, YLABELS, XLABELS, PENTITLE, PENAXES, LINETHICKNESS, SCREEN.
Parameters: Y, LINECOLOUR.

Method
For the ith mass, a pair of variates of length 2 is created; y=[mass,0] and x=[i,i]. These are plotted by the DGRAPH directive with METHOD=line.

Action with RESTRICT
DMASS takes account of restrictions on X or any of the Y variates.

See also
Procedures: BARCHART, DPROBABILITY, RUGPLOT.
Genstat Reference Manual 1 Summary section on: Graphics.

DMSCATTER

Produces a scatter-plot matrix for one or two sets of variables (J. Ollerton & R.W. Payne).

Options

PLOT = *string tokens*	Additional information to include in the scatter plots (correlation, histograms, boxplots, densities, dothistograms); default *
SCALING = *string token*	How to scale the x- and y-axes (common, equal, none); default none
PEN = *scalar* or *variate* or *factor*	Pens to plot the scatter plots; default 1
PENHISTOGRAM = *scalar*	Pens to plot the histograms; if PEN is a factor the default plots the histogram for each group separately using the pen used for that group in the scatter plots, otherwise the default is to use pen 2
PENCORRELATION = *scalar*	Pen to use to write the correlations; default 1
PENTITLE = *scalar*	Pen to use to write the axis titles; default uses the pens currently defined for the axes in the windows that are used for the plots
PENAXIS = *scalar*	Pen to use to draw the axes; default uses the currently defined pens
PENLABELS = *scalar*	Pen to use to write the axis labels; default uses the currently defined pens
NROWS = *scalar*	Number of rows of graphs to put in a single frame (i.e. page); default puts them all in one frame
NCOLUMNS = *scalar*	Number of columns of graphs to put in a single frame; default uses the same value as NROWS
ASPECTRATIO = *scalar*	Ratio of the length of the y-axis to the length of the x-axis in each graph
FRAMESHAPE = *string token*	Shape of the plotting frame (landscape, portrait, square); default squa
MARGINSIZE = *scalar*	Specifies the size of the margins at the bottom and left-hand edge of the frame

Parameters

Y = *pointers*	Each pointer contains a set of variates and/or factors to be plotted
X = *pointers*	Each pointer contains a set of variates and/or factors to be plotted as the x-variables in a rectangular scatter-plot matrix; if unset Y specifies both the x-variables and y-variables for a symmetric scatter-plot matrix
TITLE = *texts*	Overall title for the plot
YTITLES = *texts*	Labels for the axes for the Y variates and factors, to use instead of their identifiers
XTITLES = *texts*	Labels for the axes for the X variates and factors, to use instead of their identifiers
YMARKS = *variates*, *scalars* or *pointers*	Marks to use on the axes for the Y variates and factors, if any of these contains missing values, the marks and their labels are suppressed for that variate or factor
XMARKS = *variates*, *scalars* or *pointers*	Marks to use on the axes for the X variates and factors, if

any of these contains missing values, the marks and their labels are suppressed for that variate or factor

Description

Procedure DMSCATTER produces two types of scatter-plot matrix, using high-resolution graphics. For a symmetric scatter-plot matrix, the variates and/or factors to be plotted against each other must be specified, in a pointer, by the Y parameter. The scatter-plot contains a lower-triangular array of graphs, one for each pair of variables. Alternatively, for a rectangular scatter-plot matrix, there are two set of the variates and/or factors. The set that defines the y-values for the graphs are specified (in a pointer as before) by the Y parameter, and those that define the x-values for the graphs are specified (also in a pointer) by the X parameter. The scatter-plot now contains a rectangular array of graphs, one for each pair of x- and y-variables.

By default the identifiers of the relevant x- and y-variables are used for the titles of the axes at the lower and left-hand edges of the graphics frame (i.e. page). Alternatively, you define your own titles for the y-variables by setting the YTITLES to a text with a value for each Y variate or factor. Similarly, you can use the XTITLES parameter to supply your own titles for the X variates or factors. You can also use the TITLE parameter to supply an overall title.

The YMARKS parameter allows you to specify your own marks for the axes corresponding to the y-variables. (These are then used as the settings of the MARKS parameter of the YAXIS and XAXIS directives.) You can set YMARKS to single variate or scalar, if you want to use the same marks for every y-variable. Alternatively, you can set it to a pointer with a variate or factor for each Y variate or factor, if you want to specify different marks. If any of the variates or scalars contains missing values, the marks and their labels are suppressed on the corresponding axes. You can use the XMARKS parameter similarly, to specify axis marks for the x-variables.

The PEN option specifies the pens to be used to plot the graphs. The setting can be a scalar to plot all the points with the same pen, or a variate or a factor to use different pens. If PEN is set to a factor, a key is included in the plot to identify the correspondence between the pens and the groups. The default is to use pen 1.

The PLOT option allows you to specify extra information to be included in the plot, with settings:

correlation	prints the correlation of the pair of variables in each plot, at the top of the plot;
histograms	plots histograms of the variables down the diagonal of a symmetric scatter-plot matrix, or along the top and down the right-hand side of a rectangular scatter-plot matrix;
boxplots	displays boxplots of the variables down the diagonal of a symmetric scatter-plot matrix, or along the top and down the right-hand side of a rectangular scatter-plot matrix;
densities	displays one-dimensional density plots (or violin plots) of the variables down the diagonal of a symmetric scatter-plot matrix, or along the top and down the right-hand side of a rectangular scatter-plot matrix; and
dothistograms	plots dot histograms of the variables down the diagonal of a symmetric scatter-plot matrix, or along the top and down the right-hand side of a rectangular scatter-plot matrix.

Note, only one of the settings histograms, boxplots, densities, dothistograms is allowed; if more than one is set, the first item the list above is used.

The PENHISTOGRAM option specifies the pens to plot the histograms. If PEN is a set to a factor, the default for PENHISTOGRAM plots histogram for each group, using the pen used for that group in the scatter plots. Otherwise the default is to use pen 2. The PENCORRELATION option specifies the pen to use to print the correlations; default 1.

The PENTITLE, PENAXIS and PENLABELS options define the pens to use for the titles of the x- and y-axes, for the axes themselves, and for their labels. If any of these is unset, the default is to use the pens already defined for that aspect of the axes in the windows used in the plot.

The SCALING option controls the scaling of the x- and y-axes, the settings:

equal	uses equal scaling for the x- and y-axes in each graph,
common	used exactly the same axes (upper and lower limits as well as scaling) for the axes in all the graphs,
none	defines all the axes independently (the default).

By default the plots are square, but you can request rectangular plots by setting the ASPECTRATIO option to the required value for the length of the y-axis divided by the length of the x-axis.

The MARGINSIZE option specifies the size of the margins at the bottom and left-hand edge of the graphics frame. If this is unset, the margins are defined automatically, using a smaller value if all the axis marks and labels on an edge have been suppressed.

The FRAMESHAPE option specifies the shape of the graphics frame, with settings:

landscape	for a frame of size 1.4 × 1.0 i.e. wider in the x- than the y-direction,
portrait	for a frame of size 1.0 × 1.4 i.e. wider in the y- than the x-direction,
square	for a frame of size 1.0 × 1.0.

Some graphics devices do not support the use of device cordinates greater than 1.0, so the default is FRAMESHAPE=square. (See FRAME and DEVICE for more information.)

By default the graphs are all plotted in a single frame (i.e. page), but you can specify the NROWS and NCOLUMNS options to split them across several frames. NROWS specifies the number of rows of plots to put in a single frame. The default is to fit them all into one frame. NCOLUMNS specifies the number of columns of plots to put in one frame. The default is to use the same value as NROWS.

Options: PLOT, SCALING, PEN, PENHISTOGRAM, PENCORRELATION, PENTITLE, PENAXIS, PENLABELS, NROWS, NCOLUMNS, ASPECTRATIO, FRAMESHAPE, MARGINSIZE.
Parameters: Y, X, TITLE, YTITLES, XTITLES, YMARKS, XMARKS.

Action with RESTRICT
If any of the variates or factors is restricted, only the units not excluded by the restriction will be plotted.

See also
Directive: DGRAPH.
Procedures: DSCATTER, TRELLIS, BOXPLOT, DXDENSITY, DOTHISTOGRAM.
Genstat Reference Manual 1 Summary section on: Graphics.

DMST

Gives a high resolution plot of an ordination with minimum spanning tree (A.W.A. Murray).

Options

DIMENSIONS = *scalars*	Two numbers specifying the dimensions to display on the y- and x-axes; default 2,1
TITLE = *text*	Title for the graph
WINDOW = *scalar*	Window for the graph; default 1
KEYWINDOW = *scalar*	Window for the key; default 2
SCREEN = *string token*	Controls screen (clear, keep); default clea

Parameters

COORDINATES = *matrices* or *datamatrices*	Coordinates from ordination
TREE = *matrices*	Minimum spanning tree
SIMILARITY = *symmetric matrices*	Association matrix used to derive ordination
SYMBOLS = *factors* or *texts*	Symbols to label the coordinates
PENCOORDINATES = *scalars*	Pen to use for the coordinates
PENTREE = *scalars*	Pen to use for the minimum spanning tree

Description

DMST plots a minimum spanning tree using coordinates saved, for example, from a PCO. The COORDINATES parameter specifies the coordinates for the units in the plot, using either a matrix or a pointer to a set of variates (that is, a "datamatrix"). The minimum spanning tree can be supplied using the TREE parameter, or it can be calculated from the original association matrix specified using the SIMILARITY parameter. If TREE supplies a matrix with no values, these will be set to the tree calculated from the SIMILARITY matrix. If the COORDINATES structure was originally declared with row labels the procedure will automatically use these to label the plots. Alternative symbols can be defined using the SYMBOLS parameter. You can also specify the pens to be used to plot the coordinates and tree, using parameters PENCOORDINATES and PENTREE respectively. The definition of these pens, outside the procedure, thus allows the colour, size, font and linestyle of links in the tree to be controlled. By default the coordinates are plotted with colour black and the tree with colour red, symbols are 0.8 of normal size, and the tree is plotted with a dotted line.

Options TITLE, WINDOW, KEYWINDOW and SCREEN function as usual for high resolution graphics. If the WINDOW is unset a default layout with appropriately labelled axes is produced in window 1. Axes will be scaled automatically unless limits have already been set outside the procedure.

Options: DIMENSIONS, TITLE, WINDOW, KEYWINDOW, SCREEN.
Parameters: COORDINATES, TREE, SIMILARITY, SYMBOLS, PENCOORDINATES, PENTREE.

Method

A two dimensional representation of the results of a multivariate analysis, such as a PCO, is plotted on the current high resolution graphics device. A minimum spanning tree is calculated (by HDISPLAY) from an input similarity matrix if not supplied. The tree is superimposed on the plot. The procedure uses GETATTRIBUTE to access the row labels (if any) of the input structures. The input structures are converted to variates if necessary and DGRAPH is used to plot the desired data.

Action with `RESTRICT`
Restrict is irrelevant with matrix input structures. It should work as expected with variates.

See also
Directives: `HDISPLAY`, `PCO`, `PCP`.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

DOTHISTOGRAM

Plots dot histograms (L.H. Schmitt & A.R.G. McLachlan).

Options

`TITLE` = *text*	Title for the plot; default `*` i.e. none
`AXISTITLE` = *text*	Title for the axis representing the data values; default `*` uses the name of the `DATA` variate if there is only one, otherwise no title
`WINDOW` = *scalar*	Window for the plot; default `*` uses window 1 when `PEN` is set, and window 3 when `PEN` is unset
`ORIENTATION` = *string token*	Direction of the plot (`horizontal`, `vertical`); default `vert`
`YORIENTATION` = *string token*	Direction of the y-axis for horizontal plots (`reverse`, `normal`); default `reve`
`SCREEN` = *string token*	Whether to clear screen before displaying chart (`keep`, `clear`); default `clea`
`JUSTIFICATION` = *string token*	How to position the dots; (`right`, `left`, `centre`, `center`, `bottom`, `top`, `backtoback`); default `cent`
`CREATEMISSINGLEVEL` = *text*	Whether to create a level for missing `GROUPS` data (`yes`, `no`); default `no`
`OMITEMPTYLEVELS` = *text*	Whether to omit levels of `GROUPS` for which there are no `DATA` values to plot (`yes`, `no`); default `no`
`SIZE` = *scalar*	Size of the pen used to plot the dots; default 1
`KEYWINDOW` = *scalar*	Window to use for a key when `PEN` is set; default 2
`KEYDESCRIPTION` = *text*	Overall title for a key when `PEN` is set; default `*` uses name of `PEN` data structure
`SELECTION` = *string tokens*	Selects the statistics to be plotted (`mean`, `median`, `interquartilerange`); default `*` i.e. none
`BARWIDTH` = *scalar*	Width of the bars for the selected statistics; default `*` sets an appropriate width automatically
`BARTHICKNESS` = *scalar*	Thickness of the bars for the selected statistics; default 2
`CMEAN` = *scalar*, *variate* or *text*	Colour of the bars for the means
`CMEDIAN` = *scalar*, *variate* or *text*	Colour of the bars for the medians
`CINTERQUARTILE` = *scalar*, *variate* or *text*	Colour of the bars for the inter-quartile ranges

Parameters

`DATA` = *variates* or *pointers*	Data to be plotted
`GROUPS` = *factors*	Factor to divide values of a `DATA` variate into groups
`COLOURS` = *scalars*, *variates*, *texts* or *factors*	Colours for the histograms in each plot, a scalar to use the same colour for all the histograms, or a variate or factor to plot each histogram in a different colour; default `'black'`
`NOBSERVATIONS` = *tables*	Save tables of count
`PENS` = *variates*, *factors* or *pointers*	Pens to define colours for the individual dots; default uses those defined by the `COLOURS` parameter
`SYMBOLS` = *scalars*, *variates*, *texts* or *factors*	Symbols for the points
`DESCRIPTION` = *texts*	Annotation for key when `PEN` is set; default uses unique values of `PEN`

Description

Dot histograms display each observation in a set of values as a dot. The observations are allocated to bins of equal range, and all the observations in a bin are plotted in a row. The values are supplied by the DATA parameter. If this is a variate and the GROUPS parameter is set, then dot histograms are plotted side-by-side, one for each level of the grouping factor. If DATA is set to a pointer of variates, a dot histogram is plotted for each variate, side-by-side, and the GROUPS parameter is ignored.

You can control the colours of the dots, using either the COLOURS or the PENS parameter. The COLOURS parameter defines the colour of a whole histogram. You can specify a scalar or single-valued text to define the colour to be used for all the dot histograms in the plot, or you can supply a variate, text or factor, with a different value for each dot histogram in the plot. The PENS parameter specifies the colours by specifying a pen for every dot in the display, and over rides the COLOURS parameter if both are specified. (Each colour is taken from that defined for the pen concerned; other pen settings like the symbol are ignored.) So, PENS must be set to a variate or a factor if DATA is set to a variate. Alternatively, if DATA is set to a pointer, PENS too must be set to pointer, containing either a set of variates, or a set of factors (one for each variate in the DATA pointer).

The SYMBOLS parameter can specify the symbols for a whole histogram, similarly to COLOURS. Like COLOURS, this can supply a scalar, a variate, a text or a factor. You can specify a scalar or single-valued text to define the symbol to be used for all the dot histograms in the plot, or you can supply a variate, text or factor with a different value for each dot histogram in the plot.

The NOBSERVATIONS parameter can save a two-way table of the number of dots in each row for each dot histogram in the plot.

By default, the dot histograms are plotted vertically with a row for each bin arranged along the y-axis, but you can set option ORIENTATION=horizontal to plot the histograms horizontally and the dots are then in columns. When ORIENTATION=horizontal, the horizontal axis is taken to be the y-axis, so the same XAXIS and YAXIS settings can be used however the histograms are oriented. Rows of dots in each dot histogram are usually centred on the x-axis, but the JUSTIFICATION option gives control over this. They can be either centred (the default), right justified, left justified, top justified, bottom justified, or back-to-back to plot adjacent dot histograms back to back. The left and right settings and the top and bottom settings are provided for vertical and horizontal dot histograms, respectively, but left and bottom are synonyms, as are right and top. The YORIENTATION option controls the orientation of the y-axis when the histograms are plotted horizontally. By default this is reversed, so that the first histogram is at the top of the screen.

The TITLE and the AXISTITLE options can supply titles for the graph and for the axis along which the values are displayed (i.e. the y-axis when ORIENTATION=vertical). The WINDOW option specifies the window to use for the plot. The default is to use window 3 if PENS has not been set. Alternatively, if PENS has been set, the default is to use window 1, and the KEYWINDOW option specifies the window to use for a key (default 2). The KEYDESCRIPTION option can supply an overall title for the key. The default is to use the name of PENS data structure. The DESCRIPTION parameter can specify the annotation for key. With PENS variates, the default annotation uses their the unique values. With PENS factors it uses their labels if available, or otherwise their levels. The SCREEN option controls whether or not the screen is cleared before plotting.

Usually any units with missing values in the GROUPS factor are ignored, but you can set option CREATEMISSINGLEVEL=yes to create a level for these units. Also, by default, a null (blank) dot histogram is included for levels of the GROUPS factor that are unrepresented in the DATA variate but you can set option OMITEMPTYLEVELS=yes to omit these. This will also omit null dot

histograms for variates with no observations in a DATA pointer. Option SIZE can be used to make the dots smaller or larger. This can alter the number of bins into which the data set is divided. The value of SIZE is reduced if the supplied value would cause dots to overlap.

The SELECTION option can specify one or more statistics to be included in the plot. These can be means or medians or inter-quartile ranges. The BARWIDTH option specifies the width of the bar to be used for each statistic; the default is 10% wider than the widest line of dots in the histogram. The BARTHICKNESS option specifies their thickness. This can be a scalar, or a variate with the same length as the number of selected statistics, specified in the order mean, median and inter-quartile range. The CMEAN, CMEDIAN and CINTERQUARTILE options specify the colours to be used for the means, medians and inter-quartile ranges, respectively. Their default is to use the values specified by the COLOURS parameter or, if COLOURS is not set, they use the colour of pen 1.

Options: TITLE, AXISTITLE, WINDOW, ORIENTATION, YORIENTATION, SCREEN, JUSTIFICATION, CREATEMISSINGLEVEL, OMITEMPTYLEVELS, SIZE, KEYWINDOW, KEYDESCRIPTION, STATISTICS, BARWIDTH, BARTHICKNESS, CMEAN, CMEDIAN, CINTERQUARTILE.
Parameters: DATA, GROUPS, COLOURS, NOBSERVATIONS, PENS, SYMBOLS, DESCRIPTION.

Method
The dot histograms are plotted by DGRAPH, using the solid dot symbol. The colours of the histograms are defined by using the settings from the COLOURS parameter of DOHISTOGRAM in the COLOUR parameter of the PEN directive. See PEN for details of how Genstat interprets strings (defined by a text setting) or numbers (defined by a scalar or variate, or by the levels of a factor setting) as colours.

Action with RESTRICT
DATA variates or the GROUPS factor can be restricted to exclude units from the plot. Restrictions on PENS are ignored.

See also
Directive: DHISTOGRAM.
Procedures: BOXPLOT, DOTPLOT, STEM.
Genstat Reference Manual 1 Summary section on: Graphics.

DOTPLOT

Produces a dot-plot using line-printer or high-resolution graphics (J. Ollerton & S.A. Harding).

Options

GRAPHICS = *string token*	Whether to use high-resolution graphics or line-printer graphics (lineprinter, highresolution); default high
TITLE = *text*	Title for the Dot Plot; default *
WINDOW = *scalar*	Window number for the graph; default 1
SCREEN = *string token*	Whether to clear the screen before plotting or to or continue plotting on the old screen (clear, keep); default clea
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default * uses the current setting
DIRECTION = *string token*	Order in which to sort the data before plotting, DIRECTION=* implies plot unsorted data (ascending, descending); default asce
LINES = *string token*	How to draw guide lines on the plot, LINES=* omits the guide lines (todot, full); default todot draws lines from the *x*-origin to the dots

Parameters

YLABELS = *texts*	Text specifying Y labels for each dotplot
X = *variates*	Data to be plotted
PENDOTS = *scalars*	Pen to draw the dots; default 1
PENLINES = *scalars*	Pen to draw the lines; default 2

Description

DOTPLOT produces a dot-plot from two parameters, a variate of *x*-data and a text containing *y*-labels. Option GRAPHICS allows the plotting to be done using line-printer graphics instead of the default high-resolution graphics.

The display takes the form of a vertical histogram, with a single row for each value of YLABELS. The length of line for each row is specified by the corresponding value of *x*. It is customary to sort the data according to the *x*-values, into either ascending or descending order. This is controlled by the DIRECTION option, which by default is ascending; setting DIRECTION=* will plot the data unsorted.

For high-resolution plots the guide lines can also be drawn across the full width of the plot (LINES=full) or can be omitted (LINES=*). By default, pens are set up to draw the dots and lines in a form appropriate for the output device. For an interactive display, solid guide lines in pale grey are used; for other devices dashed or dotted lines are used. The plotting symbol is symbol 2 (circle), except for PostScript output which uses a solid dot (SYMBOL=-9). The parameters PENDOTS and PENLINES can be used to specify pens which have been set up with different attributes.

By default the dot-plot is produced in window 1, but this can be changed using the WINDOW option. A FRAME statement can be used before using DOTPLOT to change the size and position of the display (for example to widen the x lower margin to allow more space for the *y*-labels). The SCREEN option controls whether or not the screen is cleared before plotting and the ENDACTION option determines what action to take after completing the plot.

An XAXIS or YAXIS statement can be used to set axis titles, and modify the upper and lower bounds of the *x*-axis. If TITLE is unset and axis titles are not set explicitly, they will be generated from the identifier names of the YLABEL and X parameters.

For high-resolution plots, the default window size specifies a lower *x*-margin of size 0.12. This allows room for a title and labels of up to about 10 characters. To produce a dot-plot with longer labels, a FRAME statement should be used to specify new dimensions for the window that include a larger value for XMLOWER. A full-size window, with standard margins, has room for about 48 rows before the labels start to overlap. To produce a dot-plot with more rows the margins should be reduced or the axis pen size reduced.

Options: GRAPHICS, TITLE, WINDOW, SCREEN, ENDACTION, DIRECTION, LINES.
Parameters: YLABELS, X, PENDOTS, PENLINES.

Method
A *y*-variate is constructed with values 1...NVALUES(YLABELS) and plotted against the variate X. If required the variates are sorted (this action is performed on duplicates of the data so as not to alter the original variates).

Action with RESTRICT
DOTPLOT will obey restrictions on either YLABELS or X.

Reference
Cleveland, W.S. (1985). *The Elements of Graphing Data*. Wadsworth advanced books and software.

See also
Directive: DHISTOGRAM.
Procedures: BOXPLOT, DOTHISTOGRAM, RUGPLOT, STEM.
Genstat Reference Manual 1 Summary section on: Graphics.

DPARALLEL

Displays multivariate data using parallel coordinates (Z. Karaman).

Options

TITLE = *text*	Title for the plot
GROUPS = *factor*	Defines grouping of the units (if any); by default, different pens are used for the observations in different groups
PERMUTATIONSALL = *string token*	Whether to display all necessary permutations so that any two variates will be adjacent in at least one plot, or just display once in the order given by the DATA pointer (yes, no); default no
SCALING = *string token*	Whether to do scaling overall (scale all variates on the same scale), or to scale each variate separately (overall, separate); default sepa
PEN = *variate*	Pens to be used for different groups (if any); default * uses pens from 1 up to the number of groups (number of levels of the GROUPS factor)

Parameter

DATA = *variates*	Data variables to be plotted

Description

The scatter plot is probably the most powerful and most frequently used statistical tool for analysing the relationship between two variables. It is very intuitive way to look at the data since it corresponds to our perception of the world. The major drawback is that it does not generalize naturally to higher dimensions. Using interactive graphics devices like high-resolution screens one can rotate a point cloud in three dimensions (commonly called spinning), and further dimension can be partially encoded by using different colours, symbols, or symbol sizes; however, this technique can be used only on interactive graphics devices, and it is difficult to see relationships between all the variables at a time. Another possibility is the matrix of scatter plots (provided by procedure DSCATTER), but this has the drawback that it is difficult to follow one data point across several plots.

An alternative is to display multivariate data using parallel coordinates. The dimensions are not represented by orthogonal lines as is customary done when plotting scatter diagrams (which limits the dimensionality to two, or at most three if spinning is used). Rather, they are represented by a series of parallel lines (either horizontal or vertical), and a point in a multidimensional space is represented by a broken line connecting its coordinates in each dimension. The only limit on the number of dimensions that can be displayed simultaneously by such plot is its readability, which is a function of the underlying graphics display (hardware). The parallel coordinates geometry was developed by Inselberg (1985) in the context of computational geometry; it was applied to statistical multidimensional analysis by Wegman (1990). Inselberg also gives some interesting duality properties between classical Euclidean plane and parallel coordinates geometry.

The relationship between two variables can be visually assessed by inspecting a parallel coordinates plot. When the correlation between two variables is close to −1, the lines are crossing over and so, in the limit, we would have a pencil of lines. (A pencil of lines is a set of lines that are coincident at a single point.) On the other hand, when the correlation approaches +1, we will have fewer and fewer crossovers, so that in the limit we would have a set of parallel lines. The pairwise comparisons are easy for variables represented by adjacent axes; however, they are much more difficult for the axes far away on the graph. For n variables, there are $n!$

possible permutations, but many of these duplicate adjacencies. Wegman (1990) has shown that with a relatively small number of permutations of the axes (approximately $n/2$) one can achieve that in some permutation every variable is adjacent to every other variable. Multivariate outliers can be identified easily on this plot, since it is very intuitive to follow with one's eye the line across the axes. If the PERMUTATIONSALL option is set to yes, several plots will be produced so that every pair of variables is adjacent in at least one plot.

In our implementation we have chosen to dispose the axes vertically, since this way the readability is maximized for most output devices (either terminal screens or printers when printing in landscape mode). The variables can be independently scaled on a 0 to 1 scale, or left in original units if the values are of the same order of magnitude. In the first case it is easier to have an visual estimate of the correlation between the two adjacent variables; on the other hand, leaving the data in original units gives us a good idea of the location and spread parameters of the marginal distributions.

The data are specified, in a list of variates, using the DATA parameter. The GROUPS option can be used to specify a grouping factor. The lines for observations in each group are then plotted using different pens, thus giving an immediate insight to any patterns in data. By default, pens 1 upwards are used for the different groups, but the PEN option can be used to specify other pens, in a variate with as many values as groups. If the GROUPS option is not set, the PEN option can be set to a scalar, to select the pen to be used for all the points. The TITLE option can be used to supply a title for the plots.

Options: TITLE, GROUPS, PERMUTATIONSALL, SCALING, PEN.
Parameter: DATA.

Method

DPARALLEL uses the standard Genstat directives for data manipulation and graphics. The underlying methodology is described by Inselberg (1985) and Wegman (1990). It calls subsidiary procedure _DPARWEGMAN to generate the permutations matrix; each column of the output matrix gives one of the permutations described by Wegman (1990).

Action with RESTRICT

Restrictions are not allowed. Missing values are allowed within the input variates in DATA; the observations with missing data are not excluded form the plot, but will have the parts of their broken lines adjacent to the missing value missing from the plot.

References

Inselberg, A. (1985). The plane with parallel coordinates. *The Visual Computer*, **1**, 69-91.

Wegman, E. (1990). Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, **85**, 664-675.

See also

Procedure: DSCATTER.

Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

DPOLYGON

Draws polygons using high-resolution graphics (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

TITLE = *text*	Main title for the plot; default *
WINDOW = *scalar*	Which graphics window to use for the plot; default 1
KEYWINDOW = *scalar*	Which graphics window to use for the key; default 2
YTITLE = *text*	Title for the vertical axis; default *
XTITLE = *text*	Title for the horizontal axis; default *
YLOWER = *scalar*	Lower limit for the vertical axis
YUPPER = *scalar*	Upper limit for the vertical axis
XLOWER = *scalar*	Lower limit for the horizontal axis
XUPPER = *scalar*	Upper limit for the horizontal axis
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
KEYDESCRIPTION = *text*	Overall description for the key; default *
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default paus

Parameters

YPOLYGON = *variates*	Vertical coordinates of one or more polygons; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of one or more polygons; no default – this parameter must be set
PEN = *scalars* or *variates* or *factors*	Pen number for each graph
DESCRIPTION = *texts*	Annotation for the key

Description

DPOLYGON draws polygons onto the current graphics device. Parameters XPOLYGON and YPOLYGON specify variates containing the horizontal and vertical coordinates of the polygons. DPOLYGON uses procedure DPTMAP to produce the plot. This uses the AXES and FRAME directives to set up axes with equal scales. Options YLOWER, YUPPER, XLOWER and XUPPER can be used to specify bounds for the axes, or these can be set automatically. The axes are made to extend slightly beyond the range of values to be plotted, and are drawn using the box style. Titles for the horizontal and vertical axes can be specified using the XTITLE and YTITLE options, respectively. Options TITLE, WINDOW, KEYWINDOW, SCREEN, KEYDESCRIPTION and ENDACTION are as in DGRAPH.

By default, DPOLYGON uses a different pen for each polygon. The sequence of pens is the same as the default sequence of pens used by DGRAPH but the pens are set to use METHOD=line, SYMBOLS=0 and JOIN=given, so that each polygon is drawn as a sequence of connected line segments. Other pen styles can be specified using the PEN parameter, except that the procedure will override settings of METHOD, SYMBOLS and JOIN, replacing them by METHOD=line, SYMBOLS=0 and JOIN=given. The original settings will be restored on exiting the procedure. To draw polygons in a different style, for example, using lines and points, you can use DPTMAP directly, with an appropriate PEN setting, rather than DPOLYGON.

Options: TITLE, WINDOW, KEYWINDOW, YTITLE, XTITLE, YLOWER, YUPPER, XLOWER, XUPPER, SCREEN, KEYDESCRIPTION, ENDACTION.
Parameters: YPOLYGON, XPOLYGON, PEN, DESCRIPTION.

Method
A procedure PTCHECKXY is called to check that each pair of structures in XPOLYGON and YPOLYGON have identical restrictions. If the PEN parameter is unset then pens with METHOD=line and SYMBOLS=0 will be specified using the PEN directive. PTCLOSEPOLYGON is used to close the polygons and DPTMAP to draw them.

Action with RESTRICT
If any of the variates in XPOLYGON and YPOLYGON are restricted, only the subset of values specified by the restriction will be included in the graph.

See also
Procedures: DPTMAP, PTAREAPOLYGON, PTCLOSEPOLYGON, PTSINPOLYGON.
Genstat Reference Manual 1 Summary sections on: Graphics, Spatial statistics.

DPROBABILITY

Plots probability distributions, and estimates their parameters (D.B. Baird).

Options

PRINT = *string tokens*	Controls whether to print estimated parameters of the distribution or test statistics (parameters, tests); default para
DISTRIBUTION = *string token*	Distribution for expected values against which to plot values (normal, stdnormal, lognormal, exponential, gamma, weibull, beta, b2, pareto, chisquare, cauchy, logistic, ev1, ev2, ev3, gev, invnormal, t, f, uniform, stduniform, laplace, gpareto, ubetamix, ugammamix, loggamma, loglogistic, paralogistic, igamma, iweibull, burr, iburr); default norm
METHOD = *string token*	Method used for the plot axes (quantile, probability, stabilizedprobability); default quan
GRAPHICS = *string token*	Type of graphics (highresolution, lineprinter); default high
PLOT = *string tokens*	Whether to plot differences from expectations or the 1-1 reference line (differences, reference); default refe
CONSTANT = *string token*	Whether to estimate the constant for the distribution (estimate, omit) default omit
BANDS = *string token*	What type of confidence bands to plot, if any (simultaneous, pointwise); default simu
NSIMULATIONS = *scalar*	Number of simulations for pointwise bands; default 100
ALPHA = *scalar*	Acceptance limits for confidence bands; default 0.95
DF = *scalar*	Number of degrees of freedom of chi-square or t distribution; default 1
DFNUMERATOR = *scalar*	Numerator degrees of freedom of F distribution; default 1
DFDENOMINATOR = *scalar*	Denominator degrees of freedom of F distribution; default 1
WINDOW = *scalar*	Window to use for the plot; default 3
XMETHOD = *string token*	Scaling of X / Expected Plot axes (quantile, probability, stabilizedprobability); if unset, takes the same setting as METHOD
QMETHOD = *string token*	Whether to standardize plotted score in expected quantiles (standardized, unstandardized); default stan
TMETHOD = *string tokens*	Specifies the method used to perform the goodness-of-fit tests (likelihoodratio, traditional); default like
NTIMES = *scalar*	Number of Monte-Carlo simulations to perform for likelihood-ratio tests; default 999
SEED = *scalar*	Seed for random number generation for the likelihood-ratio tests; default 0 continues an existing sequence or, if none, selects a seed automatically

Parameters

DATA = *variates*	Values to plot
TITLE = *text*	Title for the graph; default * generates an appropriate title automatically
ESTIMATES = *variates*	Saves the estimated parameters for the distribution
SE = *variates*	Saves standard errors for the estimated parameters
LOWERTRUNCATION = *scalars*	Lower truncation points for Loss distributions
UPPERTRUNCATION = *scalars*	Upper truncation points for Loss distributions
DEVIANCE = *scalars*	Saves the deviance for the fitted distribution
PROBABILITIES = *variates*	Saves the probabilities from the goodness-of-fit tests

Description

To assess the how well empirical data approximates a particular theoretical distribution, DPROBABILITY plots the sorted values (order statistics, X_i) against the expected values of the order statistics E_i from the given distribution. However, usually the particular parameters of the distribution are not known and these have to be estimated first to obtain the expected values.

If the distribution has a cumulative density function of F(x), and the inverse of this function is G(x) (i.e. G(F(x)) = x), then the expected values of the order statistics, are approximately G((i−0.5)/n), where i = 1...n, and n is the number of values in the sample. A plot of X_i versus E_i is known as a Quantile-Quantile (or Q-Q) plot. The data can also be plotted on the probability scale by plotting the cumulative probabilities of the data under the assumed distribution against their expected probabilities, i.e. F(X(i)) versus (i−0.5)/n. This is known as a Probability-Probability (or P-P) plot.

A third plot called the stabilized probability (SP) plot (Michael 1983), was introduced, which rescales the probabilities using the transformation

$$sp = (2/\pi) \times \text{ARCSIN}(\text{SQRT}(p))$$

so that the variance of the plotted points is approximately equal over the range of probability values. In the SP plot the scaled values sp are plotted rather than the unscaled p values. The METHOD option allows the choice of which scale is used in the graph (quantile, probability or stabilizedprobability for the Q-Q, P-P or SP plots respectively).

By default the x-value used in plotting Q, P or SP is the corresponding expected value of these statistics. Alternative x-values can be used by setting the XMETHOD option to quantile, probability, or stabilizedprobability. So for example a Q-P plot can be obtained with the option settings METHOD=quantile and XMETHOD=probability or a P-Q plot with the settings METHOD=probability and XMETHOD=quantile.

The QMETHOD option allows the scaling of the expected quantiles plotted on the x-axis to be set. By default quantiles are standardized to have a mean of zero and variance of one (as in a normal score plot) but, if QMETHOD=unstandardized, the quantiles are scaled to the same mean and variance as the data.

The DATA parameter specifies the data values, in a variate. The TITLE parameter can specify a title for the graph. The ESTIMATES parameter can be used to save the values estimated for the parameters for the distribution, and the SE parameter can save their standard errors.

The distribution for the expected values against which to plot the data is specified by the DISTRIBUTION option. Some distributions (Log-Normal, Gamma, Weibull and Pareto) can have an extra parameter (a) estimated, so that $X-a$ follows the specified distribution. Setting option CONSTANT=estimate estimates a value for a. Some of the distributions (Chi Square, T and F) cannot have the parameters estimated by the usual DISTRIBUTION directive, so the procedure provides 3 options (DF, DFNUMERATOR, DFDENOMINATOR) for specifying the parameters of these distributions. However, if for example you set DF=*, the degrees of freedom are estimated along with the other parameters of the distribution.

Some distributions (normal, loggamma, loglogistic, paralogistic, igamma,

iweibull, burr, iburr) can be estimated and plotted in a truncated form. The values in the distribution less than LOWERTRUNCATION and greater than UPPERTRUNCATION are removed (if either of these are set), and the distribution between these limits is rescaled to have an area of one. If only LOWERTRUNCATION is set, the distribution is left-truncated, and it is right-truncated if only UPPERTRUNCATION is set.

The BANDS option allows two forms of confidence intervals to be displayed in the graph. BANDS=pointwise simulates NSIMULATIONS distributions of the same size as the data, from the theoretical distribution, and plots the range of values at each value of the order statistics that contain the proportion specified by the option ALPHA of simulated values. Thus a sample drawn from the assumed distribution has approximately a probability ALPHA of lying within the limits at each point. However, overall there will be a probability of less than ALPHA that a sample will completely lie within the confidence bands. The BANDS=simultaneous uses a statistic given by Michael (1983) for which the overall probability of plotted data lying completely within the confidence bands is approximately the specified value of ALPHA, under the null hypothesis that the data is a random iid sample from the specified distribution. This form of confidence limits has the advantage that it is much faster to calculate and that probability of the data points falling outside the limits is approximately constant over the range of the data.

When plotting the data against the expected values, setting option PLOT=reference allows the 1-1 line to be added to the graph, so that departures from this can be more easily observed. The other PLOT setting, difference, plots the difference between the data and the expected values, so that departures can be observed more easily in a horizontal direction rather than on a 45 degree slant. Setting option GRAPHICS=lineprinter produces a character based graph in the output window rather than in the high-resolution graphics window as usual. The WINDOW option can be used to specify which graphics window to use for a high-resolution graph.

The PRINT option control of the output that is printed. The parameters setting prints the fitted parameters of the specified distribution, and some sample statistics of the observed data. The test setting provides output from three empirical distribution tests, namely the Anderson-Darling, Cramer-von Mises and Watson statistics. The method used to perform these tests is specified by the TMETHOD option, with settings likelihoodratio for the Zhang (2002) likelihood-ratio based method, and traditional for the traditional approach. The default is to use the likelihood-ratio based tests, which are generally more powerful. Monte-Carlo simulations are used to calculate the empirical probability values of the test statistics under the likelihood-ratio based method. The NTIMES option defines how many Monte-Carlo simulations are used; default 999. The SEED option specifies the seed for the random-number generator used during the Monte-Carlo simulations. The default of zero continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, the seed is initialized automatically. The test probabilities can be saved, in a variate, by the PROBABILITIES parameter.

The distributions fitted in this procedure are described further in the books by Hogg & Klugman (1984) and Johnson, Kotz & Balakrishnan (1994, 1995).

Options: PRINT, DISTRIBUTION, METHOD, GRAPHICS, PLOT, CONSTANT, BANDS, NSIMULATIONS, ALPHA, DF, DFNUMERATOR, DFDENOMINATOR, WINDOW, XMETHOD, QMETHOD, TMETHOD, NTIMES, SEED.

Parameters: DATA, TITLE, ESTIMATES, SE, LOWERTRUNCATION, UPPERTRUNCATION, DEVIANCE, PROBABILITIES.

Method

The parameters for the distribution are estimated using the DISTRIBUTION or FITNONLINEAR directives. The cumulative distribution probability values of the observed and expected values are calculated with the CL series of functions. The goodness-of-fit tests are performed by the

EDFTEST procedure.

Action with RESTRICT

If the DATA variate is restricted, the plots and tests will be calculated using only the units included by the restriction.

Reference

Hogg, R. V. & Klugman, S. A. (1984). *Loss Distributions*. John Wiley & Sons, New York.

Johnson, N. L., Kotz, S. & Balakrishnan N. (1994). *Continuous Univariate Distributions, Volume 1, 2nd edition*. John Wiley & Sons, New York.

Johnson, N. L., Kotz, S. & Balakrishnan N. (1995). *Continuous Univariate Distributions, Volume 2, 2nd edition*. John Wiley & Sons, New York.

Michael, J. R. (1983). The stabilized probability plot. *Biometrika*, **70**, 11-17.

Zhang (2002). Powerful goodness-of-fit tests based on the likelihood ratio. *Journal of the Royal Statistical Society, Series B*, **64**, 281-294.

See also

Directive: DISTRIBUTION.

Procedures: BBINOMIAL, EDFTEST, MAVOLCANO.

Genstat Reference Manual 1 Summary sections on: Graphics, Basic and nonparametric statistics.

DPSPECTRALPLOT

Calculates an estimate of the spectrum of a spatial point pattern (C.J. Alexander & D.A.Murray).

Options

PLOT = *string tokens*	Which graphs to plot (periodogram, rspectrum, thetaspectrum, weights); default peri, rspe, thet, weig
NROWS = *scalar*	Number of rows for periodogram; default 17
NCOLUMNS = *scalar*	Number of columns for periodogram; default 32
SCALING = *string token*	Whether to normalize the coordinates of the points within the study region to a unit square (normalize, none); default norm

Parameters

Y = *variates*	Vertical coordinates of each spatial point pattern
X = *variates*	Horizontal coordinates of each spatial point pattern
YPOLGON = *variates*	Y-coordinates for the rectangular study region
XPOLYGON = *variates*	X-coordinates for the rectangular study region
YHOLEPOLGON = *variates*	Y-coordinates for the missing region polygons
XHOLEPOLYGON = *variates*	X-coordinates for the missing region polygons
HOLEGROUPS = *variates*	Grouping factor where each level represents a different polygon for the missing regions.
PERIODOGRAM = *matrices*	Saves the periodogram
WEIGHTS = *variates*	Saves the weights used for the inter-event calculation
YINTEREVENT = *variates*	Saves the y-coordinates for the inter-event calculation
XINTEREVENT = *variates*	Saves the x-coordinates for the inter-event calculation

Description

Spectral analysis looks for evidence of spatial structure in a point pattern by examining its second-order properties. Formally the spectral density function, denoted by $f(\omega)$, is defined as the Fourier transform of the auto-covariance function, $\kappa(c)$, such that

$$f(\omega) = \int \kappa(c) \exp(-i\, \omega' \, c)\, dc$$

where c is the vector separating two points in the region and $i=\sqrt{-1}$ (Bartlett 1964). An estimator of this spectral density function (or periodogram) can be calculated using the discrete Fourier transform of the coordinates from the set of N events of the spatial point pattern observed within a rectangular study region

$$F(p, q) = \sum_{j=1\ldots N} \exp\{ -2\pi i\, (p x_j^* + q y_j^*) \}$$

This is evaluated at integer values of p (= 0, ±1, ±2 ...) and q (= 0, ±1, ±2 ...) which represent the frequency over the x- and y-coordinate directions. In order to reduce bias near $p = q = 0$, it is usual to standardise the original coordinates so that x_j and y_j are rescaled from the rectangular study region to x_j^* and y_j^* within the unit square. The periodogram is then

$$f(\omega_P, \omega_Q) = F(p, q)\, F^-(p, q)$$

where $F^-(p, q)$ is the complex conjugate of the discrete Fourier transform (Bartlett 1964). Thus, the estimated spectrum analyses the point pattern for important features of both the distribution's scale and direction. For a pattern observed within a rectangular boundary, the periodogram will be approximately unbiased. However, if there are missing regions within this outer boundary, the "holes" will produce positive bias, primarily in the low order frequencies. This can be corrected using a method which weights the contribution of a particular inter-event vector in the calculation of the periodogram. Each inter-event vector is weighted proportionate to the amount to which it is affected by the presence of the missing regions and is also conditional on the

observed configuration of the spatial point pattern relative to these holes (Alexander 2006).

Although the periodogram is informative about directional as well as scale effects, it is often easier to look at these aspects through its polar representation. Mugglestone & Renshaw (1996) show that any ordinate of the periodogram $\hat{f}(\omega_P, \omega_Q)$ can be represented in the form $\hat{g}(\omega_r, \omega_\theta)$ where $r = \sqrt{(p^2 + q^2)}$ and $\theta = \tan^{-1}(p / q)$. The R-spectrum is then defined as

$$\hat{f}_R(r) = 1 / n_r \sum_{r'} \sum_{\theta} \hat{g}(\omega_{r'}, \omega_\theta) \qquad r = 1, 2 \ldots$$

where the summation is over the n_r ordinates of the periodogram where $r - 1 < r' \leq r$. This is used to investigate scales of pattern under the assumption of isotropy by averaging over those ordinates that have similar values of r. Further, the θ-spectrum is defined to be

$$\hat{f}_\zeta(\zeta) = (1/n_\theta)\, g_\wedge(\omega_r, \omega_{\theta'}) \qquad \theta = 0°, 10°, 20° \ldots 170°$$

with the summation being over the n_Θ ordinates for which

$$\theta - 5° < \theta' < \theta + 5°.$$

The homogeneous Poisson process represents complete spatial randomness (CSR) in that it generates points which are uniformly and independently distributed over the study region. This acts as a reasonable null hypothesis against which to compare any observed point pattern. The spectrum for a CSR point pattern has each ordinate equal to 2 for all values of p and q which demonstrates that there are no features of scale or direction discernable. For a clustered pattern, there will be a peak in periodogram values at the lowest frequencies which will fall off as frequency increases. Regular patterns show lower periodogram values (relative to CSR) for the lowest frequency ordinates rising to a peak at the frequency which captures the regularity of the spatial structure. Patterns with any directional structure will show peaks in periodogram values for (p, q) ordinates which match the orientation and scale of that spatial feature.

`DPSPECTRALPLOT` calculates the estimated spectral density (or periodogram) given the coordinates of a spatial point pattern (specified by the parameters `X` and `Y`) observed within a rectangular study region (specified by the parameters `XPOLYGON` and `YPOLYGON`). Coordinates of missing regions can be supplied by the `XHOLEPOLYGON` and `YHOLEPOLYGON` parameters. If there is more than one missing region then an additional factor should be supplied using the `HOLEGROUPS` parameter, where each level defines a different missing region.

The output of the procedure is a matrix of values of the periodogram estimated over a range of p and q (specified by the `NROWS` and `NCOLUMNS` options). The estimates of the spectrum can be saved using the parameter `PERIODOGRAM`. Weights used for the conditional correction can be saved in a variate using the parameter `WEIGHTS` and the associated inter-event vector coordinates can be saved using the `XINTEREVENT` and `YINTEREVENT` parameters.

If the `XPOLYGON` and `YPOLYGON` parameters do not define the unit square (0,1), then the `normalize` setting of the `SCALING` option can be used to standardise the original coordinates so that the `X` and `Y` are rescaled from the rectangular study region to within the unit square.

The type of plot is controlled using the `PLOT` option. The `periodogram` setting produces a shade plot of the periodogram matrix and, if any missing regions are the supplied, the `weight` setting will plot the conditional weights versus the inter-event vector distances. The `rspectrum` and `thetaspectrum` settings produce a plot of the polar R-spectrum and Θ-spectrum respectively.

Options: `PLOT`, `NROWS`, `NCOLUMNS`, `SCALING`.

Parameters: `Y`, `X`, `YPOLYGON`, `XPOLYGON`, `YHOLEPOLYGON`, `XHOLEPOLYGON`, `HOLEGROUPS`, `PERIODOGRAM`, `WEIGHTS`, `YINTEREVENT`, `XINTEREVENT`.

Method

For estimating the conditional correction, `DPSPECTRALPLOT` calls `PTCLOSEPOLYGON` to close the polygons specified by `XHOLEPOLYGON`, `YHOLEPOLYGON` and `HOLEGROUPS`. `DPSPECTRALPLOT` then calls procedure `PTPASS` to execute a Fortran program to calculate an estimate of the spectrum corrected for the influence of any missing regions. Similarly `PTPASS`

is used to execute other Fortran programs to calculate the periodogram, r- and θ-spectra.

Action with `RESTRICT`

The variates `X`, `Y`, `XPOLYGON`, `YPOLYGON`, `XHOLEPOLYGON`, `YHOLEPOLYGON` and factor `HOLEGROUPS` may be restricted, as long as `X` has the same restriction as `Y`, `YHOLEPOLYGON` has the same restriction as `XHOLEPOLYGON`, and `XPOLYGON` has the same restriction as `YPOLYGON` and `GPOLYGON`. Only the subset of values specified by each restriction will be included in the calculations.

References

Alexander, C.J. (2006). *Spectral Analysis of Spatial Point Patterns in Complex Study Regions*. PhD Thesis, Open University.

Bartlett, M. S. (1964). The spectral analysis of two-dimensional point processes. *Biometrika*, **51**, 299-311

Mugglestone, M.A. & Renshaw, E. (1996). A practical guide to the spectral analysis of spatial point patterns. *Computational Statistics and Data Analysis*, **21**, 43-65

See also

Directive: `FOURIER`.

Procedures: `DFOURIER`, `MCROSSPECTRUM`, `SMOOTHSPECTRUM`.

Genstat Reference Manual 1 Summary section on: Spatial statistics.

DPTMAP

Draws maps for spatial point patterns using high-resolution graphics (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

TITLE = *text*	Main title for the plot; default *
WINDOW = *scalar*	Which graphics window to use for the plot; default 1
KEYWINDOW = *scalar*	Which graphics window to use for the key; default 2
YTITLE = *text*	Title for the vertical axis; default *
XTITLE = *text*	Title for the horizontal axis; default *
YLOWER = *scalar*	Lower limit for the vertical axis
YUPPER = *scalar*	Upper limit for the vertical axis
XLOWER = *scalar*	Lower limit for the horizontal axis
XUPPER = *scalar*	Upper limit for the horizontal axis
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
KEYDESCRIPTION = *text*	Overall description for the key; default *
ENDACTION = *string token*	Action to be taken after completing the plot (continue, pause); default paus

Parameters

Y = *variates*	Vertical coordinates of one or more spatial point patterns; no default – this parameter must be set
X = *variates*	Horizontal coordinates of one or more spatial point patterns; no default – this parameter must be set
PEN = *scalars* or *variates* or *factors*	Pen number for each graph
DESCRIPTION = *texts*	Annotation for the key

Description

DPTMAP is a specially adapted version of DGRAPH designed for producing maps of spatial point patterns. The procedure uses the AXES and FRAME directives to set up axes with equal scales. Options YLOWER, YUPPER, XLOWER and XUPPER can be used to specify bounds for the axes, or these can be set automatically. The axes are made to extend slightly beyond the range of values to be plotted, and are drawn using the box style. The parameters X and Y specify pointers to variates containing the horizontal and vertical coordinates of one or more spatial point patterns. Titles for the horizontal and vertical axes can be specified using the XTITLE and YTITLE options, respectively. Options TITLE, WINDOW, KEYWINDOW, SCREEN, KEYDESCRIPTION and ENDACTION are as in DGRAPH.

Options: TITLE, WINDOW, KEYWINDOW, YTITLE, XTITLE, YLOWER, YUPPER, XLOWER, XUPPER, SCREEN, KEYDESCRIPTION, ENDACTION.
Parameters: Y, X, PEN, DESCRIPTION.

Method

A procedure PTCHECKXY is called to check that each pair of structures in X and Y have identical restrictions. If any of YLOWER, XUPPER, YLOWER and YUPPER are unset, the procedure PTBOX is used to assign suitable values based on the data in X and Y. The values of these options are then adjusted to extend the range of the axes and so produce a more attractive plot. The adjusted values are given by

```
XLOWER - 0.05 * range(X),
XUPPER + 0.05 * range(X),
YLOWER - 0.05 * range(Y),
YUPPER + 0.05 * range(Y),
```

where range(X) is the range of values in X and range(Y) is the range of values in Y. The AXES directive is then used to set up box-style axes with the required upper and lower limits and titles specified by XTITLE and YTITLE. The FRAME directive is used to ensure equal scales on the horizontal and vertical axes. Finally, the DGRAPH directive is used to draw the map on the current graphics device.

Action with RESTRICT

If any of the variates in X and Y are restricted, only the subset of values specified by the restriction will be included in the graph.

See also

Procedure: DPOLYGON.

Genstat Reference Manual 1 Summary sections on: Graphics, Spatial statistics.

DPTREAD

Adds points interactively to a spatial point pattern (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string token*	What to print (summary, monitoring); default summ, moni
WINDOW = *scalar*	Which graphics window to use for the plot; default 1

Parameters

OLDY = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
OLDY = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
NEWY = *variates*	Variates to receive the vertical coordinates of the original points and added points
NEWX = *variates*	Variates to receive the horizontal coordinates of the original points and added points

Description

DPTREAD uses the DREAD directive to add points to a spatial point pattern. The coordinates of the existing points must be supplied using the parameters OLDX and OLDY. These points will be plotted on the current graphics device using DPTMAP with a pen setting of SYMBOLS=1. The WINDOW option may be used to specify the graphics window to use for the plot.

DREAD is not always available, and its operation may vary slightly from one system to another. The Users' Note supplied with Genstat explains how to read points and terminate input on specific devices. The usual method for reading points is to click the left mouse button at the required position. The usual way to terminate input is to click the right mouse button. The points read using DREAD will be echoed using a pen setting of SYMBOLS=2. The coordinates of the new spatial point pattern containing the original points and any points which have been added may be saved using the parameters NEWX and NEWY.

Printed output is controlled using the PRINT option. The settings available are monitoring (which prints the coordinates of the points to be added) and summary (which prints the coordinates of the new pattern consisting of the original points and any that have been added under the headingss NEWX and NEWY). The default setting is for both monitoring and summary.

Options: PRINT, WINDOW.
Parameters: OLDY, OLDX, NEWY, NEWX.

Method

A procedure PTCHECKXY is called to check that OLDX and OLDY have identical restrictions. DPTMAP is used to draw a map of the original point pattern. The DREAD directive is then used to read the coordinates of points to be added. Finally, the coordinates for the original points and added points are combined in new variates using the EQUATE directive.

Action with RESTRICT

If OLDX and OLDY are restricted, only the subset of values specified by the restriction will be included in the calculations.

See also
Procedure: DRPOLYGON, PTREMOVE.
Genstat Reference Manual 1 Summary sections on: Graphics, Spatial statistics.

DQMAP

Displays a genetic map (D.A. Murray).

Options

ORIENTATION = *string token*	Orientation of map (vertical, horizontal); default vert
DCHROMOSOMES = *variate*, *text* or *scalar*	To specify a subset of the linkage groups to be displayed
TITLE = *text*	General title; default *

Parameters

CHROMOSOMES = *factors*	Factor defining the linkage groups
POSITIONS = *variates* or *pointers*	Positions of markers within the linkage groups
MKNAMES = *texts*	Names of the markers
QCHROMOSOMES = *factors*	Factor defining the linkage groups of the QTLs
QPOSITIONS = *variates*	Positions of QTLs within the linkage groups
QNAMES = *texts*	Names of the QTLs
QINTERACTIONS = *variates*	Logical variate indicating whether the QTL has significant (1) or non-significant (0) QTL-by-environment interaction

Description

DQMAP plots a genetic map of marker locations within linkage groups. The linkage groups are supplied in a factor by the CHROMOSOMES parameter. The positions within the linkage groups are supplied by the POSITIONS parameter in a variate or in a pointer of 2 variates; if a pointer is supplied, the positions of the same marker are connected by a straight line. The names of the markers can be supplied in a text using the MKNAMES parameter. In the Graphics viewer in Genstat *for Windows*, the marker names can be viewed using the *Data Info* tool.

QTLs can be displayed on the plot by supplying the linkage groups of the QTLs in a factor using the QCHROMOSOMES parameter, and the positions within the linkage groups in a variate using the QPOSITIONS parameter. The names for the QTLs can be supplied in a text using the QNAMES parameter. If no names are supplied, the QTLs will be labelled using Q1, Q2 and so on. When displaying QTLs from a multiple environment analysis, the QINTERACTIONS parameter can be used to supply a logical variate indicating whether the QTL has significant (1) or non-significant (0) QTL-by-environment interactions. The QTLs identified as having QTL-by-environment interactions will then be displayed in a different colour.

The DCHROMOSOMES option can be used to display a subset of the linkage groups. The setting can be either a variate or a scalar of the group number defining a subset of the levels of the CHROMOSOMES factor, or a text defining a subset of its labels. The ORIENTATION option controls whether the map is plotted vertically (the default) or horizontally.

The TITLE option can be used to provide a title for the graph.

Options: ORIENTATION, DCHROMOSOMES, TITLE.

Parameters: CHROMOSOMES, POSITIONS, MKNAMES, QCHROMOSOMES, QPOSITIONS, QNAMES, QINTERACTIONS.

Action with RESTRICT

Any restrictions will be ignored.

See also

Procedures: DQMKSCORES, DQMQTLSCAN, DQSQTLSCAN, QMKDIAGNOSTICS.

Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

DQMKSCORES

Plots a grid of marker scores for genotypes and indicates missing data (D.A. Murray).

Options

PLOT = *string token*	Type of plot (missing, all); default miss
LOWERGENOTYPE = *scalar*	Lower genotype for the display
UPPERGENOTYPE = *scalar*	Upper genotype for the display
DCHROMOSOMES = *variate*, *text* or *scalar*	Specify a subset of the linkage groups to be displayed
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, AMP); must be set
COLOURS = *text* or *variate*	Colours to use for the different marker scores
TITLE = *text*	Title for the graph

Parameters

MKSCORES= *pointers*	Marker score code for each marker
CHROMOSOMES = *factors*	Linkage group for each marker
PARENTS = *pointers*	Parent information
IDPARENTS = *texts*	Labels to identify the parents

Description

DQMKSCORES produces graphical displays of marker scores for a set of genotypes. The marker scores should be supplied by setting the MKSCORES parameter to a pointer containing to a set of factors (one for each marker). Each factor should have same labels, in the same order. The linkage groups for each marker are supplied in a factor by the CHROMOSOMES parameter.

The PLOT option controls the type of graph that is displayed:

all	produces a shade plot where the genotypes are displayed using different colours, and
missing	produces a shade plot displaying the missing marker genotypes.

The LOWERGENOTYPE and UPPERGENOTYPE options can be used to display a subset of genotypes, by supplying values for the lower and upper genotypes to be included in the plot.

The DCHROMOSOMES option can be used to display a subset of the linkage groups. The setting can be either a variate or a scalar referring to the levels of the CHROMOSOMES factor, or a text referring to its labels.

The type of population must be specified using the POPULATIONTYPE option. The following settings are available:

BC1	first generation backcross population,
DH1	doubled-haploid population,
F2	an F2 population,
RIL	population of recombinant inbred lines,
BCxSy	population of backcross inbred lines, and
AMP	data for association mapping.

The parent information must be supplied using the PARENTS parameter in a pointer to a set of texts. The first text in the pointer defines the alleles for parent 1, the second text defines the allele for parent 2, and so on. The labels for the parents are supplied in a text using the IDPARENTS parameter.

The COLOURS option can specify colours for the different marker scores. The colours can be supplied in a variate of RGB colours or in a text containing names of Genstat's predefined colours (see PEN). For a plot of the marker scores, the number of colours specified by the text or variate should match the number of different marker scores for the population, while for a

missing-score plot they should represent those that are present, missing and partially missing (`F2`, `RIL` and `BCxSy` populations). For example, for a plot of the marker scores for a `DH1` population, three colours should be supplied for the scores 1/1, 2/2 and `-/-`. Similarly, for a missing genotype plot for a `DH1` population, two colours should be supplied to represent the present and missing scores.

The `TITLE` option allows you to supply a title for the plot.

Options: `PLOT`, `LOWERGENOTYPE`, `UPPERGENOTYPE`, `DCHROMOSOMES`, `POPULATIONTYPE`, `COLOURS`, `TITLE`

.

Parameters: `MKSCORES`, `CHROMOSOMES`, `PARENTS`, `IDPARENTS`.

Action with `RESTRICT`

Any restrictions are ignored.

See also

Procedures: `DQMAP`, `DQMQTLSCAN`, `DQSQTLSCAN`, `QMKDIAGNOSTICS`.

Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

DQMQTLSCAN

Plots the results of a genome-wide scan for QTL effects in multi-environment trials (M.P. Boer & J.T.N.M. Thissen).

Options

`POPULATIONTYPE` = *string token*	Type of population (`BC1`, `DH1`, `F2`, `RIL`, `BCxSy`, `CP`); must be set
`METHOD` = *string token*	Method to be used for plotting (`line`, `manhattan`, `spikes`); default `line`
`THRESHOLD` = *scalar*	Threshold value for test statistic; default 0
`DCHROMOSOMES` = *scalar*, *text* or *variate*	Allows a subset chromosomes to be specified to display; default `*` i.e. all the chromosomes
`SUPPRESSLINES` = *string token*	Whether to suppress the vertical lines between the chromosomes (`yes`, `no`); default `no`
`SYMBOL` = *scalar*	Defines the plotting symbol for each point, as in the `SYMBOL` option of `PEN`, when `METHOD=manhattan`; default 2 i.e. circle
`SIZEMULTIPLIER` = *scalar*	Multiplier used in the calculation of sizes of symbols when `METHOD=manhattan`; default 1
`BLACKOUTLINE` = *string token*	Whether to draw the outer line the `SYMBOL` in black when `METHOD=manhattan` (`yes`, `no`); default `no`
`COLOURS` = *scalar*, *variate* or *text*	Colours to use for the chromosomes; default `*` uses the colours of pens 1, 2 up to the number of chromosomes
`TITLE` = *text*	General title
`YLOWERTITLE` = *text*	Title for the y-axis of the lower graph; default `'Environments'`
`YUPPERTITLE` = *text*	Title for the y-axis of the upper graph; default uses the identifier of the `STATISTICS` variate or pointer
`XTITLE` = *text*	Title for the x-axis; default `'Chromosomes'`
`YAXUPPER` = *scalar*	Upper bound for y-axis of the upper graph
`ANNOTATION` = *string token*	Whether to include annotation of the effects in the plot (`include`, `omit`); default `incl`

Parameters

`STATISTICS` = *variates* or *pointers*	Test statistics to be plotted; must be set
`CHROMOSOMES` = *factors*	Chromosome for each locus; must be set
`POSITIONS` = *variates*	Positions on the chromosome of each locus; must be set
`QEFFECTS` = *pointers*	QTL effects in the different environments; must be set
`QSE` = *pointers*	Standard errors of the QTL effects in the different environments; must be set
`ENVNAMES` = *texts*	Labels for the different environments; must be set
`IDEFFECTS` = *texts*	Labels to use to identify the effects
`IDPARENTS` = *texts*	Labels to use to identify the parents
`DFILENAME` = *texts*	Name of the graphics file for the plots

Description

`DQMQTLSCAN` plots QTL × E effects which can be calculated by the `QMQTLSCAN` procedure. The population type must be set by the `POPULATIONTYPE` option, and the chromosome numbers and positions on the chromosome of the loci must be specified by `CHROMOSOMES` and `POSITIONS`

parameters respectively. The plot consists of 2 parts: in the upper part the test statistic specified by parameter STATISTICS is plotted against the position of the QTL on each chromosome. If you also want to include the QTL main effects in the plot, you can set the STATISTICS parameter to a pointer with 2 variates. The second variate is then plotted, in blue, in the upper part of the screen. A horizontal red line is drawn at the threshold specified by the THRESHOLD scalar.

In the lower part of the screen, the effects of the different environments, supplied by the QEFFECTS parameter, are displayed by plotting squares of different colours for significant loci (assessed at the 0.05 level, using the standard errors supplied by the QSE parameter). There can be one, two or three plots here, according to the length of the QEFFECTS and QSE pointers. Titles for the plots can be supplied by the IDEFFECTS parameter. The first element of the QEFFECTS and QSE pointers supplies the additive effects and their standard errors. These are plotted with bluish colours for parent 1 and yellowish colours for parent 2. Their second elements can supply the dominance effects, which are plotted with bluish colours for negative effects and yellowish colours for positive effects. Their third elements can supply additive 2 effects, which are plotted with bluish colours used for parent 3 and yellowish colours for parent 4. By default, this information is appended to the title, together with labels of the parents supplied by the IDPARENTS parameter. However, you can set option ANNOTATION=omit to omit it. The brightness of the colours indicates the significance of the effects.

If the STATISTICS parameter is set to a variate (or a pointer of length 1), the METHOD option specifies the type of plot: either a line plot, a Manhattan (i.e. point) plot or spikes. The default is a line plot (and this is the only method available when the STATISTICS parameter is set to a pointer of length 2). The Manhattan plot is a point plot with a different colour for each chromosome. The colours to use for the chromosomes are specified by the COLOURS option using either a text of colour names or a variate of RGB values (see the PEN directive for details). If COLOURS is not set, the default is to use the default colours of the pens 1, 2, onwards, up to the number of chromosomes. The SUPPRESSLINES option allows you to suppress the vertical lines that are drawn between the chromosomes. Options SYMBOL, SIZEMULTIPLIER and BLACKOUTLINE are relevant only to Manhattan plots. SYMBOL can be set to a scalar containing the number of one of the pre-defined plotting symbols (see the SYMBOL parameter of PEN). The default value 2 gives a circle, 5 gives a square, 6 gives a diamond, 7 gives a triangle, and 8 gives a nabla. SIZEMULTIPLIER specifies scalar defining the multiplier to use in the calculation of the size of the symbols; default 1. BLACKOUTLINE can be set to yes to draw outlines of the symbols in black.

The DCHROMOSOMES option allows you to specify a subset of chromosomes to plot. These are identified using either the levels or the labels of the CHROMOSOMES factor. By default, DCHROMOSOMES is not set, all the chromosomes are plotted.

The TITLE option can provide an overall title for the plot in the upper graph. The YLOWERTITLE option specifies the title for the y-axis in the lower graph; default 'environments'. The YUPPERTITLE option specifies the title for the y-axis in the lower graph; the default uses the identifier of the STATISTICS variate. The ENVNAMES parameter is used to label the other lines along the y-axis of the lower graph. The upper bound of the y-axis in the upper graph can be specified by the YAXUPPER option.

By default, the plot is sent to the screen. However, you can supply a file for the plot, using the DFILENAME parameter. You can discover the types of graphics file that are supported by running the command.

```
DHELP possible
```

Options: POPULATIONTYPE, METHOD, THRESHOLD, DCHROMOSOMES, SUPPRESSLINES, SYMBOL, SIZEMULTIPLIER, BLACKOUTLINE, COLOURS, TITLE, YLOWERTITLE, YUPPERTITLE, XTITLE, YAXUPPER, ANNOTATION.

Parameters: STATISTICS, CHROMOSOMES, POSITIONS, QEFFECTS, QSE, ENVNAMES, IDEFFECTS, IDPARENTS, DFILENAME.

Action with RESTRICT
Restrictions are not allowed.

See also
Procedures: DQMAP, DQMKSCORES, DQSQTLSCAN, QMKDIAGNOSTICS.
Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

DQRECOMBINATIONS

Plots a matrix of recombination frequencies between markers (S.J. Welham & D.A. Murray).

Options

DCHROMOSOMES = *scalar*, *variate* or *text*	Specifies a subset of the linkage groups to be displayed
TITLE = *text*	General title for the plot
WINDOW = *scalar*	Window number for the graph; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
PALETTE = *string token*	Colour scheme for plot (colour, color, greyscale, grayscale); default colo

Parameters

RECFREQUENCIES = *symmetric matrices*	Recombination frequencies to plot
CHROMOSOMES = *factors*	Linkage group for each marker

Description

DQRECOMBINATIONS plots the recombination frequencies between markers in a shade diagram. The RECFREQUENCIES parameter must specify the recombination frequencies, in a symmetric matrix with each row representing a marker. The linkage groups for each marker can be supplied, in a factor, using the CHROMOSOMES parameter.

The DCHROMOSOMES option allows you to display a subset of the linkage groups. The setting can be either a variate or a scalar referring to the levels of the CHROMOSOMES factor, or a text referring to its labels.

The TITLE, WINDOW, and KEYWINDOW options can specify a title, the plotting window, and the key window. The PALETTE option controls the colour scheme used in the shade diagram.

Options: DCHROMOSOMES, TITLE, WINDOW, KEYWINDOW, PALETTE.
Parameters: RECFREQUENCIES, CHROMOSOMES.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedure: QRECOMBINATIONS.
Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

DQSQTLSCAN

Plots the results of a genome-wide scan for QTL effects in single-environment trials (M.P. Boer & J.T.N.M. Thissen).

Options

POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set when QEFFECTS are supplied
METHOD = *string token*	Method to be used for plotting (line, manhattan, spikes); default line
THRESHOLD = *scalar*	Threshold value for test statistic; default 0
DCHROMOSOMES = *scalar*, *text* or *variate*	Allows a subset chromosomes to be specified to display; default * i.e. all the chromosomes
SUPPRESSLINES = *string token*	Whether to suppress the vertical lines between the chromosomes (yes, no); default no
SYMBOL = *scalar*	Defines the plotting symbol for each point, as in the SYMBOL option of PEN, when METHOD=manhattan; default 2 i.e. circle
SIZEMULTIPLIER = *scalar*	Multiplier used in the calculation of sizes of symbols when METHOD=manhattan; default 1
BLACKOUTLINE = *string token*	Whether to draw the outer line the SYMBOL in black when METHOD=manhattan (yes, no); default no
COLOURS = *scalar*, *variate* or *text*	Colours to use for the chromosomes; default * uses the default colours of pens 1, 2 up to the number of chromosomes
TITLE = *text*	General title
YTITLE = *text*	Title for the y-axis; default uses the identifier of the STATISTICS variate or pointer
XTITLE = *text*	Title for the x-axis; default 'Chromosomes'
YUPPER = *scalar*	Upper bound for y-axis
WINDOW = *scalar*	Window number for the graphs; default 1
KEYWINDOW = *scalar*	Window number for key (zero for none); default 2
SCREEN = *string token*	Whether to clear the screen before displaying the graph (clear, keep); default clea

Parameters

STATISTICS = *variates* or *pointers*	Test statistic(s) to be plotted; must be set
CHROMOSOMES = *factors*	Chromosome for each locus; must be set
POSITIONS = *variates*	Position on the chromosome for each locus; must be set
QEFFECTS = *variates* or *pointers*	QTL effects along the genome,
QSE = *variates* or *pointers*	Standard errors of the QTL effects
IDEFFECTS = *texts*	Labels along the x-axis to identify the effects when QEFFECTS are supplied
IDPARENTS = *texts*	Labels to use to identify the parents
DFILENAME = *texts*	Name of the graphics file for the plots

Description

DQSQTLSCAN plots the results of a genome-wide QTL search which can be calculated by the QSQTLSCAN procedure. The positions on the chromosomes where a test for the presence of a QTL has been performed must be given by the POSITIONS parameter and the corresponding

chromosome number by the CHROMOSOMES parameter. The values of the test statistic must be set by STATISTICS parameter, and the threshold by THRESHOLD parameter. It is also possible to plot a second test statistic in the graph by setting the STATISTICS parameter to a pointer containing two variates.

The QEFFECTS parameter can be used to supply effects, for plotting in the x-axis margin. The population type must then also be specified by the POPULATIONTYPE option, The plot contains plotting squares of different colours for significant loci (assessed at the 0.05 level using the standard errors supplied by the QSE parameter). There can be one, two or three lines of squares, according to the length of the QEFFECTS and QSE pointers. The first element of the QEFFECTS and QSE pointers supplies the additive effects and their standard errors. These are plotted with bluish colours for parent 1 and yellowish colours for parent 2. Their second elements can supply the dominance effects, which are plotted with bluish colours for negative effects and yellowish colours for positive effects. Their third elements can supply additive 2 effects, which are plotted with bluish colours used for parent 3 and yellowish colours for parent 4. The brightness of the colour of each square indicates the significance of its effect. The KEYWINDOW option specifies the number of a window to contain a key for the plots; default 2. The IDPARENTS parameter can supply labels for the parents, to appear in the key. You can set KEYWINDOW=0 to suppress the key.

If the STATISTICS parameter is set to a variate (or a pointer of length 1), the METHOD option specifies the type of plot: either a line plot, a Manhattan (i.e. point) plot or spikes. The default is a line plot (and this is the only method available when the STATISTICS parameter is set to a pointer of length 2). The Manhattan plot is a point plot with a different colour for each chromosome. The colours to use for the chromosomes are specified by the COLOURS option using either a text of colour names or a variate of RGB values (see the PEN directive for details). If COLOURS is not set, the default is to use the default colours of the pens 1, 2, onwards, up to the number of chromosomes. The SUPPRESSLINES option allows you to suppress the vertical lines that are drawn between the chromosomes. Options SYMBOL, SIZEMULTIPLIER and BLACKOUTLINE are relevant only to Manhattan plots. SYMBOL can be set to a scalar containing the number of one of the pre-defined plotting sumbols (see the SYMBOL parameter of PEN). The default value 2 gives a circle, 5 gives a square, 6 gives a diamond, 7 gives a triangle, and 8 gives a nabla. SIZEMULTIPLIER specifies scalar defining the multiplier to use in the calculation of the size of the symbols; default 1. BLACKOUTLINE can be set to yes to draw outlines of the symbols in black.

The DCHROMOSOMES option allows you to specify a subset of chromosomes to plot. These are identified using either the levels or the labels of the CHROMOSOMES factor. By default, DCHROMOSOMES is not set, all the chromosomes are plotted.

The TITLE option can be used to provide a title for the graph. By default, the plot is sent to the screen. However, you can supply a file for the plot, using the DFILENAME parameter. You can discover the types of graphics file that are supported by running the command.

```
DHELP possible
```

The YTITLE and XTITLE options can supply titles for the y- and x-axis, respectively. If YTITLE is not specified, a default title is used, showing the identifier of the STATISTICS variate. The upper bound of the y-axis can be set by the YUPPER option. The WINDOW option specifies the window in which the graph is drawn (default 1), and the SCREEN parameter controls whether the screen is cleared before the graph is displayed.

Options: POPULATIONTYPE, METHOD, THRESHOLD, DCHROMOSOMES, SUPPRESSLINES, SYMBOL, SIZEMULTIPLIER, BLACKOUTLINE, COLOURS, TITLE, YTITLE, XTITLE, YUPPER, WINDOW, KEYWINDOW, SCREEN.

Parameters: STATISTICS, CHROMOSOMES, POSITIONS, QEFFECTS, QSE, IDEFFECTS, IDPARENTS, DFILENAME.

Action with RESTRICT
Restrictions are not allowed.

See also
Procedures: DQMAP, DQMKSCORES, DQMQTLSCAN, QMKDIAGNOSTICS.
Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

DREFERENCELINE

Adds reference lines to a graph (R.W. Payne).

Options

ORIENTATION = *string token*	Direction of the line (horizontal, vertical); default hori
WINDOW = *scalar*	Window in which to draw the line; default 1

Parameters

POSITION = *scalars*	Positions of the lines
PEN = *scalars*	Pen to use for each line
LABEL = *texts*	Text to plot alongside each line
YLPOSITION = *string tokens*	Position of the label in the y-direction (above, below, centre, center); default belo
XLPOSITION = *string tokens*	Position of the label in the x-direction (centre, center, left, right); default left
PENLABEL = *scalars*	Pen to use for each label

Description

The DREFERENCELINE procedure adds reference lines to a plot. The window containing the plot is specified by the WINDOW option. The ORIENTATION option controls whether the lines are horizontal (i.e. parallel to the x-axis) or vertical (i.e. parallel to the y-axis).

The POSITION parameter defines the position of each line, on the y-axis for a horizontal line, or the x-axis for a vertical line. The PEN parameter can specify the pen to use for the line. If this is not set, pen 255 is used as a default, having first been defined to draw continuous light grey lines in 0.75 thickness.

The LABEL parameter allows you to plot a label inside the frame, alongside the line. Its position is specified by the YLPOSITION and XLPOSITION parameters. The pen to use can be specified by the PENLABEL parameter. If this is not set, pen 256 is used as a default, having first been defined to omit any symbol and use the colour black.

Options: ORIENTATION, WINDOW.
Parameters: POSITION, PEN, LABEL, YLPOSITION, XLPOSITION, PENLABEL.

See also

Procedures: DARROW, DERRORBAR, DFRTEXT, DTEXT.
Genstat Reference Manual 1 Summary section on: Graphics.

DREPMEASURES

Plots profiles and differences of profiles for repeated measures data (J.T.N.M. Thissen).

Options

TITLE = *text*	Title for the plots; default *
GROUPS = *factors*	List of one or two factors; one factor gives one plot while a list with two factors gives as many plots as the number of levels of the first factor in the list; must be set
TIMEPOINTS = *variate* or *factor*	When the DATA parameter is set to a pointer containing a separate variate of observations for each time this can specify the actual time points (otherwise the suffixes of the DATA pointer are used), when there is a single DATA variate this must supply a factor to indicate the time of each observation
DIFFERENCES = *string token*	Can suppress plotting of the differences (no, yes); default no

Parameters

DATA = *pointers* or *variates*	Data observations either in a pointer to a list of variates (one for each time), or a single variate (with TIMEPOINTS set to a factor indicating the time of each observation)
GROUPMEANS = *tables*	To save the calculated treatment means at each timepoint

Description

A repeated measures experiment is one in which the same set of units, or subjects, is observed at a sequence of times to investigate treatment effects over a period of time.

DREPMEASURES produces high-resolution graphs showing the progress in time of a set of observations. These can be supplied in one of two ways. The first is to set the DATA parameter to a pointer containing a list of variates, each one containing the measurements made on the subjects at one of the successive occasions on which they were observed. The TIMEPOINTS option can then supply a variate to define the time point corresponding to each DATA variate; if TIMEPOINTS is unset, the suffixes of the DATA pointer are used. The second possibility is to supply set DATA to a variate containing the data from all the times. The TIMEPOINTS option must then be set to a factor indicating the time of each observation.

The groupings of the subjects should be specified by one or two factors, and input using the GROUPS option. If one factor is specified, the means of the observations at each level of the factor are plotted in one graph. If two factors are specified several graphs are produced: each graph is a plot of the means of the observations at the various levels of the second factor for a particular level of the first.

The means are calculated with the directive TABULATE. If the data variates contain missing values a warning is printed indicating the possibility of misleading results. (Before using DREPMEASURES the missing values can be estimated using the procedures ANTMVESTIMATE or MULTMISSING.)

If option DIFFERENCES=yes, two plots are produced, beside each other: one of the profiles and one of the differences with the first level. The default setting no gives the plot of the profiles only. Plots of differences can be produced only if the factor has more than one level. The TITLE option can be used to provide a title for the plots.

The calculated means can be saved by specifying parameter GROUPMEANS.

Options: TITLE, GROUPS, TIMEPOINTS, DIFFERENCES.
Parameters: DATA, GROUPMEANS.

Method
Means are calculated with the directive TABULATE. If restricted variates are specified in DATA, procedure SUBSET is used to remove any levels of the factors that are not present in the subset of subjects.

Action with RESTRICT
If DATA is set to a pointer, you can arrange to plot only a subset of the measurements by restricting any of the DATA variates or GROUPS factors. The variate specified by TIMEPOINTS for a DATA pointer must not be restricted. Similarly if DATA is set to a variate, you can restrict either the DATA variate or the GROUPS or TIMEPOINTS factors. If more than one variate or factor is restricted, they must all be restricted to the same set of units.

See also
Genstat Reference Manual 1 Summary sections on: Repeated measurements, Graphics.

DRESIDUALS

Plots residuals (R.W. Payne).

Options

RESIDUALS = *variate*	Residuals to plot
FITTEDVALUES = *variate*	Fitted values against which to plot the residuals
INDEX = *variate* or *factor*	X-variable for an index plot; default !(1,2...)
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	Overall title for the plots; default * i.e. none

Parameters

METHOD = *string tokens*	Type of residual plot (fittedvalues, normal, halfnormal, histogram, absresidual, index); default fitt, norm, half, hist
PEN = *scalars*, *variates* or *factors*	Pen(s) to use for each plot

Description

Procedure DRESIDUALS provides up to four types of plots of residuals. These are selected using the METHOD parameter, with settings: fitted for residuals versus fitted values, normal for a Normal plot, halfnormal for a half-Normal plot, histogram for a histogram of residuals, absresidual for a plot of the absolute values of the residuals versus the fitted values, and index for a plot against an "index" variable (specified by the INDEX option). The PEN parameter can specify the graphics pen or pens to use for each plot.

The residuals and fitted values must be supplied, in variates, using the RESIDUALS and FITTEDVALUES options, respectively. The TITLE option can supply an overall title for the plots. By default, high-resolution graphics are used. Line-printer graphics can be requested instead, by setting option GRAPHICS=lineprinter.

Options: RESIDUALS, FITTEDVALUES, INDEX, GRAPHICS, TITLE.
Parameters: METHOD, PEN.

Method

For a Normal plot, the Normal quantiles are calculated as follows:

$$q_i = \text{NED}((i-0.375) / (n+0.25))$$

while for a half-Normal plot they are given by

$$q_i = \text{NED}(0.5 + 0.5 \times (i-0.375) / (n+0.25))$$

Action with RESTRICT

If the variates are restricted, only the units not excluded by the restriction will be included in the graphs.

See also

Procedures APLOT, RCHECK, VPLOT.
Genstat Reference Manual 1 Summary section on: Graphics.

DRPOLYGON

Reads a polygon interactively from the current graphics device (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string token*	What to print (summary); default summ
WINDOW = *scalar*	Window from which to read default 1

Parameters

YPOLYGON = *variates*	Variates to receive the vertical coordinates of the polygons that are read
XPOLYGON = *variates*	Variates to receive the horizontal coordinates of the polygons that are read
PEN = *scalars*	Pen numbers to use to echo points

Description

DRPOLYGON uses the DREAD directive to read the coordinates of a sequence of points which define a polygon. The WINDOW option may be used to specify the window from which to read. The DREAD directive will only work within a window that contains a graph or a contour plot. A call to DRPOLYGON should, therefore, be preceded by a call to DPTMAP, DPOLYGON, DGRAPH or DCONTOUR.

DREAD is not always available, and its operation may vary slightly from one system to another. The Users' Note supplied with Genstat explains how to read points and terminate input on specific devices. The usual method for reading points is to click the left mouse button at the required position. The usual way to terminate input is to click the right mouse button. The last point of any polygon is implicitly connected to the first point. There is no need to re-enter the first point to draw a closed polygon – this will be done automatically after input has been terminated. The horizontal and vertical coordinates of the polygon may be saved using the parameters XPOLYGON and YPOLYGON, respectively.

The PEN parameter may be used to specify which pen to use to echo points which have been read. The default setting of PEN uses METHOD=line, LINESTYLE=1, SYMBOLS=1 and JOIN=given.

Printed output is controlled by the PRINT option. The default setting of summary prints the horizontal and vertical coordinates of the polygon under the headings XPOLYGON and YPOLYGON.

Options: PRINT, WINDOW.
Parameters: YPOLYGON, XPOLYGON, PEN.

Method

If the PEN parameter is unset then a pen with METHOD=line, LINESTYLE=1, SYMBOLS=1 and JOIN=given will be specified using the PEN directive. The DREAD directive is used to read in the coordinates of an open polygon, and then the DGRAPH directive is used to draw a line joining the last point of the polygon to the first point.

See also

Procedure: DPTREAD, PTREMOVE.
Genstat Reference Manual 1 Summary sections on: Graphics, Spatial statistics.

DSCATTER

Produces a scatter-plot matrix using high-resolution graphics (J. Ollerton).

Options

PEN = *scalar* or *variate* or *factor*	Pen number for the graph; default 1
EQUALSCALING = *string token*	Whether to have equal scaling of x- and y-axes in each plot (yes, no); default no
XDATA = *variates* or *factors*	Variables to be plotted as x-coordinates (DATA then specifies the y-coordinates); if unset DATA specifies both x-coordinates and y-coordinates
ASPECTRATIO = *scalar*	Ratio of the length of the y-axis to the length of the x-axis in each plot

Parameter

DATA = *variates* or *factors*	A list of variables to be plotted

Description

Procedure DSCATTER produces a scatter-plot matrix, from one or two sets of variates or factors, using high-resolution graphics.

The parameter DATA lists the variates or factors to be plotted. In the default display, each one is plotted against all the others, producing plots which are arranged as the lower triangle of a matrix with shared scales. Alternatively, if you set the XDATA to a list of variates or factors, a rectangular grid of plots is produced, displaying each DATA variable against each XDATA variable. Titles for the axes are the identifiers of the variables. The number of variates or factors that can be plotted by this procedure is in effect unlimited, but of course the greater the number of variables, the smaller the individual plots will be.

The pen to be used to plot the data can be specified with the option PEN. The EQUALSCALING option enables you to request that scaling of the x- and y-axes should be equal in each plot. By default the plots are square, but you can request rectangular plots by setting the ASPECTRATIO option to the required value for the length of the y-axis divided by the length of the x-axis.

Options: PEN, EQUALSCALING, XDATA, ASPECTRATIO.
Parameter: DATA.

Action with RESTRICT

If any of the variates or factors is restricted, only the units not excluded by the restriction will be plotted.

See also

Directive: DGRAPH.
Procedures: DMSCATTER, TRELLIS.
Genstat Reference Manual 1 Summary section on: Graphics.

DSEPARATIONPLOT

Creates a separation plot for visualising the fit of a model with a dichotomous (i.e. binary) or polytomous (i.e. multi-categorical) outcome (V.M. Cave).

Options

METHOD = *string token*	Method used to plot the predicted probabilities (rectangles, lines, rbands, lbands); default rect
PLOT = *string tokens*	Information to be plotted on the graph (key, traceline, expectednumber); default key, trac, expe when METHOD=rectangles or lines, and key when METHOD=rbands or lbands
SUCCESSLEVEL = *string token*	Specifies which level corresponds to success when GROUPS supplies a factor with 2 levels (first, second); default seco
LINEORDER = *string token*	If METHOD=lines, whether the failures or successes are plotted first (failurefirst, successfirst); default fail
NGROUPS = *scalar*	Number of discrete bands used to group the predicted probabilities when METHOD=rbands or lbands; default 10
TIES = *string token*	How tied data values in PROBABILITIES are handled when METHOD=rectangles or lines (permute, same); default perm
SEED = *scalar*	Seed for random number generator used to permute the tied data; default 0
COLOURS = *variate or text*	The two colours used to plot the predicted probabilities
THICKNESS = *scalar*	Thickness of the line for plotting the predicted probabilities when METHOD=lines or lbands; default 1
BACKGROUND = *scalar or text*	Colour of the background when METHOD=lines or lbands; default ligh
BORDER = *string token*	Whether to draw borders around the rectangles when METHOD=rectangles or rbands (yes, no); default no
USEPENS = *string token*	Whether to use the current pen definitions of pens 2 and 3 for plotting the traceline and expectednumber. respectively (yes, no); default no
SAVE = *rsave* or *pointer*	Regression or HGLM save structure to provide the data if PROBABILITIES, GROUPS, NSUCCESSES and NBINOMIAL are not specified

Parameters

PROBABILITIES = *variate or matrix*	Variate containing probabilities of success for a binary outcome (i.e. for binary or binomial data), or matrix containing probabilities of membership in each group for a polytomous outcome
GROUPS = *variate or factor*	Actual outcome, when NSUCCESSES and NBINOMIAL are not supplied
NSUCCESSES = *variate*	Number of successes when PROBABILITIES supplies predicted probabilities from binomial data
†NBINOMIAL = *variate* or *scalar*	Number of trials when PROBABILITIES supplies predicted probabilities from binomial data

TITLE = *text*	Title for the plot; default generates the title automatically
XTITLE = *text*	Title for the x-axis; default * i.e. none

Description

The DSEPARATIONPLOT procedure creates a separation plot, which is a graphical approach for assessing the fit of a model with a dichotomous (i.e. binary) or polytomous (i.e. multi-categorical) outcome. A separation plot provides a visualisation of a model's ability to predict occurrences of the event of interest (i.e. successes) with high probability, and non-occurrences (i.e. failures) with low probability. The procedure can accommodate models for binary, binomial and polytomous data.

The predicted probabilities are supplied using the PROBABILITIES parameter. For models for binary or binomial data, the predicted probabilities of success are supplied in a variate. For models for polytomous data, the predicted probabilities of membership to each group are supplied in a matrix.

The actual outcome is defined using the GROUPS parameter for binary and polytomous data, and the NSUCCESSES and NBINOMIAL parameters for binomial data. For models for binary data, GROUPS must supply either a binary variate (i.e. a variate containing only zeros or ones) or a factor with two levels. If a binary variate is supplied, one corresponds to success in relation to PROBABILITIES. Alternatively, if a factor is supplied the default is that the second level corresponds to success. You can set option SUCCESSLEVEL=first to specify that the first level corresponds to success instead.

You can use the SAVE option to supply a save structure, from a regression or an HGLM analysis, to provide the data if the PROBABILITIES, GROUPS, NSUCCESSES and NBINOMIAL parameters are not specified. The analyses must involve either a generalized linear model with a binomial distribution or an HGLM with a binomial distribution for the mean model. If neither those parameters nor SAVE are specified, the data are taken from the most recent regression analysis.

For models for polytomous data, GROUPS must supply a factor with the same number of levels as the columns in the matrix supplied by PROBABILITIES. The first level of the GROUPS factor then corresponds to the first column of the matrix, the second level to the second column, and so on (i.e. the predicted probabilities of membership to the group that correspond to the i^{th} level of the factor are in the i^{th} column of the matrix supplied by PROBABILITIES.)

For models for binomial data, NSUCCESSES must supply a variate giving the number of successes, and NBINOMIAL must supply either a scalar or a variate giving the number of trials. The GROUPS parameter is then ignored.

The predicted probabilities can be plotted as rectangles, lines or in banded groups. This is specified using the METHOD option with the following settings.

rectangles	the predicted probabilities, ordered from smallest to largest, are plotted as rectangles that are coloured according to whether or not the observation corresponds to a success (i.e. an actual occurrence of the event of interest); this is the default.
lines	this is similar rectangles, except that line segments are plotted instead of rectangles.
rbands	a separate graph is drawn for each actual outcome (i.e. success/failure for dichotomous data or each group for polytomous data) with the predicted probabilities of that outcome ordered from smallest to largest, and plotted as rectangles. The rectangles are coloured using a graduated band of colours formed by grouping the predicted

probabilities into distinct bands.

`lbands` this is similar to `rbands`, except that line segments are plotted instead of rectangles.

The `COLOURS` option defines the colours that are used to plot the predicted probabilities. It must supply two colours, either in a variate (containing two numbers defining the colours using the RGB system) or in a text (containing the names of two of Genstat's pre-defined colours; see `PEN` for details). When `METHOD=rectangles` or `lines`, the first colour corresponds to failures (i.e. non-occurrences of the event of interest) and the second to successes (i.e. occurrences of the event of interest); defaults are a shade of pink (RGB value = 12917629) and a shade of green (RBG value = 5083681). When `METHOD=rbands` or `lbands`, the two colours define the start and end colours values used by `DCOLOURS` to form a linear band of graduated colours, with the first colour corresponding the lowest probability band, and the second to the highest probability band; defaults are a pale shade of yellow (RGB value = 16777011) and a dark shade of red (RBG value = 15073280). The number of discrete bands (and therefore colours) used to group the predicted probabilities into bands is specified using the `NGROUPS` option. By default the predicted probabilities are grouped into 10 distinct bands; [0,0.1), [0.1,0.2), [0.2,0.3), [0.3,0.4), [0.4,0.5), [0.5,0.6), [0.6,0.7), [0.7,0.8), [0.8,0.9), [0.9,1]. (Note: the highest probability band is always a closed interval. All other probability bands are right half-open intervals.)

With large data sets, the lines on a separation plot may overlap. The `THICKNESS` option can be used modify the thickness of lines plotted when `METHOD=lines` or `lband`, by specifying a value by which the standard thickness is to be multiplied; default 1.

When `METHOD=lines`, the default is to plot the failures (i.e. non-occurrences) before the successes (i.e. occurrences of the event of interest). The success lines may then overlap and obscure the failure lines. Alternatively, you can set option `LINEORDER=success` to plot the successes lines first. The failures may then obscure the successes.

The `BACKGROUND` specifies the background colour when `METHOD=lines` or `lband`; default `lightgray`. Either a scalar (defining the colour using the RGB system) or a text (containing the name of a pre-defined colour; see `PEN` for details) may be supplied.

By default, borders are not drawn around the rectangles when `METHOD=rectangles` or `rbands`. However, you include borders by setting option `BORDER=yes`. Their appearance can be modified by altering the settings of pen −7 (see `PEN` for details).

With `METHOD=rectangles` or `lines`, the individual predicted probabilities are plotted in order from smallest to largest. The `TIES` option controls how tied probabilities are handled. The default, `TIES=permute`, randomly permutes the order in which the tied values are plotted, thereby breaking up any pre-existing patterns that may distort the appearance of the separation plot. Alternatively, `TIES=same` plots the tied values in the same order as they appear in `PROBABILITIES`.

The `SEED` option specifies the seed for the random-number generator, used by `RANDOMIZE`, to make the permutations when `TIES=permute`. The default of zero continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, it initializes the seed automatically. If you use the same (non-zero) seed more than once, the tied values will be permuted in the same way, and hence you will get same separation plot.

The `PLOT` option controls what additional information is plotted on the graph, with the following settings.

`key` adds a key to the graph.

`traceline` adds a line graph of the ordered predicted probabilities when `METHOD=rectangles` or `lines`.

`expectednumber` adds a symbol (default star) denoting the expected number of successes when `METHOD=rectangles` or `lines`. This is calculated as the sum of the predicted probabilities for

the occurrence of the event of interest (i.e. the sum of the predicted probabilities of success).

By default, the `key` is plotted. Also, when `METHOD=rectangles` or `lines`, the `traceline` and the `expectednumber` are plotted by default. You can suppress any additional information by setting `SHOW=*`.

You can set option `USEPENS=yes` to use the settings of pens 2 and 3 for the line drawn by `SHOW=traceline` and for the symbol added by `SHOW=expectednumber`, respectively. You can thus modify their appearance by modifying the settings of these pens prior to using `DSEPARATIONPLOT`. (See `PEN` for details.)

The `TITLE` and `XTITLE` parameters can supply an overall title and a x-axis title for the separation plot, respectively. If no overall title is supplied, a suitable title is generated automatically. To omit the title, a blank string can be supplied, i.e. `TITLE=' '`. By default, the x-axis title is not displayed.

Options: `METHOD`, `PLOT`, `SUCCESSLEVEL`, `LINEORDER`, `NGROUPS`, `TIES`, `SEED`, `COLOURS`, `THICKNESS`, `BACKGROUND`, `BORDER`, `USEPENS`, `SAVE`.
Parameters: `PROBABILITIES`, `GROUPS`, `NSUCCESSES`, `NBINOMIAL`, `TITLE`, `XTITLE`.

Method

`DSEPARATIONPLOT` uses the methods described by Greenhill *et al.* (2011).

Action with `RESTRICT`

The `DSEPARATIONPLOT` does not allow restrictions. A fault will result if any of `PROBABILITIES`, `GROUPS`, `NSUCCESSES` or `NBINOMIAL` are restricted.

References

Greenhill, B., Ward, M.B. & Sacks, A. (2011). The separation plot: a visual method for evaluating the fit of binary models. *American Journal of Political Science* **55**, 990-1002.

See also

Directive: `MODEL`
Genstat Reference Manual 1 Summary section on: Regression analysis.

DSPIDERWEB

Displays spider-web and star plots (W. van den Berg).

Options

METHOD = *string token*	Type of plot (spiderweb, star); default spid
MARKS = *scalar* or *variate*	Distances between the strands of the web or marks on the axes of the star (scalar), or positions of those strands or marks (variate); default 0.25
ANGLE = *scalar*	Angle to rotate the plot, in degrees; default 0
SIZEMULTIPLIER = *scalar*	Controls the size of the labels identifying the categories; default selects a size appropriate to the number of plots in the frame
FRAMESHAPE = *string token*	Shape of the plotting frame (landscape, portrait, square); default squa

Parameters

DATA = *tables*	Values to plot in each frame
CATEGORIES = *factors*	Factor specifying the categories that define the axes in the plots
GROUPS = *factors* or *pointers*	Factor specifying the groups to appear in each plot
TRELLISGROUPS = *factors* or *pointers*	Factor or factors specifying the different plots of a trellis plot of a multi-way table
PAGEGROUPS = *factors* or *pointers*	Factor or factors specifying plots to be displayed on different pages
TITLE = *texts*	Title for the graph; default none
COLOURS = *texts* or *variates*	Colours to be used for the groups

Description

A spider-web or a star plot can be used to display information on several categories observed on an individual. The categories are represented as equally-spaced radii from a common origin. The observations for the individual along each radius are joined by straight lines to form a polygon, whose shape can then be used to identify individuals with similar characteristics. The spider-web plot differs from the star plot, in that it also contains reference lines joining the radii (much like a real spider's web). In a star plot, there are still markings on the radii, but these are not joined together. Spider-web plots are produced by default, but you can set option METHOD=star to produce star plots instead.

The observations must be supplied in a table, specified by the DATA parameter. One of the factors classifying the table identifies the categories. The CATEGORIES parameter must specify which of the factors classifying the table identifies the categories.

You can use the GROUPS parameter to plot information from more than one individual in the same plot. This can specify either a factor, or a pointer containing several factors, each of whose levels (or combination of levels) provides data from one of the individuals. These factors must again be among the factors classifying the DATA table. Similarly the TRELLISGROUPS and PAGEGROUPS parameters allow you to define different plots to appear in a trellis arrangement or on different pages. At least one of GROUPS, TRELLISGROUPS and PAGEGROUPS must be set.

The TITLE parameter can supply a title for the plots, and the COLOURS parameter can supply either a text or a variate to define the colours to be used for the groups.

The MARKS option can supply either a scalar to define the distance between each pair of marks or pair of reference lines on the radii, or a variate to define their actual positions. The values must be between zero and one, as the DATA values are standardized to that range for the plot. The

ANGLE option can specify an angle by which the plots are to be rotated (default 0).

The SIZEMULTIPLIER option allows you to control the sizes of the labels that are plotted at the ends of the radii to identify the categories. By default, the size is set automatically, according to the number of plots in the frame.

The FRAMESHAPE option specifies the shape of the frame, with settings:

landscape	for a frame of size 1.4 × 1.0 i.e. wider in the x- than the y-direction,
portrait	for a frame of size 1.0 × 1.4 i.e. wider in the y- than the x-direction,
square	for a frame of size 1.0 × 1.0.

Some graphics devices do not support the use of device cordinates greater than 1.0, so the default is FRAMESHAPE=square. (See FRAME and DEVICE for more information.) There must be no more than 36 plots in a square frame. Landscape and portrait frames can hold up to 48.

Options: METHOD, MARKS, ANGLE, SIZEMULTIPLIER, FRAMESHAPE.
Parameters: DATA, CATEGORIES, GROUPS, TRELLISGROUPS, PAGEGROUPS, TITLE, COLOURS.

See also
Directive: DHISTOGRAM.
Procedure: DTABLE.
Genstat Reference Manual 1 Summary section on: Graphics.

DSTTEST

Plots power and significance for t-tests, including equivalence tests (R.W. Payne).

Options

NSAMPLES = *scalar*	Number of samples for the t-test (1 or 2); default 2
PROBABILITY = *scalar*	Significance level at which the response is to be tested; default 0.05
TMETHOD = *string token*	Type of test to be done (onesided, twosided, equivalence, noninferiority); default ones
RATIOREPLICATION = *scalar*	Ratio of replication sample2:sample1 (i.e. the size of sample 2 should be RATIOREPLICATION times the size of sample 1); default 1

Parameters

RESPONSE = *scalars*	Response to be detected
VAR1 = *scalars*	Anticipated variance of sample 1
VAR2 = *scalars*	Anticipated variance of sample 2; default * assumes the same variance as sample 1
NREPLICATES = *scalars*	Number of replicates
RDF = *scalars*	Number of residual degrees of freedom; default * calculates these automatically, assuming a standard t-test

Description

DSTTEST produces a plot showing the probability distributions for the null and alternative hypotheses for various types of t-test. This is a companion procedure to STTEST, which calculates sample sizes for t-tests. The area of the distribution for the null hypothesis, in the critical region (where the null hypothesis would be rejected), is coloured in red. Its size corresponds to the significance level of the t-test, which is set by the PROBABILITY option (default 0.05). The area of the distribution for the alternative hypothesis in the critical region is coloured in dark blue, unless it overlaps the red colour of the null hypothesis. The size of the dark blue area (including that overlapped by red) corresponds to the power of the test. The area of the distribution for the alternative hypothesis in the non-critical region (where the null hypothesis would still be accepted) is coloured in light blue.

The plots can be done for either a one-sample t-test (testing for evidence that the mean of the sample differs from a specific value), or a two-sample test (testing that means of the samples are different). The number of samples is specified by the NSAMPLES option (default 2). The size of response to be detected is supplied by the RESPONSE parameter. (This is difference between the sample mean of a one-sample test and the specific value, or the difference between the means of the two samples in a two-sample test.) The VAR1 parameter supplies the variance of the observations in the sample of a one-sample test or of the first sample of a two-sample test. If the second sample of a two-sample test has a different variance from the first sample, this can be supplied by the VAR2 parameter.

The NREPLICATES parameter specifies the size of the first sample. By default, it is assumed that the sizes of the samples in the two-sample test are equal. However, you can set the RATIOREPLICATION option to a scalar, R say, to indicate that the size of the second sample is R times the size of the first sample.

By default, DSTTEST assumes a one-sided t-test is to be used, but you can set option TMETHOD=twosided to take a two-sided t-test instead. Other settings of TMETHOD enable you to test for equivalence or for non-inferiority. To demonstrate equivalence of the two samples (TMETHOD=equivalence), their means m_1 and m_2 must differ by less than some threshold d; this is specified by RESPONSE and should represent a limit below which the difference can be

assumed to have no physical (or clinical) importance. Statistically, equivalence implies comparing a null hypothesis that the samples are not equivalent, i.e.

$$(m_1 - m_2) \leq -d$$

or

$$(m_1 - m_2) \geq d$$

with the alternative hypothesis that they are equivalent, i.e.

$$-d < (m_1 - m_2) < d$$

A one-sample test for equivalence operates similarly, but here d specifies the threshold for the sample mean itself. To demonstrate non-inferiority of sample 1 compared to sample 2, the null hypothesis becomes

$$(m_1 - m_2) \geq -d$$

(which, in fact, represents a simple one-sided t-test). See `STTEST` for further details.

Options: `NSAMPLES`, `PROBABILITY`, `TMETHOD`, `RATIOREPLICATION`.
Parameters: `RESPONSE`, `VAR1`, `VAR2`, `NREPLICATES`, `RDF`.

See also
Procedure: `STTEST`, `TTEST`.
Genstat Reference Manual 1 Summary section on: Design of experiments.

DTABLE

Plots tables (R.W. Payne).

Options

GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
METHOD = *string token*	What to plot (points, linesandpoints, onlylines, data, barchart, splines); default poin
XFREPRESENTATION = *string token*	How to label the *x*-axis (levels, labels); default labels uses the XFACTOR labels, if available
DFSPLINE = *scalar*	Number of degrees of freedom to use when METHOD=splines
YTRANSFORM = *string tokens*	Transformed scale for additional axis marks and labels to be plotted on the right-hand side of the y-axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden i.e. none
PENYTRANSFORM = *scalar*	Pen to use to plot the transformed axis marks and labels; default * selects a pen, and defines its properties, automatically
†KEYMETHOD = *string token*	What to use for the key descriptions when GROUPS specifies more than one factor (labels, namesandlabels); default name
†PLOTTITLEMETHOD = *string token*	What to use for the titles of the plots when TRELLISGROUPS specifies more than one factor (labels, namesandlabels); default name
†PAGETITLEMETHOD = *string token*	What to use for the titles of the pages when PAGEGROUPS specifies more than one factor (labels, namesandlabels); default name
†USEAXES = *string token*	Which aspects of the current axis definitions of window 1 to use (none, limits, marks, mpositions, nsubticks,); default none

Parameters

TABLE = *tables*	Tables to plot
DATA = *variates*	Data values to plot with each table when METHOD=data
XFACTOR = *factors*	Factor providing the *x*-values for the plot of each table
GROUPS = *factors* or *pointers*	Factor or factors identifying the different lines from a multi-way table
TRELLISGROUPS = *factors* or *pointers*	Factor or factors specifying the different plots of a trellis plot of a multi-way table
PAGEGROUPS = *factors* or *pointers*	Factor or factors specifying plots to be displayed on different pages
BAR = *scalars*, *tables* or *pointers*	Scalar defining the length of error bar to be plotted to indicate the overall (or average) variability of the values in each table, or table defining the variability of each individual table value, or pointer containing either two scalars or two tables defining the upper and lower positions of the error bar(s)

NEWXLEVELS = *variates*	Values to be used for XFACTOR instead of its existing levels
TITLE = *texts*	Title for the graph; default uses the identifier of the TABLE
YTITLE = *texts*	Title for the y-axis; default ' '
XTITLE = *texts*	Title for the x-axis; default is to use the identifier of the XFACTOR
BARDESCRIPTION = *texts*	Descriptions for the bars
PENS = *variates*	Defines the pen to use to plot the points and/or line for each group defined by the GROUPS factors

Description

DTABLE plots the tables specified by the TABLE parameter (each table displayed in a separate set of plots). The GRAPHICS option controls whether a high-resolution or a line-printer graph is plotted; by default, GRAPHICS=high.

The METHOD option controls how each table is plotted in high-resolution graphics, with settings:

points	to plot points at the table values;
linesandpoints	to plot points and join them by lines;
onlylines	to draw lines between the table values;
data	to draw lines between the table values, and then also plot the data values supplied (in a variate) by the DATA parameter;
barchart	to plot the table values as a barchart;
splines	to plot the points together with a smooth spline to show the trend over each group of points; the DFSPLINE specifies the degrees of freedom for the splines; if this is not set, 2 d.f. are used when there are up to 10 points, 3 if there are 11 to 20, and 4 for 21 or more.

By default METHOD=points (and this is the only display available in line-printer graphics).

The XFACTOR parameter defines the factor against whose levels the values of the table are plotted. With a multi-way table, there will be a plot of the table values against the XFACTOR levels for every combination of levels of the other factors classifying the table. The GROUPS parameter specifies factors whose levels are to be included in a single window of the graph. So, for example, if you specify

```
DTABLE [METHOD=line] Table; XFACTOR=A; GROUPS=B
```

DTABLE will plot the values of Table in a single window with factor A on the x-axis, and a line for each level of the factor B. You can set GROUPS to a pointer to specify several factors to define groups. For example

```
POINTER [VALUES=B,C] Groupfactors
DTABLE [METHOD=line] Table; XFACTOR=A; GROUPS=Groupfactors
```

to plot a line for every combination of the levels of factors B and C. Similarly, the TRELLISGROUPS option can specify one or more factors to define a trellis plot. For example,

```
DTABLE [METHOD=line] Table; XFACTOR=A; GROUPS=B;\
       TRELLISGROUPS=C
```

will produce a plot for each level of C, in a trellis arrangement; each plot will again have factor A on the x-axis, and a line for each level of the factor B. Likewise, the PAGEGROUPS parameter can specify factors whose combinations of levels are to be plotted on different pages. So

```
DTABLE [METHOD=line] Table; XFACTOR=A; GROUPS=B; PAGEGROUPS=C
```

will again produce a plot for each level of C, but now on separate pages.

If XFACTOR is unset, DTABLE will select the XFACTOR according to the following criteria (in decreasing order of importance): that the factor has no labels, that it has levels that are not the default integers 1 upwards, or that it has more levels than the other factors. If GROUPS is unset, it will be set to all the factors except the XFACTOR. (So, if you want to use either TRELLISGROUPS or PAGEGROUPS, you must also specify XFACTOR and GROUPS.)

The BAR parameter can be set to a scalar to specify an overall (or average) error bar, such as a standard error for differences between any pair of table values. Alternatively, it can be set to a table to specify a different error value, such as an effective standard error, for every table value; DTABLE then plots a bar of the defined size above and below each table value. Finally, it can be set to a pointer containing either two scalars or two tables, specifying the upper and lower positions of the error bar(s). Note, however, that the table setting may be unsuitable for plots other than barcharts when there are GROUPS, as the error bars may overlap each other.

The NEWXLEVELS parameter enables different levels to be supplied for XFACTOR if the existing levels are unsuitable. If XFACTOR has labels, these are used to label the x-axis unless you set option XFREPRESENTATION=levels.

The TITLE, YTITLE and XTITLE parameters can supply titles for the graph, the y-axis and the x-axis, respectively. The symbols, colours and line styles that are used in a high-resolution plot are usually set up by DTABLE automatically. If you want to control these yourself, you should use the PEN directive to define a pen with your preferred symbol, colour and line style, for each of the groups defined by combinations of the GROUPS factors. The pen numbers should then be supplied to DTABLE, in a variate with a value for each group, using the PENS parameter.

The YTRANSFORM option allows you to include additional axis markings, transformed onto another scale, on the right-hand side of the y-axis. Suppose, for example, the table contains means from an analysis of a variate of percentages that had been transformed to logits. You might then set YTRANSFORM=ilogit (the inverse-logit transformation) to include markings in percentages alongside the logits. The settings are the same as those of the TRANSFORM parameter of AXIS (which is used to add the markings). You can control the colours of the transformed marks and labels, by defining a pen with the required properties, and specifying it with the PENYTRANSFORM option. Otherwise, the default is to plot them in blue.

When there is more than one GROUPS factor, the KEYMETHOD controls whether to use the factor names with their labels (or levels for factors with no labels) or just the labels (or levels) in the key descriptions. The default is to use the names and the labels (or levels). Similarly, the PLOTTITLEMETHOD specifies what to use for the titles of the plots when there is more then one TRELLISGROUPS factor, and the PAGETITLEMETHOD specifies what to use for the titles of the plots when there is more then one PAGEGROUPS factor. You can setKEYMETHOD=* to have no key at all.

The USEAXES option allows you to control various aspects of the axes. First you need to use the XAXIS and YAXIS directives to define them for window 1. Then specify which of the aspects of the axes in window 1 are to be used by DTABLE, by specifying USEAXES with the following settings:

limits	y- and x-axis limits (LOWER and UPPER parameters);
marks	location and labelling of the tick marks (MARKS, LABELS, LDIRECTION, LROTATION, DECIMALS, DREPRESENTATION, and VREPRESENTATION parameters);
mpositions	positions of the tick marks (MPOSITION parameter); and
nsubticks	number of subticks per interval (NSUBTICKS parameter).

By default none are used.

Options: GRAPHICS, METHOD, XFREPRESENTATION, DFSPLINE, YTRANSFORM, PENYTRANSFORM, KEYMETHOD, PLOTTITLEMETHOD, PAGETITLEMETHOD, USEAXES.

Parameters: TABLE, DATA, XFACTOR, GROUPS, TRELLISGROUPS, PAGEGROUPS, BAR,

NEWXLEVELS, TITLE, YTITLE, XTITLE, BARDESCRIPTION, PENS.

See also
Directives: TABLE, TABULATE, PREDICT, VPREDICT.
Procedures: AGRAPH, AUGRAPH, MTABULATE, SVTABULATE, VGRAPH.
Genstat Reference Manual 1 Summary section on: Graphics.

DTEXT

Adds text to a graph (S.A. Harding).

Option

WINDOW = *scalar*	Window number of the graph; default 1

Parameters

Y = *variates* or *scalars*	Vertical coordinates
X = *variates* or *scalars*	Horizontal coordinates
TEXT = *texts*	Text to plot
PEN = *scalars*, *variates* or *factors*	Pens to use; default 1

Description

The DTEXT procedure provides a convenient way of adding textual annotation or description to a plot. The text to plot is specified by the TEXT parameter. This can be either a single string, or a Genstat text structure containing several lines of text. The Y and X parameters specify where to plot the text, with scalars for a single string or line, or with variates for several lines. The PEN parameter specifies the pen or pens to use (default 1), and the WINDOW option specifies the window where the plot is taking place (default 1).

Option: WINDOW.
Parameters: Y, X, TEXT, PEN.

Action with RESTRICT

DTEXT takes account of restrictions on any set of Y, X and TEXT parameters.

See also

Procedures: DFRTEXT, DARROW, DERRORBAR, DREFERENCELINE.
Genstat Reference Manual 1 Summary section on: Graphics.

DTIMEPLOT

Produces horizontal bars displaying a continuous time record (S.J. Clark).

Options

`TITLE` = *text*	Title for the plot; default `*` i.e. none
`WINDOW` = *numbers*	Which high-resolution graphics windows to use; default 3 for single plots and 5...8 for the composite plot
`SCREEN` = *string token*	Whether to clear the graphics screen before plotting (`clear`, `keep`); default `clea`

Parameters

`DATA` = *variates*	Bout lengths
`GROUPS` = *factors*	Factor defining act performed during each bout
`LABELS` = *texts*	Labels for each act
`METHOD` = *texts*	Type of plot to produce for each `DATA` variate (`barplot`, `cumulative`, `log`, `survivor`, `composite`); default `comp`

Description

`DTIMEPLOT` produces graphical displays from a continuous time record of behaviour. A variate of bout lengths (see below) is specified by the `DATA` parameter and the `GROUPS` parameter specifies a factor defining the act (see below) performed during the bout. The type of plot to be produced is specified by the `METHOD` parameter – only one setting is allowed giving either a bar plot (`barplot`), cumulative bout length plot (`cumulative`), log bout length plot (`log`), log-survivor plot (`survivor`) or a composite plot of all four (`composite`). By default a composite plot is produced. The parameter `LABELS` can be used to specify labels to distinguish each act on the vertical axis of the barplot (group 1 appears at the lower end of the axis) – this parameter is ignored for `METHOD` settings `cumulative`, `log` and `survivor`. Parameters `DATA` and `GROUPS` must be set; they must have the same number of values and must not contain any missing values.

The graphical display can be controlled as usual using the `TITLE`, `WINDOW` and `SCREEN` options. By default single plots are produced in frame 3 and the composite plot in frames 5, 6, 7 and 8, plots have no title and are drawn on a new screen.

Options: `TITLE`, `WINDOW`, `SCREEN`.
Parameters: `DATA`, `GROUPS`, `LABELS`, `METHOD`.

Method

The bar plot, cumulative bout length plot, log bout length plot and log-survivor plot are graphical displays of a continuous time record of behaviour (Haccou & Meelis 1992). The behaviour of an individual during a period of observation is classified into a mutually exclusive and exhaustive set of behaviours (acts). A bout is a time interval during which a certain act is performed. A bout length is the duration of such a time interval. The period of observation is assumed to begin at time zero. In the bar plot the acts are arranged along the vertical axis and the observation time is given on the horizontal axis; occurrences of each act are represented by solid bars. The cumulative bout length plot, in which the cumulative bout lengths for an act are plotted against the sequence numbers of the bouts, is a useful display for detecting changes in mean bout length (which are indicated by a sudden and consistent change in the slope of the line). In a log bout length plot the logged bout lengths are plotted against the start times of the bouts. This plot provides a check for exponentiality (the vertical width of scattering should be approximately constant), and provides evidence for sudden changes in termination rate. The empirical log-survivor plot should approximate to a straight line if the bout lengths for an act are a sample from

an exponential distribution.

Action with `RESTRICT`
Neither the input variate nor the input factor must be restricted.

Reference
Haccou, P. & Meelis, E. (1992). *Statistical Analysis of Behavioural Data*. Oxford University Press, Oxford.

See also
Genstat Reference Manual 1 Summary sections on: Graphics, Repeated measurements, Survival analysis.

DVARIOGRAM

Plots fitted models to an experimental variogram (S.A. Harding, D.A. Murray & R. Webster).

Options

MODELTYPE = *string token*	Defines which model to plot (power, boundedlinear, circular, spherical, doublespherical, pentaspherical, exponential, besselk1, gaussian, affinepower, linear, cubic, stable, cardinalsine, matern); default powe
ISOTROPY = *string token*	Defines whether this is an isotropic or geometrical anisotropic model (isotropic, geometrical); default isot
WINDOW = *scalar*	Window in which to plot a graph; default 1
TITLE = *text*	Title for the graph

Parameters

VARIOGRAM = *variates*	Experimental variogram to which the model or matrices has been fitted, as a variate if in only one direction or as a matrix if there are several
DISTANCE = *variates*	Mean lag distances for the points in each or matrices variogram
DIRECTION = *variates*	Directions in which each variogram was computed
ESTIMATES = *variates*	Estimated parameter values
XUPPER = *scalar*	Upper limit for the x-axis in the graph
PENDATA = *scalar*	Pen to be used to plot the data; default 1
PENMODEL = *scalar*	Pen to be used to plot the model; default 2

Description

DVARIOGRAM plots fitted models to an experimental variogram using estimates produced by MVARIOGRAM.

The data for the procedure can be taken directly from the FVARIOGRAM directive and MVARIOGRAM procedure. The parameters DISTANCES and VARIOGRAMS correspond to those with the same names in FVARIOGRAM. The data will be in variates if the variogram was calculated in only one direction. If it is in several, they can either be in matrices (as generated by FVARIOGRAM) or in variates. For a geometrical anisotropic model, directions must be supplied using the DIRECTIONS parameter. These should be in a variate with one value for each column if the other data are in matrices; alternatively, they should be in a variate of the same length as the other variates.

The MODELTYPE and ISOTROPY options specify the fitted model that is to be plotted, exactly as in the MVARIOGRAM procedure. The estimates for the model parameters are supplied in a variate using the ESTIMATES parameter. These can be taken directly from MVARIOGRAM using the ESTIMATES parameter. The number of values within the variate for the estimates will depend on the model that has been fitted. See MVARIOGRAM for details.

The placement of the graph within the graphical frame can be controlled using the WINDOW option. The TITLE option can supply a title for the plot. Option XUPPER can define an upper value for the x-axis (i.e. distance), and PENDATA and PENMODEL can supply the numbers of the pens to be used to plot the experimental variogram and the fitted model respectively (by default 1 and 2).

Options: MODELTYPE, ISOTROPY, WINDOW, TITLE.
Parameters: VARIOGRAM, DISTANCE, DIRECTION, ESTIMATES, XUPPER, PENDATA, PENMODEL.

Action with `RESTRICT`
If the data variates are restricted, only the units not excluded by the restriction will be plotted.

References
Chiles, J-P. & Delfiner, P. (1999). *Geostatistics: Modeling Spatial Uncertainty*. Wiley, Chichester.
Webster, R. & Oliver, M.A. (2001). *Geostatistics for Environmental Scientists*. Wiley, Chichester.
Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, **41**, 434-449.

See also
Directives: `FVARIOGRAM`, `KRIGE`.
Procedures: `MVARIOGRAM`, `DCOVARIOGRAM`.
Genstat Reference Manual 1 Summary sections on: Spatial statistics, Graphics.

DXDENSITY

Produces one-dimensional density (or violin) plots (D. B. Baird).

Options

BANDWIDTH = *scalar*	Bandwidth for kernel smoothing (0-1); default density is chosen according to the number of observations
GAP = *scalar*	The size of the gap (0-1) between envelopes when there are several densities; default 0.1
TRANSFORM = *string token*	Transformed scale for the data (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default is to use the transform defined for XAXIS
AXISTITLE = *text*	The title for the data axis; default is the name of the DATA variate
GROUPSTITLE = *text*	The title for the groups or variates axis; default is to use the name of the GROUPS factor
WINDOW = *scalar*	Window number for the graph; default 3
ORIENTATION = *string token*	Orientation of plots (horizontal, vertical); default vert
METHOD = *string token*	Method for plotting the density envelope (fill, line); default fill
SCREEN = *string token*	Whether to clear screen before the plot (clear, keep, resize); default clea

Parameters

DATA = *variates* or *pointer*s	The data whose density is to be plotted
GROUPS = *factor*s	Factor to divide values of a single variate into groups; default * i.e. none
TITLE = *texts*	Title for graph; default uses the names of the data variates and type of plot

Description

DXDENSITY produces density (or *violin*) plots, using high-resolution graphics. The data are specified by the DATA parameter. For a single density plot, DATA should be set to a variate. To plot several densities, you can set DATA to a pointer of variates. Alternatively, it can be set to a single variate, with the GROUPS parameter then specifying a factor to identify groups of points whose densities are to be plotted separately.

The points are plotted along a line, with a kernel density smooth on either side to indicate the density of points along the line. The BANDWIDTH option specifies the band width for the kernel smoothing; larger values make it smoother, and smaller values allow it to be rougher. The default is chosen automatically, according to the number of observations. The gap between the envelopes for different variates or groups can be defined by the GAP option; this must be between 0 and 1 (default 0.1).

The METHOD option controls how the density envelope is drawn around the points, either as a filled region or as a line. You can use the TITLE parameter to supply an overall title for the plot. The AXISTITLE and GROUPSTITLE options can be used to supply titles for the data and groups axes respectively. The WINDOW option specifies the number of the window to use for the plot (default 3), and the SCREEN option controls whether the screen is cleared first, as usual (see DGRAPH).

The data can be transformed by using the TRANSFORM option. If this is not set, DXDENSITY uses the setting of TRANSFORM, defined by the XAXIS directive for the specified WINDOW. The

available settings are the same as those of the TRANSFORM option of XAXIS.

The ORIENTATION option controls whether the data axis is horizontal or vertical (default). The XAXIS and YAXIS directives can be used, prior to using DXDENSITY, to set attributes of the axes of the plot in the window that is to be used. Note that, when the orientation is horizontal, the attributes of the x- and y-axes are swapped, so that the x-axis settings are always applied to the data axis.

Options: BANDWIDTH, GAP, TRANSFORM, AXISTITLE, GROUPSTITLE, WINDOW, ORIENTATION, METHOD, SCREEN.
Parameters: DATA, GROUPS, TITLE.

Action with RESTRICT
If any of the variates or factors are restricted, only the units not excluded by the restriction will be plotted.

See also
Directive: DHISTOGRAM.
Procedures: DXYDENSITY, DOTHISTOGRAM, BOXPLOT, KERNELDENSITY, RUGPLOT.
Genstat Reference Manual 1 Summary section on: Graphics.

DXYDENSITY

Produces density plots for large data sets (D. B. Baird).

Options

PLOT = *string tokens*	How to plot the density (pointplot, shadeplot, contourplot, histogram, surface); default poin
NGROUPS = *scalar*	Number of sections into which to divide each axis (4-400); default 50
METHOD = *string token*	Method to use to smooth the density (thinplate, radialspline, tensorspline, kernel); default * i.e. none
DF = *scalar*	Degrees of freedom for smoothing methods (2-50); default 12
BANDWIDTH = *scalar*	Bandwidth for kernel smoothing (0-1); default 0.2
MEANFIT = *string tokens*	What smooth regression fits to the means to plot (yx, xy); default * i.e. none
NCONTOURS = *scalar*	Number of contours in the contour plot; default 9
SYMBOL = *string token*	Symbol to use in a point plot (circle, square); default circ
COLOURS = *text*, *variate* or *scalar*	Colour to use to draw the symbols, shades, contours or surface; default !t(red, blue, black)
XTRANSFORM = *string token*	Transformed scale for the x-axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden
YTRANSFORM = *string token*	Transformed scale for the y-axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden
ZTRANSFORM = *string token*	Transformed scale for the z-axis (identity, percentile, root); default iden
WINDOW = *scalar*	Window number for the graphs; default 3
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep, resize); default clea

Parameters

Y = *variate* or *factor*	Y-coordinates of the data
X = *variate* or *factor*	X-coordinates of the data
TITLE = *text*	Title for graph; default uses the names of the data and type of plot

Description

Procedure DXYDENSITY produces a density plot of two variables, using high-resolution graphics. A density plot provides a better visual representation of the 2-dimensional spread of points than a scatter plot if there are a large number of points or many points overlap each other, and is quicker to plot. A density plot displays the number of points in small regions of the x-y plane, using various methods to plot the density.

The x and y axes are divided into equally spaced sections, to give a grid of rectangular cells covering the x-y plane. The density is calculated as the number of points that falls into each cell. The number of sections is specified by the NGROUPS option, as a scalar if the same number is required in each direction, or as a variate with two values to specify different numbers for the

y-axis (first value) and the x-axis (second value). Having a large number of cells preserves more detail, but increases the time required to create and plot the graph.

The x- or y-axes can be transformed before forming the sections and calculating the density, by using the XTRANSFORM or YTRANSFORM options. The settings are the same as those of the TRANSFORM option of the XAXIS and YAXIS directives.

The PLOT option controls how the density is plotted, with settings:

pointplot	point plot , using the symbol size to indicate the number of points in each cell;
shadeplot	shade plot, using intensity of colour to indicate the number of points in each cell;
contourplot	contour plot, with contours showing the density;
surface	surface plot, with density as height;
histogram	3-dimensional histogram of the density.

By default PLOT=pointplot.

The density can be smoothed by using the METHOD option, with settings:

thinplate	a 2-dimensional thin plate spline is fitted to the counts using the THINPLATE procedure;
radialspline	a 2-dimensional radial spline is fitted to the counts using the RADIALSPLINE procedure;
tensorspline	a 2-dimensional tensor spline is fitted to the counts using the TENSORSPLINE procedure;
kernel	a 2-dimensional kernel smoother is fitted to the counts.

By default no smoothing is done.

The DF option specifies the number of degrees of freedom for the splines (default 12); smaller values make the surface smoother, and larger values allow it to be rougher. The BANDWIDTH option specifies the band width for kernel smoothing; larger values make the surface smoother, and smaller values allow it to be rougher.

The shape of each point in a point plot is specified by the SYMBOL option, as either a circle (default) or square. The COLOURS option specifies the colours that are used, in a scalar or a text or variate with up to three values. For a line plot, the first value specifies the colour for the points, and the second and third values define the colours for any lines fitted by the MEANFIT option. For a histogram, the first value of COLOURS defines the colour of the bars. For shade, contour and surface plot, if COLOURS has two or more values, the first is used for high densities, the second is used for low densities, and intermediate densities are plotted in the corresponding intermediate colour; if COLOURS has only one value, the low densities are plotted in white. If COLOURS has three values, the third is used for the contours of contour and surface plots.

The scaling of densities is controlled by the ZTRANSFORM option with settings:

identity	no scaling (default),
root	takes the square root of the densities, giving more emphasis to low counts,
percentile	takes a rank transform and plots these, so that percentiles are equally spaced.

The MEANFIT option allows you can to add a smoothing spline regression of y on x or of x on y to a point plot. The available settings are

yx	for a regression of y on x, and
xy	for a regression of x on y.

The DF option again specifies the number of degrees of freedom for the spline (default 12). By default neither are done.

The Y and X parameters specify the y- and x-coordinates of the data values, in either variates or factors. Their identifiers are used for the titles of the axes at the lower and left-hand edges of the graphics frame (i.e. page). You can also use the TITLE parameter to supply an overall title

for the plot.

The WINDOW options specifies the number of the window to use for the plot, and the SCREEN option controls whether the screen is cleared first, as usual (see e.g. DGRAPH).

Options: PLOT, NGROUPS, XTRANSFORM, YTRANSFORM, ZTRANSFORM, METHOD, MEANFIT, DF, BANDWIDTH, NCONTOURS, COLOURS, SYMBOL, WINDOW, SCREEN.
Parameters: Y, X, TITLE.

Action with RESTRICT

If any of the variates or factors are restricted, only the units not excluded by the restriction will be plotted.

See also

Directive: DCONTOUR, DGRAPH, DSHADE, D3GRAPH.
Genstat Reference Manual 1 Summary section on: Graphics.

DXYGRAPH

Draws two-dimensional graphs with marginal distribution plots alongside the y- and x-axes (D.A. Murray).

Options

YMETHOD = *string token*	Distribution plot to display in the margin of the y-axis (histogram, rugplot, boxplot); default hist
XMETHOD = *string token*	Distribution plot to display in the margin of the x-axis (histogram, rugplot, boxplot); default hist
YNGROUPS = *scalar*	Defines the number of groups in a margin plot of a histogram of the Y variate; default is then 10, or the integer value nearest the square root of the number of values in the Y variate if that is smaller
XNGROUPS = *scalar*	Defines the number of groups in a margin plot of a histogram of the X variate; default is then 10, or the integer value nearest the square root of the number of values in the X variate if that is smaller
YCOLOUR = *scalar* or *text*	Colour to use for the Y margin plot
XCOLOUR = *scalar* or *text*	Colour to use for the X margin plot

Parameters

Y = *variates* or *factors*	Vertical coordinates
X = *variates* or *factors*	Horizontal coordinates
TITLE = *texts*	General title for the plot; default *
WINDOW = *scalars*	Window number for the graphs; default 1
KEYWINDOW = *scalars*	Window number for the key (zero for no key); default 2
PEN = *scalars*, *variates* or *factors*	Pen number for each graph; default * uses pens 1, 2, and so on for the successive graphs
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea

Description

The DXYGRAPH procedure draws high-resolution two-way plots with a distribution plot alongside the y- and the x-axis. The main part of the graph is an ordinary two-dimensional graph (e.g. a point or line plot), which is plotted by the DGRAPH directive in the usual way. The Y and X parameters supply the y- and x-coordinates of the items to be plotted, exactly as in DGRAPH, and the PEN parameter can specify graphics pens to define how the plotting is done. See DGRAPH for full details.

The YMETHOD option specifies the type of distribution plot to be displayed alongside the y-axis. By default this is a histogram. Alternatively, you can set YMETHOD=rugplot to produce a rug plot, or YMETHOD=boxplot to display a schematic boxplot. Similarly, the XMETHOD option controls the distribution plot displayed below the x-axis. By default the number of groups used to draw a marginal plot of a histogram is either 10, or the integer value nearest the square root of the number of values in the associated variate if that is smaller. Alternatively, you can specify the number of groups using the YNGROUPS and XNGROUPS for the Y and X marginal plots respectively. The YCOLOUR and XCOLOUR options can be used to specify the colours to be used for the margin plots.

The WINDOW parameter defines the window in which the graph is drawn (default 1), and the KEYWINDOW parameter specifies the window in which the key appears (default 2). You can set

KEYWINDOW=0 to suppress the key. The TITLE parameter can be used to provide a title for the graph, and the SCREEN parameter controls whether the graphical display is cleared before the graph is plotted.

Options: YMETHOD, XMETHOD, YNGROUPS, XNGROUPS, YCOLOUR, XCOLOUR.
Parameters: Y, X, TITLE, WINDOW, KEYWINDOW, PEN, SCREEN.

Action with RESTRICT
You can arrange to plot only a subset of the points specified by a particular pair of Y and X vectors and associated PEN vector, by restricting any one of them. If more than one of these is restricted, then they must all be restricted in exactly the same way.

See also
Directive: DGRAPH, DHISTOGRAM.
Procedures: BOXPLOT, RUGPLOT.
Genstat Reference Manual 1 Summary section on: Graphics.

DYPOLAR

Produces polar plots (D. B. Baird).

Options

MODULUS = *scalar*	Number of units required to give a complete revolution in X; default 360
TOPANGLE = *scalar*	Angle at the top of the plot; default is a quarter of the MODULUS
COLOUR = *scalar* or *text*	Colour for the lines marking rings and sectors; default 'black'
LINESTYLE = *scalar*	Linestyle for the lines marking rings and sectors; default 1
YORIGIN = *scalar*	Origin for the y-values; default 0 or the minimum of Y if this is less than 0
YMARKS = *variate*	Y-values for the rings, plotted in the background of the plot
XMARKS = *variate*	X-values for the lines marking the sectors, plotted in the background of the plot
YLABELS = *text*	Labels for the rings
XLABELS = *text*	Labels for the sectors
YTRANSFORM = *string token*	Transformed scale for the y-values (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default is to use the transform defined for YAXIS
NRINGS = *scalar*	Number of rings to be plotted, if YMARKS is not set; default 9
NSECTORS = *scalar*	Number of sectors to be plotted, if XMARKS is not set; default 12
WINDOW = *scalar*	Window number for the graph; default 1
KEYWINDOW = *scalar*	Window number for the graph key; default 2
SCREEN = *string token*	Whether to clear the screen before the plot (clear, keep); default clea
KEYDESCRIPTION = *text*	Overall description for the key; default *

Parameters

Y = *variates, factors* or *pointer*s	Y-values specifying the amplitudes of the points
X = *variates, factors* or *pointer*s	X-values specifying the angles of the points
GROUPS = *factor*s	Factor to divide the points into groups; default * i.e. none
TITLE = *texts*	Title for the graph; default forms a title automatically with the names of the Y and X structures
PEN = *scalar* or *variates*	Pens used to plot the data; default 1
DESCRIPTION = *texts*	Annotation for key; default uses the names of the Y and X structures, or the labels of GROUPS if set

Description

DYPOLAR produces polar plots, using high-resolution graphics. The data to be plotted are specified by the Y and X parameters. To plot several sets of data on the same plot, you can set Y and/or X to a pointer containing several variates or factors. Alternatively, you can use the GROUPS parameter to supply a factor to identify groups of points that are to be plotted separately.

In the plot, the X values are converted to angles (by dividing by MODULUS, and multiplying by

360 degrees). The Y values define the amplitudes of the points. An example might be to plot a time series of observations against the day of the week, month or year.

The TOPANGLE option defines the angle at the top of the plot. By default this is a quarter of the MODULUS, so that the angle zero is plotted horizontally, from left to right. The YORIGIN option defines the y-value to be used as the origin of the polar plot. By default this is zero, or the minimum of Y if this is less than zero.

The PEN parameter specifies the pens to be used to plot the data, and thus defines the method, symbols, colours etc. (See the PEN directive for more details.) The setting should be a scalar if a single set of points is to be plotted, or a variate with a value for each set if there are several.

The Y values can be transformed by using the YTRANSFORM option. The settings are the same as those of the TRANSFORM option of YAXIS. If this is not set, DYPOLAR uses the setting of the TRANSFORM option, defined by the YAXIS directive for the specified WINDOW.

Some rings and sectors are plotted in the background of the plot, to provide a sense of scale. You can specify a number of rings to display at regular intervals, by setting the NRINGS option. Similarly, the NSECTORS option allows you to specify a number of regularly spaced sectors. Alternatively, you can define the positions of the background rings and sectors explicitly, by setting the YMARKS or XMARKS options, respectively. You can provide the labels for the rings and sectors by using the YLABELS and XLABELS options (and the number of labels must match the number of rings and sectors that have been requested). The line styles and colours of the rings and sectors are controlled by the LINESTYLE and COLOUR options. You can use the XAXIS and YAXIS directives, prior to DYPOLAR, to set attributes and titles for the outside ring and the amplitude scale of the plot, respectively.

The TITLE parameter allows you to supply an overall title for the plot. You can use the DESCRIPTION parameter to provide labels for the key. If Y or X is a pointer, DESCRIPTION should contain the same number of items as the pointer or, if GROUPS is set, it should contain the same number of items as the number of groups.

The WINDOW and KEYWINDOW options specify the numbers of the windows to use for the plot and key respectively, and the SCREEN option controls whether the screen is cleared first, in the usual way (see DGRAPH). You can specify a title for the key using the KEYDESCRIPTION option.

Options: MODULUS, TOPANGLE, COLOUR, LINESTYLE, YORIGIN, YMARKS, XMARKS, YLABELS, XLABELS, YTRANSFORM, NRINGS, NSECTORS, WINDOW, KEYWINDOW, SCREEN, KEYDESCRIPTION.
Parameters: Y, X, GROUPS, TITLE PEN, DESCRIPTION.

Action with RESTRICT
If any of the variates or factors are restricted, only the units not excluded by the restriction will be plotted.

See also
Directives: DGRAPH, XAXIS, YAXIS, PEN.
Procedures: CASSOCIATION, CDESCRIBE, DCIRCULAR, RCIRCULAR, WINDROSE.
Genstat Reference Manual 1 Summary section on: Graphics.

ECABUNDANCEPLOT

Produces rank/abundance, *ABC* and *k*-dominance plots (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (summary); default summ
PLOT = *string token*	Controls the type of plot (rankabundance, kdominance, abc); default rank, kdom
GROUPS = *factor*	Defines the groups if there is more than one sample

Parameters

INDIVIDUALS = *variates*	Number of individuals per species
SPECIES = *variates*	Number of species
BIOMASS = *variates*	Biomass data for each species for an ABC plot

Description

A rank/abundance plot (or Whittaker plot) can be used to visualize species abundance distributions. In this plot, the number of individuals of each species are sorted in descending order, and the proportion of the total number of individuals for each species is then plotted on the log scale against the species rank. The shape of the rank/abundance plot can provide an indication of dominance or evenness, for example, steep plots signify assemblages with high dominance and shallower slopes indicate higher evenness.

A *k*-dominance plot displays the cumulative proportion abundance against the log species rank. For this type of plot, more elevated curves represent less diverse assemblages.

An abundance/biomass comparison (or *ABC* curve) is an adaption of the *k*-dominance curve where two measures of abundance are plotted: the number of individuals and biomass data. This plot is useful to explore the level of disturbance affecting assemblage.

The numbers of individuals per species are specified using the INDIVIDUALS parameter. The SPECIES parameter specifies a variate containing the number of species for the associated number of individuals specified in the corresponding element of INIDIVIDUALS. SPECIES can be omitted if each of the values in INDIVIDUALS corresponds to one species. The GROUPS option can be used to plot the relative abundance for different samples.

The PLOT option can be used to produce a rank/abundance plot, *k*-dominance curve and an *ABC* curve. You can display a summary of the number of individuals and species by setting the option PRINT=summary. Selecting this option will also display the *W* statistic for an *ABC* curve.

Options: PRINT, PLOT, GROUPS.
Parameters: INDIVIDUALS, SPECIES, BIOMASS.

Method

For a rank/abundance plot the numbers of individuals of each species are sorted in descending order and the proportion of the total number of individuals for each species is then plotted on the log scale against the species rank. In a *k*-dominance plot the cumulative proportion abundance, or for an *ABC* curve the cumulative percentage abundance, is plotted against the log species rank.

The *W* statistic for an *ABC* curve is defined by

$$W = \sum_i (B_i - A_i) / (50 \times (S - 1))$$

where S is the total number of Species, B_i is the biomass value of each species rank i, and A_i is the abundance value of each species rank i.

Action with RESTRICT

If a parameter is restricted the graphs will be drawn using only those units included in the restriction.

Reference

Magurran, A.E. (2003). *Measuring Biological Diversity*. Blackwell, Oxford.

See also

Genstat Reference Manual 1 Summary section on: Ecological data.

ECACCUMULATION

Plots species accumulation curves for samples or individuals (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (summary); default summ
CURVE = *string token*	Controls the type of species accumulation curve (collector, random, coleman); default coll
PLOT = *string token*	Controls plot type (sac); default sac
METHOD = *string token*	Controls collector curve when data supplied in variate or factor with groups (individual, sample); default samp
GROUPS = *factor*	Grouping factor for samples when data are supplied in variate or factor of individuals
NPERMUTATIONS = *scalar*	A scalar defining the number of permutations to be performed for the random method; default 100
SEED = *scalar*	Seed for random number generator; default 0
SCREEN = *string token*	Whether to clear screen before displaying the graph (keep, clear); defaul clea
WINDOW = *scalar*	Window for the graph; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
PEN = *scalar*	Pen number to draw the curve; default 1

Parameters

DATA = *variates*, *factors*, *matrices* or *pointers*	For individual-based collector curves, a variate or factor containing the individuals in the order they were collected; for sample-based species accumulation curves, a pointer or matrix specifying the number of individuals for each species for different sites/samples
RICHNESS = *variates*	Saves the observed number of species for the collector method and the average or expected number of species at each sample size for the Coleman and random methods
VARIANCE = *variates*	Saves the variance for the richness (Coleman and random methods only)

Description

Species accumulation curves show the rate at which new species are found within a community, and can be extrapolated to provide an estimate of species richness. The simplest type of species accumulation curve is the *collectors* curve. This plots the cumulative number of species recorded as a function of sampling effort (i.e. number of individuals collected or cumulative number of samples). The order in which samples are included in a species accumulation curve will influence the overall shape. A smooth accumulation curve can be produced by repeating a process of randomly adding the samples to the accumulation curve and then plotting the mean of these permutations. ECACCUMULATION can be used to produce these types of species accumulation curves, and can plot a *Coleman* curve of the expected number of species based on the method of Coleman *et al.* (1982); see *Method*.

For sample-based species accumulation curves, the data can be supplied using the DATA parameter, either as a matrix where the rows contain the number of individuals for each species and the columns specify the different samples or sites, or as a pointer to variates containing samples for the individuals for each species. Alternatively, the individual species numbers or labels can be supplied in either a variate or factor using the DATA parameter while the samples

are identified by supplying a grouping factor using the GROUPS option. Individual-based species accumulation curves can be formed using the collector method, where the individual species numbers or labels are specifed in either a variate or factor using the DATA parameter. The species numbers or labels must be specified in the order in which they were collected within the variate or factor. Different samples of individuals can be plotted on the same graph by supplying a grouping factor using the GROUPS option and specifying the individual setting of the METHOD option. For the collector curve the observed number of species can be saved using the RICHNESS parameter. For the random and Coleman curves the average and expected number of species and associated variance can be saved using the RICHNESS and VARIANCE parameters respectively. The type of species accumulation curve (collector, random or Coleman) is specified using the CURVE option. If the collector curve is chosen and the data have been supplied using the individual values with a grouping factor, the METHOD option can be used to choose whether to produce a sample-based plot or a plot of the individual-based curves. The number of permutations used for the random method can be supplied using the NPERMUTATIONS option, by default 100 permutations are used. The SEED option specifies the seed to use for the sub-sampling without replacements. The default value of zero continues an existing sequence of random numbers or, if the generator has not yet been used in this run of Genstat, initializes the generator automatically.

The PRINT option controls printed output, with settings:

summary	the species richness and variance (for Coleman and random methods).

A plot of the species accumulation curve can be specified using the sac setting of the PLOT option. The graphical display can be controlled using the SCREEN, WINDOW, KEYWINDOW and PEN options. By default the curves are produced in window 1 using pen 1 and drawn on a new screen.

Options: PRINT, CURVE, PLOT, METHOD, GROUPS, NPERMUTATIONS, SEED, SCREEN, WINDOW, PEN.

Parameters: DATA, RICHNESS, VARIANCE.

Method

For the collector curve the samples or individuals are added in the order they appear in the data. The random method finds the mean number of species and variance from random permutations using sub-sampling without replacement.

For the Coleman curve the expected number of species is calculated by:

$$s_\alpha = S - \sum_{i=1\ldots S} (1 - \alpha)^{ni}$$

where S is the number of species, n^i is the number of individuals belonging to ith species and α is the relative area

$$\alpha = a / \sum a_k$$

The variance is estimated by

$$v_\alpha = \sum_{i=1\ldots S} (1 - \alpha)^{ni} - \sum_{i=1\ldots S} (1 - \alpha)^{2 \times ni}$$

Further details of this method are given in Coleman *et al.* (1982).

Action with RESTRICT

If a parameter is restricted the statistics will be calculated using only those units included in the restriction.

References

Coleman, B.D., Mares, M.A. Willig, M.R. & Hsieh, Y,-H. (1982). Randomness, area, and species richess. *Ecology*, **63**, 1121-1133.

Magurran, A.E. (2003). *Measuring Biological Diversity*. Blackwell, Oxford.

See also
Genstat Reference Manual 1 Summary section on: Ecological data.

ECANOSIM

Performs an analysis of similarities i.e. *ANOSIM* (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (test); default test
PLOT = *string token*	Type of plot (boxplot, histogram); default hist
NTIMES = *scalar*	Number of permutations to make; default 999
BLOCKS = *factor*	Factor specifying groups for a stratified test; default * i.e. none
SEED = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically

Parameters

DATA = *symmetric matrices*	Similarity matrix
GROUPS = *factors*	Specify the different groups for each matrix
STATISTIC = *scalars*	Save the *R* statistics
PROBABILITY = *scalars*	Save the probabilities

Description

Analysis of similarities (*ANOSIM*) is a nonparametric method to test whether there is a significant difference between two or more groups of sampling units (Clarke 1993). The method performs a permutation test based on the ranks of measures of similarity between sampling units. The data should be supplied as a similarity matrix using the DATA parameter. The GROUPS parameter specifies a factor containing the groups for each corresponding row of the similarity matrix.

The *ANOSIM* statistic *R* is calculated by the difference of the between-group (r_b) and within-group (r_w) mean rank similarities:

$$R = (\text{mean}(r_b) - \text{mean}(r_w)) / (n \times (n - 1) / 4)$$

The denominator is chosen so the *R* lies in the range (−1, 1) where 0 represents no difference between the groups. The similarites are ranked where a rank of 1 corresponds to the highest similarity.

The statistical significance of the *R* statistic is assessed by a permutation test. ECANOSIM performs 999 random permutations (made using a default seed), and calculates the *R* statistic for each permutation. The probability for the *R* statistic is then determined from its distribution over the randomly permuted datasets. The NTIMES option of ECANOSIM allows you to request another number of permutations, and the SEED option allows you to specify another seed. For designs with no blocking ECANOSIM checks whether NTIMES is greater than the number of possible permutations available for the data set. If so, ECANOSIM does an exact test instead, which uses each possible permutation once.

The histogram setting of the PLOT option can be used to produce a distribution of the *R* values. *ANOSIM* assumes under the null hypothesis that distances within groups are smaller than those between groups, and that the ranked dissimilarities within groups have equal median and range. The boxplot setting for the PLOT option can be used to help check these assumptions.

The *R* statistic can be saved using the STATISTIC parameter, and the probability can be saved using the PROBABILITY parameter.

The PRINT option controls printed output, with a setting:

test	to print the *R* statistic and probability.

Options: PRINT, PLOT, NTIMES, BLOCKS, SEED.
Parameters: DATA, GROUPS, STATISTIC, PROBABILITY.

Method
The R statistic is calculated by:

$$R = (\text{mean}(r_b) - \text{mean}(r_w)) / (n \times (n - 1) / 4)$$

where mean(r_w) is the average of all rank similarities among replicates within sites, mean(r_b) is the average of rank similarities from all pairs of replicates between sites and n is total number of samples.

Action with `RESTRICT`
The data must not be restricted.

Reference
Clarke K,R. (1993). Non-parametric multivariate analyses of changes in community structure. *Australian Journal of Biology*, **18**, 117-143.

See also
Procedure: `MANTEL`.
Genstat Reference Manual 1 Summary section on: Ecological data.

ECDIVERSITY

Calculates measures of diversity with jackknife or bootstrap estimates (D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (index, estimate); default inde
INDEX = *string token*	Controls the type of measurement to be calculated (hshannon, qstatistic, simpsonyule, bergerparker, ibrillouin, ebrillouin, dmcintosh, emcintosh, evar, logseriesalpha, lognormallambda, jshannon, margalef, isimpson, richness); default hsha
GROUPS = *factor*	Defines the groups if there is more than one sample
BMETHOD = *string token*	Controls whether to use the bootstrap or jackknife method (jackknife, bootstrap); default jack for multiple samples and boot for individual samples
NBOOT = *scalar*	Number of times to resample in bootstrap; default 100
SEED = *scalar*	Seed for random number generator for bootstrap; default 0
CIPROBABILITY = *scalar*	Probability for the confidence interval produced by either jackknife or bootstrap method; default 0.95

Parameters

INDIVIDUALS = *variates*	Number of individuals per species
SPECIES = *variates*	Number of species
SAVE = *variate* or *pointer*	Saves the diversity indices

Description

A diversity index is a measure of species diversity within a community that consists of co-occurring populations of several (two or more) different species. There are two components to diversity: richness and evenness. Richness is the measure of the number of species within a sample where the more species in a community the higher the diversity (or greater richness). Evenness is a measure of the relative abundance of the different species within a community. The more nearly equal the species relative abundances the higher the diversity.

ECDIVERSITY can be used to calculate several different measures of diversity. Amongst these indices are the log series α and log-Normal λ which are estimated by fitting an underlying species abundance model, and the Q statistic which is derived from cumulative ranked frequencies. Other available indices include the Margalef and Simpsons $1/D$ which emphasize the richness component of diversity. The indices that highlight the evenness component of diversity include Simpsons $1-D$, McIntosh D and E, Shannon-Weiner H' and J', Brillouin diversity and evenness index, Berger-Parker and Smith-Wilson evenness measure. Confidence intervals for the measures can be estimated by bootstrapping. For multiple samples, ECDIVERSITY calculates the overall values of the diversity indices, and provides an option to perform jackknifing to produce less bias estimates with a confidence interval.

The numbers of individuals per species are specified using the INDIVIDUALS parameter. The SPECIES parameter specifies a variate containing the number of species for the associated number of individuals denoted in the corresponding element of INIDIVIDUALS. SPECIES can be omitted if each of the values in INDIVIDUALS corresponds to one species. The GROUPS option can be used to calculate measures of diversity for different samples. The SAVE parameter allows the diversity indices to be saved in a variate or in a pointer to a set of variates for each group.

The PRINT option controls printed output, with settings:

index	the index of diversity or evenness,
estimate	bootstrap or jackknife estimate with confidence limits for the statistic.

The BMETHOD option can be used to select either the bootstrap or jackknife (for multiple samples) method to produce an estimate of the diversity measure with an associated confidence interval. To produce a bootstrap or jackknife estimate for multiple samples each sample must contain the same number of values where each element corresponds to the same species within each sample. For the calculation of the bootstrap confidence intervals of the diversity measures, the NBOOT option specifies how many bootstrap samples to take (default 100). The probability level for the confidence interval can be set by the CIPROBABILITY option; by default 0.95. The SEED option specifies the seed to use in the random number generator used to construct the bootstrap samples. The default value of zero continues an existing sequence of random numbers or, if the generator has not yet been used in this run of Genstat, it initializes the generator automatically.

Options: PRINT, INDEX, GROUPS, BMETHOD, NBOOT, SEED, CIPROBABILITY.
Parameters: INDIVIDUALS, SPECIES, SAVE.

Method

The log series α index is estimated by fitting a log series model using the ECFIT procedure. The log-Normal λ is the ratio of the S^* and σ parameters estimated by fitting a Poisson-log-Normal distribution using the ECFIT procedure.

The Q statistic is calculated by:

$$Q = (0.5 \times n_{R1} + \sum_{r=R1+1 \ldots R2-1} \{ n_r \} + 0.5 \times n_{R2}) / \log(R2 / R1),$$

where n_r is the total number of species with abundance r, $R1$ and $R2$ are the 25% and 75% quartiles, n_{R1} is the number of species where R1 lies, and n_{R2} is the number of species where R2 lies.

The Shannon-Weiner index is evaluated by:

$$H' = - \sum_i (n_i / N) \times \log(n_i / N)$$

where n_i are the individuals, N is total number of individuals.

The Shannon-Weiner evenness (Pielou J) is given by

$$J' = H' / \log(S)$$

where H' is the Shannon index and S is the total number of species.

The Brillouin index is given by

$$HB = (\log(N!) - \sum_i \{\log(n_i!)\}) / N$$

where n_i is the individual in species i and N is total number of individuals.

The Brillouin evenness index is then calculated by

$$E = HB / HBmax$$

and

$$HBmax = 1 / N \times \log(N! / ((N/S)!^{S-r} \times ((N/S)+1)!^{r})$$

where N/S is the integer of N/S and $r = N - S(N/S)$

Simpsons index D is calculated by

$$D = \sum_i \{n_i \times (n_i - 1)\} / (N \times (N - 1))$$

and is expressed in the output as both $1-D$ and $1/D$

The Margalef index is:

$$Dmn = (S - 1) / \log(N)$$

where S is total number of species and N is total number of individuals.

McIntosh's measure of diversity is expressed as

$$D = (N - \sqrt{}(\sum_i \{n_i^2\} / (N - \sqrt{}(N))$$

and the evenness measure is given by

$$E = (N - \sqrt{(\sum_i \{n_i^2\})}) / (N - N / \sqrt{(S)})$$

where n_i is the individual in species i and N is total number of individuals.

The Berger-Parker index is

$$d = Nmax / N$$

where *Nmax* is the number of individuals in the most abundant species.

The Evar (Smith and Wilson 1996) evenness index is evaluated by

$$Evar = 1 - 2 / \pi \times \arctan(\sum_i \{ \log(n_i) - \sum_j \{ \log(n_j) \} \}^2 / S)$$

where n_i and n_j are the number of individuals in species i and j respectively, and S is the total number of species

Species richness is the total number of species.

The jackknife estimate and standard error are generated by the JACKKNIFE procedure where the estimates are calculated from all samples, and then for the situations where one sample is omitted in turn. The confidence interval is calculated by:

$$\varphi +/- t(n-1) \times se(\varphi)$$

where n is the number of samples.

The bootstrap confidence intervals are generated using the BOOTSTRAP procedure where all individuals are sampled with replacement and the diversity measures are calculated from these samples.

Action with RESTRICT

If a parameter is restricted the statistics will be calculated using only those units included in the restriction.

References

Magurran, A.E. (2003). *Measuring Biological Diversity*. Blackwell, Oxford.

Smith, B, & Wilson, J.B. (1996). A consumer's guide to evenness indices. *Oikos*, **76**, 70-82.

See also

Genstat Reference Manual 1 Summary section on: Ecological data.

ECFIT

Fits models to species abundance data (D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (summary, estimates, fittedvalues); default summ, esti
MODELTYPE = *string token*	The model or distribution fitted to the data (logseries, plognormal, negativebinomial, geometric, zipf, mandelbrotzipf); default logs
GROUPS = *factor*	Defines the groups if there is more than one sample
LOGBASE = *string token*	Log base to use to form the octaves for the logseries, Poisson log-Normal and negative binomial distributions (two, ten); default two
PLOT = *string token*	Plots the fitted values (fittedabundance, rankabundance); default fitt

Parameters

INDIVIDUALS = *variates*	Number of individuals per species
SPECIES = *variates*	Number of species
ESTIMATES = *variates*	Saves the model estimates
EGROUPS = *factors*	Saves the grouping of the estimates

Description

ECFIT provides a range of distributions and models that can be used to describe species abundance data. For the log series, Poisson log-Normal and negative binomial distributions the species abundance data are grouped into "octaves" using a logarithmic scale. These distributions are then fitted using the DISTRIBUTION directive using the octave classes. The geometric series, Zipf and Zipf-Mandelbrot models are fitted to the observed abundance data using the non-linear regression facilities.

The numbers of individuals per species are specified using the INDIVIDUALS parameter. The SPECIES parameter specifies a variate containing the number of species for the associated number of individuals specified in the corresponding element of INDIVIDUALS. SPECIES can be omitted if each of the values in INDIVIDUALS corresponds to one species. The GROUPS option can be used to fit models for different samples.

The distribution or model to be fitted to the data is specified by the MODELTYPE option. The log base for forming the octaves for the log series, Poisson log-normal and negative binomial distributions can be supplied using the LOGBASE option. The default is to use log base 2, i.e. representing doubling in species abundance. The parameter estimates from the fitted model can be saved using the ESTIMATES parameter. The EGROUPS factor saves a factor indicating the group strucure of the estimates.

The PRINT option controls printed output, with settings:

summary	summary of the analysis,
estimates	the parameter estimates,
fittedvalues	the fitted values.

The PLOT option can be used to produce a plot of the fitted model or distribution. For the geometric series, Zipf and Zipf-Mandelbrot models, the fitted model can also be displayed on a rank/abundance plot on the log-scale.

Options: PRINT, MODELTYPE, GROUPS, LOGBASE, PLOT.
Parameters: INDIVIDUALS, SPECIES, ESTIMATES, EGROUPS.

Method

The log series, negative binomial and poisson log normal are fitted using the DISTRIBUTION directive.

For the geometric series the abundances are ranked from the most to least abundant and fitted using FITNONLINEAR where the series is given by

$$a_i = N / (1 - (1 - k)^S) \times k \times (1 - k)^{i-1}$$

where a_i is the total number of individuals in the ith species, N is the total number of individuals, k is the proportion of remaining niche space, and $1 / (1 - (1 - k)^S)$ is a constant that ensures $\sum_i a_i = N$.

The Zipf and Zipf-Mandelbrot models are fitted using FITNONLINEAR. The Zipf model is given by

$$A_i = A_1 \times i^{-\gamma}$$

where A_1 is the fitted abundance of the most abundant species, and γ is a constant representing the average probability of the appearance of a species.

The Zipf-Mandelbrot is an extension of the Zipf model and is expressed as

$$A_i = A_1 \times (i + \beta)^{-\gamma}$$

where A_1 and gamma are as before, and beta is a constant.

Action with RESTRICT

If a parameter is restricted the models will be fitted using only those units included in the restriction.

References

Kempton, R.A. & Taylor, L.R. (1974). Log-series and log-normal parameters as diversity determinants for the Lepidoptera. *Journal of Animal Ecology*, **43**, 381-399

Magurran, A.E. (2003). *Measuring Biological Diversity*. Blackwell, Oxford.

Wilson, J.B. (1991). Methods for fitting dominance/diversity curves. *Journal of Vegetation Science*, **2**, 35-46

See also

Genstat Reference Manual 1 Summary section on: Ecological data.

ECNICHE

Generates relative abundance of species for niche-based models (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (model, expected, replications); default mode, expe
MODELTYPE = *string token*	The niche model (powerfraction, fixedratio, preemption, randomfraction, macarthurfraction); default powe
METHOD = *string token*	Whether to use the Fortran DLL to calculate the relative abundance (dll, commands); default * uses the DLL in Windows implementations, and commands for other platforms
POWER = *scalar*	Power for the Power fraction model, must be in the range 0 to 1
URATIO = *scalar*	Ratio for the fixed ratio model
SEED = *scalar*	Seed for random number generator for the random division of the niche space; default 0
PLOT = *string token*	Plots the average relative abundance (relativeabundance); default rela

Parameters

NREPLICATES = *scalars*	Number of replications
NSPECIES = *scalars*	Number of species
EXPECTED = *variates*	Saves the expected average relative abundance
SDEXPECTED = *variates*	Saves the standard deviation for the expected mean relative abundance

Description

The relative abundance of species can be modelled using deterministic models, such as the log series, or by stochastic models based on assumed patterns of resource use, such as niche-based models. ECNICHE can be used to simulate relative abundances (proportional abundance of species) for niche-apportionment, where species are considered to be associated with different processes of niche division, and sequential breakage models. Niche apportionment and sequential breakage models generate relative abundances using a two step process. In the first step the target niche (the total niche space in the very first step) is divided using a given probability distribution, for example, a random selection using the uniform distribution. In the second step a new target niche space is selected using a probabilistic weighting. The process is then repeated by dividing a selected target niche and selecting a new niche for division. ECNICHE includes Tokeshi's (1993, 1996) niche apportionment models for the dominance preemption, random fraction, power fraction and MacArthur fraction. The dominance preemption model assumes that each species in turn preempts over half the remaining niche space and is dominant over all remaining species combined. The random faction model represents the situation where new species compete for the niche space of existing species, and takes a random proportion of the previously existing niche. Therefore, species with different niche sizes or abundances have the same chance of being selected for a subsequent niche division. In the power fraction model, the probability of selection is proportional to niche size (or abundance) raised to a power exponent k ($0 \leq k \leq 1$). In the MacArthur fraction model (broken-stick model) the probability of a niche being selected for division is related to its size. So, larger niches are more likely to be invaded by species. ECNICHE also provides the sequential breakage model where the target niche is selected at random and then divided to produce two

segments relative to a ratio such as 0.75:0.25.

The number of replications for the model are specified using the NREPLICATES parameter. The NSPECIES parameter specifies the number of species within the assemblage. The mean relative abundance of species and associated standard deviations can be saved using the EXPECTED and SDEXPECTED parameters respectively.

The model to use to generate the relative abundances for the species is specified by the MODELTYPE option. The power for the Power fraction model is specified using the POWER option, and must range between 0 and 1. For the sequential breakage model, the largest value of the ratio of division is specified using the URATIO option, and must range between 0.5 and 1. The SEED option specifies the seed to use in the random division of the niche space. The default value of zero continues an existing sequence of random numbers or, if the generator has not yet been used in this run of Genstat, initializes the generator automatically.

For a large number of replications the calculation of the relative abundance of species can be slow. For the PC Windows implementation, a Fortran DLL is available that uses the OWN calculate function. By default the procedures uses the DLL, however, you can choose to use the Genstat commands by setting option METHOD=commands.

The PRINT option controls printed output, with settings:

model	the niche model,
expected	the expected mean relative abundance,
replications	the relative abundances for each replication; this can produce a lot of output, so it is recommended that this be used only for monitoring.

By default PRINT=model,expected.

The PLOT option controls whether ECNICHE produces a plot of the average relative abundance on the log scale; the default PLOT=relativeabundance gives the plot.

Options: PRINT, MODELTYPE, METHOD, POWER, URATIO, SEED, PLOT.
Parameters: NREPLICATES, NSPECIES, EXPECTED, SDEXPECTED.

Method

For the dominance preemption model the niche space is divided by a random (uniform) split between 0.5 and 1.0. This model is similar to the geometric series, and over many replications will produce a similar distribution of species abundance when $k = 0.75$ (see ECFIT for details of geometric series). For the power fraction the probability of a niche being selected for division is $p_i = a\, x_i^k$, where a is constant common to all the species in an assemblage and $\sum_i \{a\, x_i^k\} = 1$, and x_i denotes the niche size of species i. The random fraction is formed using the power fraction model with $k = 0$, i.e. a size-independent selection probability. Similarly, the MacArthur fraction is calculated using the power fraction with $k = 1$, i.e. probability of selection is a linear function of the segment length. The sequential breakage or fixed ratio model uses a deterministic division where the segments are divided to produce lengths relative to a ratio, such as 0.75:0.25.

References

Magurran, A.E. (2003). *Measuring Biological Diversity*. Blackwell, Oxford.

Tokeshi, M. (1993). Species Abundance Patterns and Community Structure. *Advances in Ecological Research*, **24**, 111-186.

Tokeshi, M. (1996). Power fraction: a new explanation of relative abundance patterns in species-rich assemblages. *Oikos*, **75**, 543-550.

See also

Genstat Reference Manual 1 Summary section on: Ecological data.

ECNPESTIMATE

Calculates nonparametric estimates of species richness (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (summary, estimates); default summ, esti
GROUPS = *factor*	Grouping factor for different samples
NBOOT = *scalar*	A scalar defining the number of bootstrap samples to be performed; default 100
SEED = *scalar*	Seed for random number generator; default 0

Parameters

DATA = *variates*, *matrices* or *pointers*	A variate containing abundances of species or a pointer or matrix specifying the individuals for each species for different sites/samples
ESTIMATES = *variates* or *pointer*	Saves the estimated species richness in a variate, or in a pointer if GROUPS are specified
SE = *variates* or *pointers*	Saves the analytic standard errors in a variate, or in a pointer if groups are specified
BSE = *variates* or *pointers*	Saves the bootstrap standard errors in a variate, or in a pointer if groups are specified

Description

Richness is the measure of the number of species within a sample. ECNPESTIMATE provides a number of nonparametric estimators for measuring true species richness. These estimators include the Chao 1, Chao 2, ACE, ICE, first-order jackknife, second-order jackknife and bootstrap. The Chao 1 and ACE are based on the abundances within the samples, whereas the other estimators are incidence-based using frequencies of species in a set of samples. Standard errors are calculated using analytical results where possible. In addition, for multiple samples, standard errors are calculated by resampling with replacement.

The data can be supplied using the DATA parameter either as a matrix where the rows contain the number of individuals for each species and the columns specify the different samples or sites, or as a pointer to variates containing samples for the individuals for each species. Alternatively, the individual species numbers can be supplied in a variate for a single sample/site. The GROUPS option can supply a grouping factor to produce estimates for different groups. The estimates and standard errors can be saved using the ESTIMATES, SE (analytic standard errors) and BSE (bootstrap standard errors) parameters. If a grouping factor is supplied then they will be saved in a pointer to variates, otherwise they are saved in a variate.

The PRINT option controls printed output, with settings:

summary	a summary of the data,
estimates	the species richness estimates and standard errors.

The NBOOT option specifies how many bootstrap samples to take to calculate the bootstrap standard errors and confidence intervals (default 100). The probability level for the confidence interval can be set by the CIPROBABILITY option; by default 0.95. The SEED option specifies the seed to use in the random number generator used to construct the bootstrap samples. The default value of zero continues an existing sequence of random numbers or, if the generator has not yet been used in this run of Genstat, it initializes the generator automatically.

Options: PRINT, GROUPS, NBOOT, SEED.
Parameters: DATA, ESTIMATES, SE, BSE.

Method

The Chao 1 estimator of the absolute number of species in an assemblage is calculated by:

$$\text{s(Chao 1)} = S_{obs} + F_1^2 / (2 \times F_2)$$

where S_{obs} is the number of species in the sample, F_1 is the number of observed species represented by a single individual (frequency of singletons), and F_2 is the number of species that have exactly two individuals (frequency of doubletons). The variance for the estimate is given by:

$$\text{var(Chao 1)} = F_2 \times \{ 0.5 \times (F_1 / F_2)^2 + (F_1 / F_2)^3 + 0.25 \times (F_1 / F_2)^4 \}$$

When F_2 equals 0 the modified bias-corrected estimate is used:

$$\text{s(Chao 1)} = S_{obs} + F_1 \times (F_1 - 1) / 2$$

and

$$\text{var(Chao 1)} = \{F_1 \times (F_1 - 1) / 2\} + \{F_1 \times (2 \times F_1 - 1)^2 / 4\} - F_1^4 / (4 \times \text{s(Chao 1)})$$

The Chao 2 estimator is calculated by:

$$\text{s(Chao 2)} = S_{obs} + Q_1^2 / (2 \times Q_2)$$

where S_{obs} is the number of species in sample, Q_1 is the number of species that occur in exactly one sample (uniques), and Q_2 is the number of species that occur in exactly two samples (duplicates). The variance for the estimate is given by:

$$\text{var(Chao 2)} = Q_2 \times \{ 0.5 \times (Q_1 / Q_2)^2 + (Q_1 / Q_2)^3 + 0.25 \times (Q_1 / Q_2)^4 \}$$

When Q_2 equals 0 the modified bias-corrected estimate is used:

$$\text{s(Chao 2)} = S_{obs} + Q_1 \times (Q_1 - 1) / 2$$

and

$$\text{var(Chao 2)} = \{(H - 1) / H\} \times Q_1 \times (Q_1 - 1) / 2 + \{(H - 1) / H\}^2 \times Q_1 \times \{2 \times Q_1 - 1)^2\} / 4 + \{(H - 1) / H\}^2 \times Q_1^4 / (4 \times \text{Chao2})$$

where H is the total number of samples.

The first-order jackknife estimate is evaluated by:

$$\text{s(jack1)} = S_{obs} + Q_1 \times (H - 1) / H$$

with variance

$$\text{var(jack1)} = \{(H - 1) / H\} \times \{ \sum_{j=1...S} (j^2 \times f_j) - (Q_1^2 / H) \}$$

where S is the number of species, Q_1 is the number of species that occur in exactly one sample and f_j is the number of samples with j unique species.

The second-order jackknife estimate is given by:

$$\text{s(jack2)} = S_{obs} + Q_1 \times (2 \times H - 3) / H - Q_2 \times (H - 2)^2 / \{H \times (H - 1)\}$$

where Q_1 is the number of species that occur in exactly one sample, and Q_2 is the number of species that occur in exactly two samples.

The bootstrap estimate is calculated by:

$$\text{s(boot)} = S_{obs} + \sum_{j=1...S} (1 - p_j)^H$$

where p_j is the proportion of species j. The variance is calculated using the method given in Smith & van Belle (1984).

The abundance-based coverage estimator (ACE) is given by:

$$\text{s(ACE)} = S_{abund} + S_{rare} / C_{ACE} + (F_1 / C_{ACE}) \times \gamma^2$$

where S_{abund} is the number of abundant species (>10), S_{rare} is the number of rare species ($\leq$10), F_1 is the number of singletons,

$$C_{ACE} = 1 - F_1 / N_{rare}$$

where N_{rare} is the total number of individuals in rare species, and

$$\gamma = \max \{(S_{rare} / C_{ACE}) \times \sum_{i=1...10} \{i \times (i-1) \times F_i\} / (N_{rare} \times (N_{rare} - 1)) - 1, 0\}$$

The incidence-based coverage estimator (ICE) is given by:

$$\text{s(ICE)} = S_{freq} + S_{infr} / C_{ICE} + (Q_1 / C_{ICE}) \times \gamma^2$$

where S_{freq} is the number of frequent species (>10), S_{infr} is the number of infrequent species (<=10), Q_1 is the number of uniques, $C_{ICE} = 1 - Q_1 / N_{infr}$ where N_{infr} is the total number of

occurrences of infrequent species, and

$$\gamma = \max\{(S_{infr}/C_{ICE}) \times (M_{infr}/(M_{infr}-1)) \times (\sum_{i=1...10} \{i \times (i-1) \times Q_i\} / N_{infr}^2) - 1, 0\}$$

where M_{infr} is the number of samples with at least one infrequent species.

The bootstrap standard errors are generated using the `BOOTSTRAP` procedure sampling with replacement, and the species richness estimates are calculated from these samples.

Action with `RESTRICT`

If the data are in a variate, the statistics are calculated using only those units included in the restriction. If data are in a pointer or matrix, the restriction are ignored.

References

Chao, A. (1987). Estimating the population size for capture-recapture data with unequal catchability. *Biometrics*, **43**, 783-791.

Magurran, A.E. (2003). *Measuring Biological Diversity*. Blackwell, Oxford.

Smith, E.P. & van Belle, G. (1984). Nonparametric estimation of species richness. *Biometrics*, **40**, 119-129.

See also

Genstat Reference Manual 1 Summary section on: Ecological data.

ECRAREFACTION

Calculates individual or sample-based rarefaction (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (summary); default summ
METHOD = *string token*	Controls the type of rarefaction (individual, sample); default indi
PLOT = *string token*	Controls plot type (expected); default expe
SAMPLESIZES = *scalar* or *variate*	A scalar defining a step between sample sizes or number of samples to estimate the number of species; alternatively, a variate specifing the actual sample size values or number of samples
CIPROBABILITY = *scalar*	Probability for the confidence interval; default 0.95

Parameters

DATA = *variates*, *matrices* or *pointers*	For individual-based rarefaction, a variate containing the number of individuals for each species; for sample-based rarefaction, a pointer or matrix specifying the number of individuals for each species for different sites/samples
EXPECTED = *variates*	Saves the expected number of species at each sample size
VARIANCE = *variates*	Saves the variance for the expected number of species
LOWER = *variates*	Saves the lower confidence limit at each sample size
UPPER = *variates*	Saves the upper confidence limit at each sample size

Description

Rarefaction is a method that can be used to estimate the number of species that would be found if sampling effort was reduced to a specified level. This then allows comparisons amongst communities where sampling effort is unequal. For individuals in a sample, individual-based rarefaction can be used to estimate the number of species that would be observed given a smaller number of individuals (Heck *et al*. 1975). Sample-based rarefaction can be used to estimate the expected number of species that would be observed given a smaller number of samples (Colwell *et al*. 2004). Rarefaction assumes that individuals have been sampled randomly and sample-based rarefaction assumes a random sample ordering. The method also assumes that the samples that are to be compared are not obtained by different collecting techniques or from communities that are intrinsically different.

For individual-based rarefaction, the number of individuals for each species is specified in a variate using the DATA parameter. For sample-based rarefaction, the data can be supplied using the DATA parameter either as a matrix where the rows contain the number of individuals for each species and the columns specify the different samples, or as a pointer to variates containing samples for the individuals for each species. The expected number of species and associated variance can be saved using the EXPECTED and VARIANCE parameters respectively. The LOWER and UPPER parameters can be used to save the lower and upper bounds for the confidence interval. The type of rarefaction (individual or sample-based) is specified using the METHOD option. The SAMPLESIZES option specifies the sample sizes or number of samples for which the expected number of species is calculated. A scalar can be supplied to specify a step between each sample size, or a variate can be provided containing the actual sample sizes. By default the expected values are calculated for all possible sample sizes.

The PRINT option controls printed output, with settings:

summary	the expected species richness, variance and confidence

limits.

A plot of the expected number of species and confidence limits can be specified using the `expected` setting of the `PLOT` option. The probability level for the confidence intervals can be set by the `CIPROBABILITY` option; by default 0.95.

Options: `PRINT`, `METHOD`, `PLOT`, `SAMPLESIZES`, `CIPROBABILITY`.
Parameters: `DATA`, `EXPECTED`, `VARIANCE`, `LOWER`, `UPPER`.

Method

For individual-based rarefaction the expected number of species in a sample of size n is calculated by:

$$E(S_n) = S - (1 / C(n, N)) \times \sum_i \{ C(n, N-N_i) \}$$

where N_i is the number of individuals in species i of the unrarefied sample, $C(n, N)$ is the number of combinations of n from N and $C(n, N-N_i)$ is the number of combinations of n from $N-N_i$. The variance, var(S_n), is outlined in Heck *et al*. (1975).

Sample-based rarefaction is calculated by

$$t(h) = S_{obs} - \sum_{j=1...H} \{ a_{jh} \times s_j \} \quad \text{for } h = 1 \ldots H$$

where s_j is the number of species found in exactly j samples of a total of H samples, S_{obs} is defined by

$$S_{obs} = \sum_{j=1...H} \{ s_j \}$$

and the combinational coefficients a_{jh} are estimated by

$$a_{jh} = ((H - h)! \times (H - j)!) / ((H - h - j)! \times H!) \quad \text{for } j + h \le H$$

$$a_{jh} = 0 \quad \text{otherwise}$$

The variance is estimated by

$$\text{var}(h) = \sum \{ (1 - a_{jh})^2 \times s_j - t(h)^2 / S^{\sim}$$

where

$$S^{\sim} = S_{obs} + (H - 1) \times s_1^2 / (2 \times H \times s_2)$$

Further details of this method are given in Colwell *et al*. (2004).

Action with `RESTRICT`

If a parameter is restricted the statistics will be calculated using only those units included in the restriction.

References

Colwell, R.K., Mao, C.X. & Chang, J. (2004). Interpolating, extrapolating comparing incidence-based species accumulation curves. *Ecology*, **85**, 2717-2727.

Heck, K.L., van Belle, G. & Simberloff, D. (1975). Explicit calculation of the rarefaction diversity measurement and the determination of sufficient sample size. *Ecology*, **56**, 1459-1461.

See also

Genstat Reference Manual 1 Summary section on: Ecological data.

EDDUNNETT

Calculates equivalent deviates for Dunnett's simultaneous confidence interval around a control (R.W. Payne).

Options

METHOD = *string token*	Form of the alternative hypothesis (`twosided`, `greaterthan`, `lessthan`); default `twos`
NTREATMENTS = *scalar*	Number of treatments being compared
DF = *scalar*	Number of residual degrees of freedom
REPTREATMENTS = *scalar* or *variate*	Specifies the replication of the treatments
REPCONTROL = *scalar*	Specifies the replication of the control
TOLERANCE = *scalar*	Tolerance for the difference between the probability for the calculated equivalent deviate and that requested by CIPROBABILITY; default 0.0001

Parameters

CIPROBABILITY = *scalars*	Specifies the probability for the confidence interval
ED = *scalars*	Saves the equivalent deviate

Description

Dunnett's test is useful when you want to compare several treatments with a control treatment, and use a critical value that controls the chance that any one comparison may be found significant when there are no true differences. (It is designed thus to take account of the fact that you are making multiple comparisons with the control.) The test can be preformed in Genstat using the AMDUNNETT procedure, and this uses EDDUNNETT to calculate the critical value (i.e. the equivalent deviate of the probability distribution).

The METHOD option defines the type of interval that is formed. By default EDDUNNETT assumes a two-sided interval. If you set METHOD=lowerthan, it assumes a lower confidence interval, assessing the one-sided test of the null hypothesis that the treatment means are not lower than the control mean. Alternatively, you can set METHOD=greaterthan, to assume an upper confidence interval, assessing the one-sided test of the null hypothesis that the treatment means are not greater than the mean of the control.

The NTREATMENTS option specifies the number of treatments that are being compared with the control. The REPTREATMENTS option specifies their replication, in a *scalar* if they all have the same replication, or in a *variate* if their replications differ. The REPCONTROL option specifies the replication of the control, and the DF option specifies the number of residual degrees of freedom.

The probability for the confidence interval is specified by the CIPROBABILITY parameter, and the ED parameter saves the equivalent deviate.

The equivalent deviate is estimated by an iterative process. The TOLERANCE option controls the accuracy of the estimation, defining the how close the probability corresponding to the calculated equivalent deviate must be to that requested by CIPROBABILITY; default 0.0001

Options: METHOD, NTREATMENTS, DF, REPTREATMENTS, REPCONTROL, TOLERANCE.
Parameters: CIPROBABILITY, ED.

See also

Procedure: AMDUNNETT.

EDFTEST

Performs empirical-distribution-function goodness-of-fit tests (V.M. Cave).

Options

PRINT = *string tokens*	Controls printed output (summary, tests); default summ, test
PLOT = *string tokens*	What graphs to plot (kerneldensity, histogram); default *
TEST = *string tokens*	Specifies the type of goodness-of-fit test to perform (andersondarling, cramervonmises, kolmogorovsmirnov); default ande, cram, kolm
DISTRIBUTION = *string tokens*	Continuous distribution that is hypothesized to have generated the DATA; (beta, b2, burr, cauchy, chisquare, ev1 (or gumbel), ev2 (or frechet), ev3, exponential, fdistribution, gamma, gev, gpareto, iburr, igamma, invnormal, iweibull, laplace, loggamma, logistic, loglogistic, lognormal, normal, paralogistic, pareto, stdnormal, stduniform, tdistribution, ubetamix, ugammamix, uniform, weibull, calculated); default norm
CONSTANT = *string tokens*	Whether to estimate a constant for the distribution, when the parameter values are estimated from the DATA (estimate, omit); default omit
TMETHOD = *string tokens*	Specifies the method used to perform the goodness-of-fit tests (likelihoodratio, traditional); default like
PARAMETERS = *scalar* or *variate*	Parameter values for the hypothesized distribution; if this is not set, parameter values are estimated from the DATA
NAMES = *text*	Names to identify the parameters in PARAMETERS; if this is not set, the default parameter ordering is assumed
CDFCALCULATION = *expression*	Expression, formed using argument X, that defines the cumulative distribution function of the hypothesized distribution; must be specified when DISTRIBUTION = calculated
MCPARAMETERS = *string tokens*	Whether the parameters are re-estimated or fixed during the Monte-Carlo simulations, when the parameter values are estimated from the DATA (fix, estimate); default esti
NTIMES = *scalar*	Number of Monte-Carlo simulations to perform; default 999
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
TITLE = *text*	Title for the graphs; default generates the title automatically
YTITLE = *text*	Y-axis title for the graphs; default generates the title automatically
XTITLE = *text*	X-axis title for the graphs; default generates the title automatically
WINDOW = *scalar*	Window to use for the graphs; default 3

SCREEN = *string tokens*	Whether to clear the screen before plotting the graph or to continue plotting on the old screen, when a single graph is requested (clear, keep); default clear

Parameters

DATA = *variate*	Identifier of the variate holding the data
STATISTIC = *pointer*	Pointer to scalar(s) to save the test statistic(s)
MCSTATISTICS = *pointer*	Pointer to variates(s) to save the Monte-Carlo simulated test statistic(s)
PROBABILITY = *pointer*	Pointer to scalar(s) to save the probability value(s) of the test statistic(s)

Description

EDFTEST performs one-sample two-sided empirical-distribution-function goodness-of-fit tests to assess whether a sample of data comes from a specified continuous distribution. The data values must be supplied, in a variate, using the DATA parameter. The type of tests to be performed are specified by the TEST option, with settings andersondarling (Anderson-Darling), cramervonmises (Cramér-von Mises) and kolmogorovsmirnov (Kolmogorov-Smirnov).

The method used to perform these tests is specified by the TMETHOD option, with settings likelihoodratio for the Zhang (2002) likelihood-ratio based method, and traditional for the traditional approach. The default is to use the likelihood-ratio based tests, which are generally more powerful.

The distribution from which the data are assumed to arise is specified using the DISTRIBUTION option; default normal. Values for the parameters can be supplied, in either a scalar or a variate, by the PARAMETERS option. However, when parameter values are supplied, a value must be specified for every parameter. If parameter values are not supplied, they are estimated from the DATA (except when DISTRIBUTION is set to stdnormal, stduniform or calculated).

The NAMES option specifies a text to identify the individual parameter values within a variate of PARAMETERS. The parameter names associated with each distribution are given below. When the names are not supplied, the default ordering of the parameters is assumed. (This matches the ordering in which parameter estimates are saved using the ESTIMATES parameter of the DPROBABILITY procedure,) The parameter names are listed below, in the default parameter ordering for each distribution:

Beta Type I (beta)	ashape, bshape;
Beta Type II (b2)	ashape, bshape, rate;
Burr (burr)	ashape, scale, bshape;
Cauchy (cauchy)	location, scale;
Chi-square (chisquare)	df;
Extreme Value Type I (ev1 or gumbel)	location, scale;
Extreme Value Type II (ev2 or frechet)	location, scale, shape;
Extreme Value Type III (ev3)	location, scale, shape;
Exponential (exponential)	rate;
F (fdistribution)	ndf, ddf;
Gamma (gamma)	shape, rate, constant (optional);
Generalized Extreme Value (gev)	shape, location, scale;

Generalized Pareto (gpareto)	shape, scale;
Inverse Burr (iburr)	ashape, scale, bshape;
Inverse Gamma (igamma)	shape, scale;
Inverse Normal (invnormal)	mean, shape;
Inverse Weibull (iweibull)	scale, shape;
Laplace (laplace)	location, scale;
Log-Gamma (loggamma)	shape, rate;
Logistic (logistic)	location, scale;
Log-Logistic (loglogistic)	shape, scale;
Log-Normal (lognormal)	mean, sd, constant (optional);
Normal (normal)	mean, sd;
Paralogistic (paralogistic)	shape, scale;
Pareto (pareto)	shape, scale, constant (optional);
t (tdistribution)	df;
Uniform-Beta mixture (ubetamix)	weight, ashape, bshape;
Uniform-Gamma mixture (ugammamix)	weight, shape, scale;
Uniform (uniform)	min, max;
Weibull (weibull)	shape, rate, constant (optional);

The Gamma, Log-Normal, Pareto and Weibull distributions can have an extra constant parameter, so that the data values minus the constant then follow the specified distribution. When PARAMETERS are not supplied, you can set option CONSTANT = estimate to estimate a constant from the DATA. The default is not to estimate a constant.

The DISTRIBUTION option provides the common distributions. Alternatively, for traditional tests (i.e. TMETHOD = traditional) you can set DISTRIBUTION=calculated to define your own distribution. You must then use the CDFCALCULATION option to provide an expression, formed using argument X, to calculate the cumulative distribution function. For example, the exponential distribution with rate parameter of 2 could be specified by setting options

```
DISTRIBUTION=calculated
```

and

```
CDF=!E(X=1-EXP(-2*X))].
```

Monte-Carlo simulations are used to calculate the empirical probability values of the test statistics under the likelihood-ratio based method (i.e. TMETHOD = likelihoodratio), or, by default, under the traditional method when the parameters are estimated from the DATA. The NTIMES option defines how many Monte-Carlo simulations are used; default 999. The SEED option can be set to initialize the random-number generator used during the Monte-Carlo simulations; if the procedure is called again with the same settings, you will get identical results. The default of zero continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, the seed is initialized automatically.

By default, when parameters are estimated from the DATA during the Monte-Carlo simulations, the parameters are re-estimated to ensure that the correct probability values are obtained. However, this can be overridden by setting the MCPARAMETERS option to fix.

Printed output is controlled by the PRINT option, with settings:

summary	to print summary information; and
tests	to print the test statistic(s), with its probability value(s)

under the assumption that the data are from the hypothesized distribution (so a low probability indicates that the data are unlikely to be from the hypothesized distribution).

The printed output can be suppressed by setting option PRINT = *. The default is to print the summary and the tests.

The PLOT option controls graphical output, with settings:

histogram	to plot a histogram of the Monte-Carlo simulated test statistics; and
kerneldensity	to produce a kernel density plot of the Monte-Carlo simulated test statistics.

By default, nothing is plotted.

The TITLE, YTITLE and XTITLE options can supply an overall title, a y-axis title and a x-axis title for the graphs, respectively. If these are not supplied, suitable titles are generated automatically. When a single plot is requested, you can set option SCREEN = keep to plot the graph on an existing screen; by default the screen is cleared first. The WINDOW option defines the window to use for the plots; default 3.

The STATISTIC, PROBABILITY and MCSTATISTICS parameters allow the test statistics, their probabilities and the Monte-Carlo simulated test statistics to be saved, respectively, in pointers.

Options: PRINT, PLOT, DISTRIBUTION, CONSTANT, TMETHOD, PARAMETERS, NAMES, CDFCALCULATION, MCPARAMETERS, NTIMES, SEED, TITLE, YTITLE, XTITLE, WINDOW, SCREEN.

Parameters: DATA, STATISTIC, MCSTATISTICS, PROBABILITY.

Method

If TMETHOD=traditional, EDFTEST calculates the traditional Anderson-Darling, Cramér-von Mises and Kolmogorov-Smirnov goodness-of-fit tests. When PARAMETERS are supplied (or if MCPARAMETERS = fix), the probability of the Anderson-Darling test statistic is calculated using the fast algorithm (adinf) of Marsaglia & Marsaglia (2004), the probability of the Cramér-von Mises test statistic is calculated using the one-term linking approximation (equation 1.8) of Csörgő & Faraway (1996), and the probability of the Kolmogorov-Smirnov test statistic is calculated using the method of Carvalho (2015) for data sets with fewer than 171 values or using the Wang *et al.* (2003) approximation for larger data sets. When PARAMETERS are not supplied, Monte-Carlo simulation is used by default to obtain empirical probability values of the test statistics. However, empirical probability values are not available for DISTRIBUTION = ubetamix or ugammamix.

If TMETHOD = likelihoodratio, EDFTEST calculates likelihood-ratio based goodness-of-fit test statistics using the method of Zhang (2002). (Note, however, that the likelihood-ratio based method is not available for DISTRIBUTION = ubetamix, ugammamix, or calculated.) The resulting tests are generally more powerful than their traditional analogues. Monte-Carlo simulation is used to obtain empirical probability values of the test statistics.

When PARAMETERS are not supplied, maximum-likelihood estimates are obtained using the methods in the DPROBABILITY procedure. When MCPARAMETERS = estimate, the parameter values are re-estimated for each simulated data set using the DPROBABILITY procedure.

The kernel-density plot is generated by the KERNELDENSITY procedure, using the method of Sheather & Jones (1991), with the default number of grid points. The simulated test statistics are plotted using red + symbols along the x-axis, and the location of the test statistic is denoted by a blue line. As the observed test statistic contributes to the null distribution, it is included in the calculation of both the kernel density and histogram.

Action with RESTRICT

The DATA variate can be restricted to assess a subset of the data.

References

Carvalho, L. (2015). An improved evaluation of Kolmogorov's distribution. *Journal of Statistical Software*, **65**(3), 1-7.

Csörgő, S. & Faraway, J.J. (1996). The exact and asymptotic distributions of Cramér-von Mises statistics. *Journal of the Royal Statistical Society, Series B*, **58**, 221-234.

Marsaglia, G. & Marsaglia, J. (2004). Evaluating the Anderson-Darling distribution. *Journal of Statistical Software*, **9**(2), 1-5.

Sheather, S.J. & Jones, M.C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B*, 53, 683-690.

Wang, J., Tsang, W.W. & Marsaglia, G. (2003). Evaluating of Kolmogorov's distribution. *Journal of Statistical Software*, **8**(18), 1-4.

Zhang (2002). Powerful goodness-of-fit tests based on the likelihood ratio. *Journal of the Royal Statistical Society, Series B*, **64**, 281-294.

See also

Directive: DISTRIBUTION.

Procedures: DPROBABILITY, NORMTEST, KOLMOG2, WSTATISTIC.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

EXAMPLE

Obtains and runs a Genstat example program, PC Windows only (R.W. Payne).

Option

EXECUTE = *string token*	Whether to run the example when Genstat is running interactively (no, yes); default no

Parameters

EXTYPE = *texts*	Types of example
EXNAME = *texts*	Names of example
SOURCE = *texts*	Texts to store the source code of each example
STATEMENT = *texts*	Saves a command to obtain each example (useful if the name and type information has been specified in response to questions from EXAMPLE)

Description

EXAMPLE provides a convenient way of accessing the example programs that are distributed with Genstat. EXAMPLE is easiest to use interactively (and this is what happens if you select Help followed by Example Programs from the menu bar of Genstat *for Windows*). It then lists the types of example, followed (once you have selected the type) by a list of the available examples. If, however, you wish to access the same example later, the STATEMENT parameter allows you to save a Genstat text structure containing a command specifying the necessary EXTYPE and EXNAME parameters. For example

```
EXAMPLE 'plib'; 'GLMM'
```

obtains the example for the GLMM procedure (the plib module contains all the procedures in the current library).

When you are running Genstat interactively, the example is put into a new input window. If the EXECUTE option is set to yes, the example will also be executed. If EXECUTE is not set, you will be asked if you want to run it. When Genstat is running in batch, the example is printed to the output window (and cannot be executed). You can also save the example in a Genstat text structure, using the SOURCE parameter.

Option: EXECUTE.
Parameters: EXTYPE, EXNAME, SOURCE, STATEMENT.

See also

Directive: HELP.
Procedure: LIBEXAMPLE.

EXPORT

Saves data structures in Genstat, Excel, R, Quattro, dBase, SPlus, Gauss, MatLab, SAS, Instat, Image or text files (D.B. Baird).

Options

`PRINT` = *string token*	What to print (`summary`); default `summ`
`OUTFILE` = *text*	Data file to be written
`METHOD` = *string token*	Action to take if the file already exists (`add`, `append`, `concatenate`, `merge`, `overwrite`, `prompt`, `fail`, `replace`); default `prompt` in interactive mode, `fail` in batch mode
`PLAINNAMES` = *string token*	Whether to leave the column names in the file in plain form rather than decorating them with the column type information i.e. `!` for factors, `:D` for dates etc (`yes`, `no`) default `no`
`SHEETNAME` = *text*	Name of new sheet to be added to an existing Excel file
`NONAMES` = *string token*	Whether to suppress column names in output to spreadsheet or text file (`yes`, `no`); default `no`
`TITLE` = *text*	Description for spreadsheet
`READONLY` = *string token*	Whether to define the complete sheet as read only (`yes`, `no`); default `no`
`ANALYSIS` = *text*	Genstat commands to analyse columns in the spreadsheet
`ASETUP` = *text*	Genstat commands to be run once before the analysis of any columns in the spreadsheet
`ADUMMY` = *text*	The name of the dummy (if any) used the `ANALYSIS` commands
`CSVOPTIONS` = *string tokens*	Options for CSV files (`noquotes`, `pack`, `round`, `fixed`, `align`); default `pack`
`HTMLOPTIONS` = *string token*	Options for HTML files (`allowformats`, `nogrid`, `centre`, `rightjustify`); default `*` i.e. none
`COLMATCH` = *string token*	How to match columns when appending (`name`, `position`); default `posi`
`GROUPS` = *factor* or *text*	Identifier for the factor, or text containing the name of the factor, to identify appended sections in the output file
`GLABEL` = *texts*	Labels for the `GROUPS` factor for the current appended section, and also for the original section if no previous sections have been appended
`MATCH` = *texts*, *variates* or *pointers*	Up to four `DATA` variables to use as keys when `METHOD=merge`; default `*` uses the first `DATA` variable
`WITH` = *texts*, *variates* or *pointers*	Columns in the file to use as keys when `METHOD=merge`; default `*` uses as many columns of the initial columns in the file as are needed to give a column for each `MATCH` column
`UPDATE` = *string token*	Whether to use columns with matching names to replace existing columns when concatenating or merging (`yes`, `no`); default `no` changes the names of columns with the same name as existing columns so that they become unique
`OUTOPTIONS` = *text*	Optional output file arguments to be passed to the

	`Dataload.dll`
`ROWCOLOURS` = *factor*	The factor to be used for colouring the rows (the factor must have colours defined by the `FACCOLOURS` parameter)
`TABLEFORMAT` = *string token*	The format to use when displaying tables with two or more classifying factors (`page`, `column`); default `page`
`MISSING` = *text*	String to represent a numerical missing value when writing to a text file (`.TXT`, `.TAB` or `.CSV`) or a spreadsheet file (Excel, Quattro or Open Office); default is to use `'*'` in `.TXT` or `.TAB` files, and leave cells with missing values empty in csv or spreadsheet files
`DELETESHEETS` = *string token*	Whether to delete sheets if you are overwriting a multiple paged file with a single page (`always`, `never`, `prompt`); default `prom` when running interactively and `neve` when running in batch
`NONASCII` = *string token*	Specifies how to output non-ASCII characters to text files (`utf8`, `unicode`); default `utf8`
`TIMEOUT` = *scalar*	Number of seconds to wait when a file is open in another process; default 10

Parameters

`DATA` = *identifiers*	The data structures to be written to the file, these must be compatible (i.e. of the same length)
`COLUMNS` = *texts*	Names for the columns to be saved
`PROTECT` = *scalars*	Whether the column is to be defined as read only when option `READONLY=no` (`yes`, `no`); default `no`
`FACCOLOURS` = *variates, texts* or *pointers*	Specifies background colours for factor columns
`FOREGROUND` = *variates, texts, scalars* or *pointer*	Specifies foreground colours for columns
`BACKGROUND` = *variates, texts, scalars* or *pointer*	Specifies background colours for columns
`DECIMALS` = *variates* or *scalars*	Specifies numbers of decimals for the columns

Description

`EXPORT` saves the data structures specified by the `DATA` parameter to a disk file specified by the `OUTFILE` option in a foreign data format specified by the extension of the file name. The available extensions are: `.XLS` for Excel, `.XLSX` for Excel 2007-13, `.WQ1` for Quattro, `.ODS` for Open Office Spreadsheet, `.DBF` for dBase, `.FMT` for Gauss, `.SDD` for SPlus, `.RDA` for R, `.TPT` for SAS transport, `.WOR` for Instat, `.MAT` for MatLab, `.ARFF` for Weka Attribute, `.TXT` for plain ASCII text, `.CSV` for comma delimited text, `.TAB` for tab delimited text, `.HTM` for a HTML table, `.RTF` for Word Rich text format, `.GSH` for Genstat spreadsheet, `.GWB` for Genstat work book, and `.BMP`, `.EMF`, `.GIF`, `.JPG`, `.TIF`, `.PNG` or `.PSD` for an image file. An image file can be created either from single matrix containing RGB colour values, or three columns of variates or factors columns (specifying x-coordinates, y-coordinates and RGB colour values), or five columns of variates or factors columns (specifying x-coordinates, y-coordinates and red, green and blue colour values). The coordinate (0, 0) corresponds to the top left corner of the image, and the y-values increase as you move down the image.

Note that, if you save a file in .XLS format (Excel 2- 2003 file format) from Genstat and then open it in Excel 2010 or later versions, you will get a warning. Excel 2010 or later versions

always do this when asked to open a file in .XLS format that was not saved by Excel. The warning will say

```
Office has detected a problem with this file. Editing it may harm
your computer. Click for more details.
```

If you click this warning and then click the `Edit Anyway` button, the file will open as expected with no further issues. Saving the file with Excel will stop this happening in the future. However, if you are using Excel 2010 or later versions, it is always best to use the Excel 2007-13 .XLSX file format, and then this will not happen.

The `SHEETNAME` option allows you to specify the name of the sheet to add to an Excel file, rather than using the default `'Genstat data'`. The name should only contain letters, numbers and spaces.

The `METHOD` option controls how `EXPORT` behaves when asked to overwrite an existing file. The available settings are `add`, `append`, `concatenate`, `merge`, `overwrite`, `prompt`, `fail` and `replace`, with a default of `prompt` in interactive mode, and `fail` in batch mode. The following example shows how `METHOD=add` can be used to build up pages in a Genstat workbook file:

```
TEXT   OutFile; VALUE='Results.GWB'
EXPORT [OUTFILE=OutFile; METHOD=overwrite; SHEET='Maximums']\
       MaxLane,MaxLoc,WarpLoc,MaxAmp
EXPORT [OUTFILE=OutFile; METHOD=add; SHEET='Parameters']\
       ParLane,ParLoc,ParSig,ParAmp
EXPORT [OUTFILE=OutFile; METHOD=add; SHEET='Components']\
       PeakLane,PeakLoc,PeakSig,PeakAmp,PeakHt
EXPORT [OUTFILE=OutFile; METHOD=add; SHEET='Lanes']\
       eX,eLane[]
EXPORT [OUTFILE=OutFile; METHOD=add; SHEET='Warping'] eX,W[]
```

The `append` setting allows you to append new values to an existing page. The `GROUPS` option can define a factor column in the output file to identify the blocks of values that are appended; this can be set either to an existing factor (whose identifier will then be used) or a text containing the name to be used for that column in the file. The `GLABEL` option can supply labels for the appended blocks. On the first append, this may be set to two values, where the first value identifies the new (appended) block, and the second identifies the original block of values. The `COLMATCH` option controls whether the columns are matched by name or position. For example:

```
CALCULATE X1,Y1,Z1 = GRNORMAL(1000; 0; 1)
EXPORT    [OUTFILE='Test.gsh'; METHOD=overwrite] X1,Y1,Z1
CALCULATE X2,Y2,Z2 = GRNORMAL(100; 3; 4)
EXPORT    [OUTFILE='Test.gsh'; METHOD=append; GROUPS=Source;\
          GLABEL=!T('Contaminated','Standard')] X2,Y2,Z2
```

If `METHOD=concatenate`, the new data are added as new columns on the right-hand side of an existing page. The new data can also be added as new columns on the right-hand side the page by setting `METHOD=merge`. The `DATA` variables are now merged with the original rows using up to four key columns specified by the `MATCH` and `WITH` options (for the new and original rows, respectively). If `MATCH` is not specified, the first `DATA` variable is used. If `WITH` is not specified, the `MATCH` variables are matched with the same number of initial columns in the page. If a column with the same name already exists in the page when concatenating or merging, the default action is to rename the new column by adding a number to the end of the name to make it unique. Alternatively, if you set option `UPDATE=yes`, the new column will replace the existing column.

If `METHOD=replace`, then for Genstat spreadsheet files or Excel .XLSX files, the sheet may replace an existing sheet within the file. The name of the sheet to be replaced must be supplied by the `SHEETNAME` option. If no matching sheet is found, the sheet will be added to the file.

The `DELETESHEETS` option controls what happens when you are adding data to a file,

containing multiple sheets, that is in a format does not support the updating of one page (e.g. older Excel .XLS, Quattro or Open Office files). The settings are:

`always`	always delete the sheets that are not being updated,
`never`	give a fault if the file contains multiple sheets,
`prompt`	in an interactive run, prompt to check whether the sheets should be deleted, or a fault should be given and the file left unchanged.

In an interactive run the default is `prompt`, and in a batch it is `never`.

The `CSVOPTIONS` option controls aspects of the output to CSV files:

`noquotes`	suppresses the use of quotes around text,
`pack`	removes any spaces around the columns to give a more compact but less readable file,
`round`	rounds numerical values to 6 significant figures,
`fixed`	writes the numerical values without using scientific notation, and
`align`	adds spaces to align the columns to make the file more readable.

The `HTMLOPTIONS` option controls the format of a table written to an HTML file:

`allowformats`	interprets HTML format characters in cells (/& etc) as formats rather than including them as as literal text,
`nogrid`	suppresses the grid between cells,
`centre`	centres the information within each cell, and
`right`	right-justifies the information.

If neither `centre` or `right` are selected, the information in each cell will be left justified.

The `OUTOPTION` option allows extra options to be passed to `Dataload.dll`. See `IMPORT` for details.

The `TABLEFORMAT` option controls how tables with two or more classifying factors are stored in spreadsheet files, with settings:

`page`	to put each table onto a separate page, with the last classifying factor displayed across the columns, and
`column`	to put each table into a single column, so that several tables are displayed on a single page.

The default is `TABLEFORMAT=page`.

The `NONASCII` option specifies how to output non-ASCII characters to a text file: either in UTF-8 format (default), or in Unicode.

The `TIMEOUT` option specifies the number of seconds to wait when a file that needs to be deleted or replaced is open in another process; default 10. This allows time for anti-virus and disk synchronization programs to finish their processing.

The `COLUMNS` parameter can specify names for the columns to be saved. The setting is a text with a single line except for a matrix, where it should have a line for each column and also an extra initial line if the matrix has row labels.

The `PLAINNAMES` option allows you to suppress the additional type information that Genstat adds by default to the column names (`!` for factors, `:D` for dates etc). Alternatively, you can set option `NONAMES=yes` to suppress the names altogether.

The `MISSING` option allows you to specify the string to use to represent a numerical missing value when writing to a text file (`.TXT`, `.TAB` or `.CSV`) or a spreadsheet file (Excel, Quattro or Open Office). If `MISSING` is not set, the string `'*'` is used in `.TXT` and `.TAB` files, while in spreadsheet and csv files the cell is left empty. Missing text values are always output as empty strings.

The `TITLE` option can supply a text containing a title or description of the spreadsheet. This

will be saved in a GSH or GWB file, and will be the heading of an HTML file.

You can set option READONLY=yes to make the entire spreadsheet read-only (so that its contents cannot be changed). Alternatively, you can use the PROTECT parameter to protect any individual column (by making it read-only). Settings of the PROTECT parameter override the setting of the READONLY option.

The ANALYSIS option can supply a text containing Genstat commands to analyse columns in the spreadsheet. The ASETUP option can similarly define commands that should be to be run once before the analysis of the columns, and the ADUMMY option can be used to define the name of the dummy (if any) used in the ANALYSIS commands.

The colours displayed in the cells of a spreadsheet can be controlled by using the FOREGROUND and BACKGROUND parameters to specify the foreground and background colours of the cells in each column. The setting can contain either colour names or RGB values (see the PEN directive for details). You can specify a scalar or a text of length one if all the cells in a column have the same colour. You can specify a variate or text with several values to define different a colour for each cell. Finally you can specify a single pointer to a set of variates or texts if the corresponding DATA setting will need several columns in the spreadsheet. Alternatively, you can specify the background columns for factor columns using the FACCOLOURS parameter. This should be set to a variate or text with the same number of values as the number of levels of the factor. You can apply the colours defined for background of each cell of a factor to the cell's complete row by setting the ROWCOLOURS option to the identifier of the factor. A missing value, empty string or undefined setting for any of these parameters will retain the default colour for the foreground or background.

The DECIMALS parameter allows you to specify the number of decimal places to use for columns. When saving a Genstat or Excel file the columns will be displayed with that precision, but saved with full precision. However, with a text file (.TXT, .TAB or .CSV), the values will be rounded to that number of decimal places, and precision will be thus be lost. The default for text files, when DECIMALS is unset, is to save the data with full precision (15 significant figures).

When DATA contains a pointer, the corresponding COLUMNS setting should be a text of the same length as the pointer. The DECIMALS setting can be either a scalar, or a variate of the same length as the DATA pointer. Similarly, the FACCOLOURS, FOREGROUND and BACKGROUND settings can be either a single variate, scalar or text, or a pointer containing the same number of variates, scalars and/or texts as the length of the DATA pointer. For example:

```
EXPORT [OUT='Test.xls'] !P(U,V),X,!P(Y,Z); \
COLUMNS=!T(A,B),'C',!T(D,E); DECIMALS=!(2,2),3,!(4,5)
```

(Note: EXPORT replaces the procedure %DSAVE from earlier editions of Genstat.)

Options: PRINT, OUTFILE, METHOD, PLAINNAMES, SHEETNAME, NONAMES, TITLE, READONLY, ANALYSIS, ASETUP, ADUMMY, CSVOPTIONS, HTMLOPTIONS, COLMATCH, GROUPS, GLABEL, MATCH, WITH, UPDATE, OUTOPTIONS, ROWCOLOURS, TABLEFORMAT, MISSING, DELETESHEETS, NONASCII.

Parameters: DATA, COLUMNS, PROTECT, FACCOLOURS, FOREGROUND, BACKGROUND, DECIMALS.

Method

The procedure calls the FSPREADSHEET procedure to create a temporary GSH file, which is translated to the required file type using the DATALOAD.DLL library.

Action with RESTRICT

Any restrictions are ignored. However, if the restrictions on the structures are not consistent, a fault will occur.

See also
Procedures: FSPREADSHEET, IMPORT.
Genstat Reference Manual 1 Summary section on: Input and output.

EXTRABINOMIAL

Fits the models of Williams (1982) to overdispersed proportions (M.S. Ridout & P.W. Goedhart).

Options

PRINT = *string tokens*	What to print if iterative estimation process converges successfully and whether to monitor the iterations (model, summary, accumulated, estimates, correlations, fittedvalues, monitoring); default *
CONSTANT = *string token*	How to treat constant (estimate, omit); default esti
FACTORIAL = *scalar*	Limit for expansion of model terms; default 3
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality); default *
METHOD = *string token*	Which model to fit to take account of the extra variation (II, III); default II
MODIFYMODEL = *string token*	Whether to leave the modified MODEL settings (WEIGHTS and DISPERSION) or whether to restore the original situation (yes, no); default no
WEIGHTS = *variate*	To save estimated weights
PHI = *scalar*	To save estimated overdispersion parameter
MAXCYCLE = *scalar*	Maximum number of iterations; default 10
TOLERANCE = *scalar*	Convergence criterion; default 0.01

Parameter

TERMS = *formula*	Model terms to be fitted; if unset it is assumed that the model consists only of a constant term

Description

In binomial regression models, residual variability is often larger than would be expected if the data were indeed binomially distributed. This may be due to a few outliers or a poor choice of link function but often it simply indicates that the data are from a distribution more variable than the binomial. Such data are said to be "overdispersed" or to exhibit "extra-binomial variation".

Williams (1982) discusses two possible models to extend the usual binomial model (Model I). Model II assumes that the true variance exceeds the binomial variance by a factor

$$V = 1 + (\text{NBINOMIAL}-1) \times \varphi \quad (0 \le \varphi \le 1)$$

If the overdispersion parameter *PHI* were known, the data could be analysed using a binomial model with prior weights $1/V$. Procedure EXTRABINOMIAL estimates φ so that the residual chi-square statistic from this weighted analysis is (approximately) equal to the residual degrees of freedom (Moore 1987). If the binomial totals are all equal, Method II is equivalent to setting the DISPERSION option of MODEL equal to the residual chi-square statistic divided by its degrees of freedom.

Alternatively, Model III assumes that the linear predictor varies about its expectation with a constant variance. Usually this variation is assumed to follow a normal distribution; if there is then a logit link, the error distribution will be a logistic normal. Extensions to Model III to have several normal distributions contributing to the variation on the linear predictor, similar to those that occur in stratified analysis of variance, form the basis of many methods suggested for analysing generalized linear mixed models. For Model III, there is generally no simple expression for the exact variance. But the delta method can be used to show that, approximately, the variance exceeds the binomial variance by a factor

$$V = 1 + (\texttt{NBINOMIAL} - 1) \times \varphi \times F^2 / (P \times (1 - P))$$

where φ is variance on the scale of the linear predictor, P is the fitted probability and F is the derivative of the inverse of the link function, evaluated at the fitted value of the linear predictor.

Before using `EXTRABINOMIAL` a `MODEL` statement must be given, in the usual way, to define the y-variate, the binomial totals, the link and any offset. The error distribution must also of course be set to `binomial` but any settings of `WEIGHTS` or `DISPERSION` are ignored.

The form of `EXTRABINOMIAL` is similar in many ways to the `FIT` directive. There is a single parameter `TERMS` to define the model terms to be fitted, and the first four options, `PRINT`, `CONSTANT`, `FACTORIAL`, and `NOMESSAGE`, all have the same syntax and purpose as in `FIT`. The remaining options are specific to `EXTRABINOMIAL`.

The `METHOD` option selects which model to use (`II` or `III`); by default `METHOD=II`. Both models involve the estimation of the weight variate ($1/V$) required to fit the model using the standard Genstat facilities for generalized linear models. If option `MODIFYMODEL=yes`, `EXTRABINOMIAL` will leave the `MODEL` statement in its modified form (provided the iterative estimation of φ converges), with the `WEIGHTS` option set to these weights and the `DISPERSION` option set to 1, so that directives like `DROP` can be used to study the effects of individual terms in the model in the usual way. The `TERMS` directive will also be left set to the model specified by the `TERMS` parameter of `EXTRABINOMIAL`, and this model will be the one most recently fitted, so further output can be obtained using `RDISPLAY`.

Options `WEIGHTS` and `PHI` allow the weights and the estimated value of φ, respectively, to be saved. The `MAXCYCLE` option specifies the maximum number of iterations in the estimation, and the `TOLERANCE` option defines the convergence criterion:

`ABS`(Chi-square − Residual d.f.) < `TOLERANCE` × Residual d.f.

Options: `PRINT`, `CONSTANT`, `FACTORIAL`, `NOMESSAGE`, `METHOD`, `MODIFYMODEL`, `WEIGHTS`, `PHI`, `MAXCYCLE`, `TOLERANCE`.
Parameter: `TERMS`.

Method

If the binomial totals are all equal, φ is determined (non-iteratively) from the residual chi-square statistic.

Otherwise, φ must be found iteratively and the method used (Williams, 1982) involves nested iterations. Each outer iteration (involving a model fit) requires an inner iteration (which uses only `CALCULATE` statements) to get the updated estimate of φ. The option `MAXCYCLE` controls the maximum number of outer iterations. The maximum number of inner iterations is fixed at 10.

Very precise convergence is not important in practice; the default setting of the `TOLERANCE` option (1%) should give a perfectly adequate estimate of φ, usually within 3 iterations.

Action with `RESTRICT`

Any of the following structures may be restricted: the `Y` variate; the `NBINOMIAL` variate; the `WEIGHTS` variate; the `OFFSET` variate; any variate or factor appearing in the model formula. Restrictions on different structures must be compatible. Restricted units are excluded from the analysis.

References

Moore, D.F. (1987). Modelling the extraneous variance in the presence of extra-binomial variation. *Applied Statistics*, **36**, 8-14.

Williams, D.A. (1982). Extra-binomial variation in logistic linear models. *Applied Statistics*, **31**, 144-148.

See also
Procedures: GLMM, HGANALYSE, RNEGBINOMIAL, R0INFLATED.
Genstat Reference Manual 1 Summary section on: Regression analysis.

FACAMEND

Permutes the levels and labels of a factor (J.T.N.M. Thissen).

Option

DIRECTION = *string token*	Order into which to sort the levels or labels of FACTOR (ascending, descending); default asce

Parameters

FACTOR = *factor*	Factor whose levels or labels are to be permuted
NEWLEVELS = *variate* or *text*	To specify the new order of the factor levels or labels

Description

Occasionally a clear presentation of results requires permutation of factor levels. For example, describing the interaction between two qualitative factors you may want to change the order of the factor levels to clarify the interaction structure.

The factor whose levels are to be permuted must be specified using the FACTOR parameter. The new order of the factor levels can be specified using the NEWLEVELS parameter, either by way of the numerical levels (in a variate) or by the labels (in a text).

If the NEWLEVELS parameter is not specified and the factor has labels, the levels are sorted so that these are in alphabetic order. If NEWLEVELS is not set and there are no labels, the levels of the factor are put into numerical order. The DIRECTION option determines whether the sort is into ascending or descending order; by default DIRECTION=ascending.

Option: DIRECTION. Parameters: FACTOR, NEWLEVELS.

Method

The procedure uses the SORT directive and a new factor declaration to change the order of the factor levels or labels.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedures: FACEXCLUDEUNUSED, FACLEVSTANDARDIZE, FACSORT, FACUNIQUE, FDISTINCTFACTORS.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FACCOMBINATIONS

Forms a factor to indicate observations with identical values of a set of variates, texts or factors (R.W. Payne).

Options

FLABELS = *string token*	When to form labels (always, ifredeclared, only, never); default ifre
SEPARATOR = *text*	Separator to use when constructing labels; default ' '
ISEPARATOR = *text*	Separator to use between identifiers and levels or labels; default ' '
IMETHOD = *string token*	Whether to include identifiers in the labels (include, omit); default omit

Parameters

VECTORS = *pointers*	Pointers containing sets of vectors (variates, and/or factors, and/or texts)
FACTOR = *factors*	Saves a factor for each set of vectors with a level for every different combination of their values

Description

FACCOMBINATIONS forms a factor whose levels identify the units that share the same combinations of values of a set of vectors (i.e. variates, factors or texts). The vectors are specified, in a pointer, by the VECTORS parameter, and the factor to be formed is specified by the FACTOR parameter.

This may be useful, for example, in regression analyses if you want to assess the lack of fit of a particular model. Suppose you have a multiple linear regression with explanatory variates X1, X2 and X3. If the data set contains units that have identical values for X1, X2 and X3, we can use these to obtain an estimate of the true residual variation, which can then be compared with the lack of fit of the model. If we put

```
FACCOMBINATIONS !p(X1,X2,X3); FACTOR=X123
```

the factor X123 will have a level for every combination of X1, X2 and X3 values that occurs in the data set. The residual sum of squares is then given by the sum of squares within the levels of X123, and the difference between the residual sum of squares of the model and the X123 sum of squares represents the lack of fit. (FACCOMBINATIONS is used in exactly this way within procedure FITINDIVIDUALLY.)

The FLABELS option controls whether labels are formed for the FACTOR, with settings:

always	labels are always formed,
ifredeclared	labels are formed only if the new factor has not been declared already with the correct number of levels (default),
only	only labels are formed (i.e. with this setting the factor is not given any values), and
never	labels are never formed.

The labels are constructed by listing the values of the original factors. The IMETHOD option controls whether the identifiers of the vectors are included too (each one before its values); by default they are excluded. The SEPARATOR option specifies the string to use to separate each identifier (if present) and value from the next, and the ISEPARATOR option specifies the string to use to separate the identifiers from the values; by default a single space is used for both of these.

Options: FLABELS, SEPARATOR, ISEPARATOR, IMETHOD.
Parameters: VECTORS, FACTOR.

Action with RESTRICT
If any of the vectors is restricted, the values of the factor will be formed only for the units not excluded by the restriction.

See also
Procedures: AFUNITS, FACDIVIDE, FACPRODUCT, FBASICCONTRASTS, FDISTINCTFACTORS.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Design of experiments.

FACDIVIDE

Represents a factor by factorial combinations of a set of factors (R.W. Payne).

Option

OLDFACTOR = *factor*	Factor whose levels are to be represented by the factorial combinations of the NEWFACTORs

Parameters

NEWFACTOR = *factors*	Factors formed to represent OLDFACTOR
LEVELS = *scalars* or *variates*	Levels of the NEWFACTORs

Description

FACDIVIDE allows a set of factors to be formed with a set of levels for every level of another factor. (It is thus provides the opposite operation to the procedure FACPRODUCT, which forms a factor with a level for every combination of the levels of a set of factors.) FACDIVIDE may be useful, for example, if a design for a single factor, such as a Latin square, is to be used to study several factors and their interactions: e.g. a 12 by 12 Latin square could be used to study the main effects and interaction of factors A and B with 3 and 4 levels respectively.

The original factor is specified by the OLDFACTOR option, and the NEWFACTOR and LEVELS parameters specify the new factors and their levels. So, to represent the 12-level factor Treat by factors A and B as above, would require

```
FACDIVIDE [OLDFACTOR=Treat] NEWFACTOR=A,B; LEVELS=3,4
```

As in the FACTOR directive, the LEVELS parameter can be set to a scalar if the new factor is to have the standard levels 1, 2 and so on, or to a variate if you want to specify some other levels.

Options: OLDFACTOR. Parameters: NEWFACTOR, LEVELS.

Method

FACDIVIDE uses GENERATE to form a set of mapping vectors containing the levels of the new factors, in standard order. It then uses these in the NEWLEVELS function to calculate the levels of the new factors.

Action with RESTRICT

If any OLDFACTOR is restricted, the levels will be formed only for the units not excluded by the restriction.

See also

Procedures: AFUNITS, FACPRODUCT, FACCOMBINATIONS, FBASICCONTRASTS, FDISTINCTFACTORS.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Design of experiments.

FACEXCLUDEUNUSED

Redefines the levels and labels of a factor to exclude those that are unused (R.W. Payne).

No options

Parameters

FACTOR = *factors*	Factors with unused levels
NEWFACTOR = *factors*	New factors, with levels (and labels) that exclude those that are unused; if unset, the original factor is redefined

Description

FACEXCLUDEUNUSED allows you to redefine a factor to remove levels that do not occur within its current values. This can be useful, for example, if you want to produce tables that are not cluttered with empty rows or columns.

The factor is specified by the FACTOR parameter, and a new factor (excluding the unused levels) can be saved by the NEWFACTOR parameter. If NEWFACTOR is not set, the original factor is redefined with the new levels (and labels, if any).

Options: none.
Parameters: FACTOR, NEWFACTOR.

Action with RESTRICT

Any restrictions are ignored.

See also

Procedures: FACAMEND, FACLEVSTANDARDIZE. FACSORT. FACUNIQUE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FACGETLABELS

Obtains the labels for a factor if it has been defined with labels, or constructs labels from its levels otherwise (V.M. Cave).

Options

PRINT = *string token*	Controls printed output (labels); default *
PREFIX = *text*	Supplies a single line of text to be used as a prefix when constructing labels from the factor levels; default * i.e. none

Parameters

FACTOR = *factors*	Factor whose labels are to be obtained
LABELS =*texts*	Specifies text structures to save the labels of each factor
EXIST = *scalars*	Specifies a scalar for each factor, set to the value 1 if its labels already existed or 0 if they had to be constructed

Description

FACGETLABELS can be used to obtain the labels of a factor. If the factor has been defined without any labels, they are constructed from its levels.

The factor whose labels are to obtained (or constructed) is specified using the FACTOR parameter. The LABELS parameter can be used to save the labels in a text structure. The EXIST parameter can save a scalar that indicates whether the labels already existed (= 1), or had to be constructed (= 0).

The PRINT option can be set to labels to print the resulting labels; by default, nothing is printed. The PREFIX option can supply a text containing a single string, to be used as a prefix when constructing labels from factor levels. The string is inserted before the level numbers to form the labels. For factors with defined labels, PREFIX is ignored.

Options: PRINT, PREFIX.
Parameters: FACTOR, LABELS, EXIST.

Method

FACGETLABELS uses GETATTRIBUTE to get the necessary attributes. If there are no labels, TXCONSTRUCT is used to form them from the levels and any prefix.

Action with RESTRICT

Any restrictions on the factors supplied by FACTOR are ignored.

See also

Directive: GETATTRIBUTE, TXCONSTRUCT.

FACLEVSTANDARDIZE

Redefines a list of factors to coordinate their levels or labels (R.W. Payne).

Options

FREPRESENTATION = *string token*	Whether to coordinate the factors to have the same levels, labels or (ordinal) number of levels (`levels`, `labels`, `ordinals`); default `leve`
DIRECTION = *string token*	How to sort the levels or labels (`ascending`, `descending`, `given`); default `asce`
CASE = *string token*	Case to use for labels (`given`, `lower`, `upper`, `sentence`, `title`); default `give`
REMOVEUNUSED = *string token*	Whether to remove unused levels (`yes`, `no`); default `no`

Parameters

FACTOR = *factors*	Factors to be coordinated
NEWFACTOR = *factors*	New factors, redefined to share the same levels or labels; if unset, the original FACTOR is redefined

Description

FACLEVSTANDARDIZE allows you to redefine a set of factors so that they have the same set of levels or labels. The original factors are listed by the FACTOR parameter, while the NEWFACTOR parameter saves the redefined factors. If no NEWFACTOR is defined for one of the factors in the FACTOR list, the original factor itself is redefined.

The FREPRESENTATION option controls whether it is the levels or labels that must be the same for the redefined factors. Alternatively, if you set FREPRESENTATION=ordinals, the levels and labels are ignored and the factors are simply redefined so that their numbers of levels become identical. By default, FREPRESENTATION=levels.

The DIRECTION option controls whether the set of levels or labels for the redefined factors is sorted into ascending or descending order, or whether they are kept in the order in which they are met in the FACTOR list.

The CASE option enables you to change the case of letters in the labels. The default setting, given, leaves the labels unchanged. So, for example, 'Control' and 'control' would be treated as two different labels. The available settings are:

given	leaves the case of each letter exactly as given in the string;
upper	changes all letters to upper case (or capitals);
lower	changes all letters to lower case;
sentence	puts the first character in the text (if a letter) into upper case, then uses upper case only at the start of each new sentence;
title	begins each new word with a capital letter, but otherwise uses lower case.

The REMOVEUNUSED option allows you to remove any levels that are defined for a factor, but are not present in its values.

Options: FREPRESENTATION, DIRECTION, CASE, REMOVEUNUSED.
Parameters: FACTOR, NEWFACTOR.

Method

FACLEVSTANDARDIZE uses the SETCALCULATE directive to form the combined set of levels or labels, SORT (if necessary) to sort them, and EQUATE to transfer the values from the original to the redefined factors (matching their levels or labels as required). The cases of the labels are

changed using TXCONSTRUCT.

Action with RESTRICT
Any restrictions are ignored.

See also
Procedures: FACAMEND, FACEXCLUDEUNUSED, FACSORT, FACUNIQUE, FDISTINCTFACTORS.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FACMERGE

Merges levels of factors (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (summary); default * i.e. none
OLDFACTOR = *factor*	Original factor
NEWFACTOR = *factor*	New factor with merged levels

Parameters

MERGE = *variates* or *texts*	Levels to merge
LEVMERGED = *variates*	Level to assign to the merged levels
LABMERGED = *texts*	Label to assign to the merged levels

Description

FACMERGE allows you to merge several levels of a factor into a single level in a new factor. The original factor is specified using the OLDFACTOR option, and the new factor is saved using the NEWFACTOR option.

The levels to merge are specified by the MERGE parameter, and can be identified either by their levels or by their labels. The LEVMERGED parameter can supply a level to use for the merged levels; if this is not specified, the first level of the original factor, in the list of merged levels, is used. Similarly, the LABMERGED parameter can supply a level to use for the merged levels; if this is not specified, the label corresponding to the first merged level is used.

Printed output is controlled by the PRINT option, using the following settings:

summary	summary of merges.

By default, nothing is printed.

Options: PRINT, OLDFACTOR, NEWFACTOR.
Parameters: MERGE, LEVMERGED, LABMERGED.

Action with RESTRICT

The merging process ignores any restrictions but, when this has been completed, any restriction on OLDFACTOR is applied to NEWFACTOR.

See also

Procedure: FACAMEND.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FACPRODUCT

Forms a factor with a level for every combination of other factors (R.W. Payne).

Options

FLABELS = *string token*	When to form labels (always, ifredeclared, only, never); default ifre
SEPARATOR = *text*	Separator to use when constructing labels; default ' '
LMETHOD = *string token*	Whether to define levels for all combinations or only for those present in the data (all, present); default pres
ISEPARATOR = *text*	separator to use between identifiers and levels or labels; default ' '
IMETHOD = *string token*	Whether to include identifiers in the labels (include, omit); default omit

Parameters

FACTORS = *pointers* or *formulae*	Factors contributing to each product
PRODUCT = *factors*	Factors to be formed

Description

This procedure allows a factor to be formed whose levels represent all the combinations of a list of other factors. This may be useful, for example, if a design is generated by regarding a set of the treatments as though they were the factorial combinations of a list of factors in order to confound some of the contrasts, say, with blocks. It may then be very much easier to set up the levels of the factors in the list rather than those of the full treatment factor (which can then be formed by this procedure). Similarly, as shown in the example, it can be used to put the values in a multi-way table back into a variate, inserting the value in each cell of the table into the units with that level of the classifying factors.

The FACTORS parameter gives the list of factors from which the new factor is to be formed. These factors can be input in either a pointer or a model formula. The PRODUCT parameter specifies the identifier of the new factor.

The FLABELS option controls whether labels are formed for the new factor, with settings:

always	labels are always formed,
ifredeclared	labels are formed only if the new factor has not been declared already with the correct number of levels (default),
only	only labels are formed (i.e. with this setting the factor is not given any values), and
never	labels are never formed.

The labels are constructed by listing the levels (or labels, if available) of the original factors. The IMETHOD option controls whether the identifiers of the factors are included too, each one before its levels (or labels); by default they are excluded. The SEPARATOR option specifies the string to use to separate each identifier (if present) and level/label from the next, and the ISEPARATOR option specifies the string to use to separate the identifiers from the levels/labels; by default a single space is used for both of these.

Usually the PRODUCT factor has levels defined only for the combinations of levels of the factors that actually occur in the values that are formed. However, you can set option LMETHOD=all to request that there is a level for every combination (and this is the default when FLABELS=only).

Options: FLABELS, SEPARATOR, LMETHOD, ISEPARATOR, IMETHOD.
Parameters: FACTORS, PRODUCT.

Method

The FCLASSIFICATION directive is used, if necessary, to form lists of factors whose product is to be calculated. The levels for the factor are calculated according to the formula

$$level = 1 + \sum_{i=1...p} \{ (m_i - 1) \times n_{i+1} \times ... \times n_p \}$$

where p is the number of factors in the list, m_i is the ordinal level of factor i, and n_i is the number of levels of factor i (the ordinal levels for factor i are the numbers $1...n_i$). If LMETHOD=present, the levels are then renumbered to omit any that do not occur in the data.

Action with RESTRICT

If any of the factors is restricted, the levels will be formed only for the units not excluded by the restriction.

See also

Procedures: AFUNITS, FACCOMBINATIONS, FACDIVIDE, FBASICCONTRASTS, FDISTINCTFACTORS.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Design of experiments.

FACSORT

Sorts the levels of a factor according to an index vector (R.W. Payne).

Options

DIRECTION = *string token*	Direction in which to sort the index (`ascending`, `descending`); default `asce`
SETATTRIBUTES = *string tokens*	Which aspects of each NEWFACTOR to define (`levels`, `labels`, `values`); default `*` i.e. labels and values if defined for FACTOR, also levels if not the integers 1,2...

Parameters

FACTOR = *factors*	Factors whose levels are to be reordered
INDEX = *variate*, *text* or *one-way table*	Index vectors defining the ordering of the levels of each factor
NEWFACTOR = *factors*	New factors with reordered levels; if unset, the original FACTOR is redefined
NEWLEVELS = *variates*	Saves the (reordered) levels as defined for each NEWFACTOR

Description

This procedure reorders the levels of a factor. The factor is specified by the FACTOR parameter. The NEWFACTOR parameter can specify the identifier for the new reordered factor (so that FACTOR is left unchanged). If this is not supplied, the original FACTOR is redefined with its levels in the new order.

The order is defined by an index vector, specified by the INDEX parameter. This can be a variate, or a text, or a one-way table, whose number of values is equal to the number of levels of the factor. The levels are thus sorted in parallel with the INDEX (using the SORT directive), and the DIRECTION option indicates whether this is to be into ascending or descending order.

The SETATTRIBUTES option specifies which of the labels, levels and values to define for the NEWFACTOR. If SETATTRIBUTES is not set, the default is to define whichever of these has been defined for the FACTOR. In particular, note that levels are then not defined if the FACTOR levels are simply 1,2...

The NEWLEVELS parameter can specify a variate to save the levels of the NEWFACTOR. You can use this as the setting of the OLDPOSITIONS and NEWPOSITIONS parameters of the COMBINE directive in order to reorder tables classified by the factor. Or, you can discover the (ordinal) number of the original level corresponding to each new level by

```
GETATTRIBUTE [ATTRIBUTE=levels] FACTOR; SAVE=FacLev
CALCULATE OldLevelNumber\
          = POSITION(NEWLEVELS; FacLev['levels'])
```

Options: DIRECTION, SETATTRIBUTES.
Parameters: FACTOR, INDEX, NEWFACTOR, NEWLEVELS.

Method

FACSORT uses the standard Genstat manipulation commands, such as SORT.

Action with RESTRICT

Any restrictions are ignored.

See also

Procedures: FACAMEND, FACEXCLUDEUNUSED, FACLEVSTANDARDIZE, FACUNIQUE, FDISTINCTFACTORS.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FACUNIQUE

Redefines a factor so that its levels and labels are unique (R.W. Payne).

Options

MERGESAME = *string tokens*	What must be the same for groups defined by the factor to be merged (levels, labels); default * i.e. no groups are merged
INCREMENT = *scalar*	Value to use to modify duplicate levels; default * i.e. a suitable (small) value is determined automatically
ADDTO = *string token*	Whether to add the increment to the value or the absolute value of duplicated levels (value, absolutevalue); default abso

Parameters

OLDFACTOR = *factors*	Factors whose levels and labels are to be made unique
NEWFACTOR = *factors*	New factors with unique levels; if unset, the original OLDFACTOR is redefined
CHANGED = *scalars*	Indicates whether the factor has changed

Description

FACUNIQUE allows you to correct mistakes in the definition of a factor that may have caused the same level number or the same label to occur more than once. Genstat does not fault duplicate levels or labels as there are some occasions when this may be deliberate. For example, you might want to use the same labelling for more than one line of a table, and might therefore have that label repeated in the factor classifying its rows.

The factors to correct are listed by the OLDFACTOR parameter, and the NEWFACTOR parameter can save the redefined factors. If no NEWFACTOR is defined for one of the factors in the OLDFACTOR list, the original factor itself is redefined. The CHANGED parameter can save a scalar that set to one if changes were needed, or otherwise set to zero.

By default, FACUNIQUE appends the characters '_1', '_2' and so on to each duplicate label. The default for levels is that it adds a small increment to each zero or positive duplicate level, and subtracts that increment from each negative duplicate level. If you would prefer to add the increment to both positive and negative levels, you can set option ADDTO=value. (This indicates that the increment is to be added to the value, rather than the absolute value of the duplicate level.) The default increment is taken as the largest power of 10 that is small enough to modify each duplicate level while preserving their numerical order. So, for example, if you had levels 1, 2, 3, 2, 5 and 2, the increment would be 0.1; the second instance of 2 would become 2.1, and the third would become 2.2. As another example, if the levels were 0.1, 0.2, 0.3, 0.2, 0,5 and 0.2, the increment would be 0.01; the second instance of 0.2 would then become 0.21, and the third would become 0.22. Alternatively, the INCREMENT option allows you to supply your own increment.

The MERGESAME option allows you to merge some of the groups that are defined by the factor. If you set MERGESAME=levels, any groups that have the same level will be merged (but any duplicate labels will still be made unique). If you set MERGESAME=labels, any groups that have the same label will be merged (but any duplicate levels will still be made unique). Finally, if you set MERGESAME=labels,levels, only groups that have the same level and label will be merged (but any remaining duplicate levels or labels will be made unique).

Options: MERGESAME, INCREMENT, ADDTO.
Parameters: OLDFACTOR, NEWFACTOR, CHANGED.

Method
The unique levels and labels are obtained using procedure FUNIQUEVALUES.

Action with RESTRICT
Any restrictions are ignored.

See also
Procedures: FACAMEND, FACEXCLUDEUNUSED, FACLEVSTANDARDIZE, FACSORT, FUNIQUEVALUES, FDISTINCTFACTORS.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FALIASTERMS

Forms information about aliased model terms in analysis of variance (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (aovtable, aliasedterms); default alia
TREATMENTSTRUCTURE = *formula*	Treatment model for the design; if this is not set, the default is taken from any existing setting defined by the TREATMENTSTRUCTURE directive
BLOCKSTRUCTURE = *formula*	Block model for the design; if this is not set, the default is taken from any existing setting defined by the BLOCKSTRUCTURE directive
FACTORIAL = *scalar*	Limit on number of factors in a treatment term; default 3
RESTRICTION = *variate*	Defines a restriction on the units for the analysis; default * i.e. none

Parameters

TERMS = *formula*	Model terms whose aliased terms are to be identified; the default is to take all the terms in the treatment model
ALIASTERMS = *formula* or *pointer*	Saves the aliased terms

Description

When a term is aliased in an analysis of variance, it is listed in the *Information summary* produced by

```
ANOVA [PRINT=information]
```

under the heading "Aliased model terms". However, ANOVA does not indicate the terms with which it is aliased. This information can be obtained using procedure FALIASTERMS.

The block structure of the design can be specified by the BLOCKSTRUCTURE option. If this is not set, the default is taken from any existing setting defined by the BLOCKSTRUCTURE directive. Similarly, the treatments are specified by the TREATMENTSTRUCTURE option, with a default taken from any existing setting defined by the TREATMENTSTRUCTURE directive. As in ANOVA, the FACTORIAL option sets a limit on the number of factors in a treatment term; default 3. The RESTRICTION option can supply a variate to define a restriction on the units of the design. (Units where the variate contains a zero value are excluded from the analysis.)

The terms whose aliasing is to be investigated are specified by the TERMS parameter. If this is not specified, all the terms in the treatment model are investigated.

The terms to which they are aliased can be saved using the ALIASEDTERMS parameter. They are saved in a formula structure if a single term has been specified by TERMS, or in a pointer containing a formula structure for each term if it has specified several.

Printed output is controlled by the PRINT option with settings:

aovtable	to print the analysis-of-variance table; and
aliasterms	to print the terms to which each term is aliased (default).

Options: PRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, FACTORIAL, RESTRICTION.
Parameters: TERM, ALIASTERMS.

Method

The procedure calculates a set of dummy effects for the aliased model term, and then forms and analyses a variate in which only these effects are present. The analysis detects the model terms

to which the term is aliased as those that have non-zero sums of squares.

Action with `RESTRICT`
A restriction can be specified using the `RESTRICTION` option.

See also
Directives: `ANOVA`, `FPSEUDOFACTORS`.
Procedures: `AEFFICIENCY`, `ALIAS`.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, Design of experiments.

FBASICCONTRASTS

Breaks a model term down into its basic contrasts (R.W. Payne).

Options

TERM = *formula*	Model term to split into basic contrasts
PSEUDOFACTORS = *pointer*	Pseudo-factors representing the basic contrasts
NEWTERMS = *formula structure*	Model formula containing the term followed by the pseudofactors

No parameters

Description

It is well known that the interaction between factors F1, F2 ... Fm (each with prime p numbers of levels) can be partitioned into (p−1)**(m−1) orthogonal sets of contrasts, known as basic contrasts (e.g. Kempthorne 1952, page 321). They are usually written as

F1**k_1 F2**k_2 ... Fn**k_n

where conventionally k_1=1 and $1 \le k_i < p$. Each set of basic contrasts represents comparisons between p sets of factor combinations, the ith of which contains the factor combinations f_1 ... f_m such that

$$k_1 \times f_1 + k_2 \times f_2 + \ldots k_m \times f_m = i \bmod(p)$$

In most straightforward experimental designs, all the contrasts of each treatment interaction are estimated in a single stratum. When this is not feasible, a popular strategy is form the design so that different sets of basic contrasts are estimated in different strata. This underlies the design key method, which is used by the GENERATE directive and the AKEY procedure.

When a design key has been used to generate the design, the FPSEUDOFACTORS directive may be used to form the pseudo-factors required to cope with interactions (or other model terms) whose contrasts are estimated in different strata. If this pseudo-factoring is not included, the ANOVA directive will warn that the design contains partial confounding.

When no design key is available, provided all the treatment terms are main effects or interactions all of whose factors have the same prime number of levels, an alternative strategy is to use the FBASICCONTRASTS procedure to break up each partially-confounded interaction into its sets of basic contrasts. The interaction is specified using the TERM option. The PSEUDOFACTORS option saves a pointer containing the factors generated to represent the basic contrasts. Finally, the NEWTERMS option can save a new model formula containing the interaction followed by the pseudo-factor operator // and then the list of pseudo-factors. For example, for the interaction of two 3-level factors A and B, the NEWTERMS formula would be

```
A.B // (Pf[1,2])
```

where Pf[] is the pointer of pseudo-factors.

Options: TERM, PSEUDOFACTORS, NEWTERMS.
Parameters: none.

Method

FBASICCONTRASTS uses the PRIMEPOWER procedure to check that all factors have a prime number of levels. It then constructs a design key with a row for every set of basic contrasts, and forms the pseudo-factors using GENERATE.

See also

Procedures: AFUNITS, FACCOMBINATIONS, FACDIVIDE, FACPRODUCT.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Design of experiments.

FBETWEENGROUPVECTORS

Forms variates and classifying factors containing within-group summaries to use e.g. in a between-group analysis (R.W. Payne).

Options

CLASSIFICATION = *factors*	Factors defining the groups; must be set
COUNTS = *variate*	Saves a variate counting the number of units with each factor combination; default *
WEIGHTS = *variate*	Weights to be used to calculate the within-group summaries; default * indicates that all units have weight 1
METHOD = *string token*	How to summarize the data variates (totals, nobservations, means, minima, maxima, variances, quantiles, sds, skewness, kurtosis, semeans, seskewness, sekurtosis); default mean
PERCENTQUANTILES = *scalar*	Percentage point for quantiles; default 50
OMITEMPTYCELLS = *string token*	Whether to omit units arising from empty cells in the summary table (yes, no); default no
SETLEVELS = *string token*	Whether to redefine the levels of factors (yes, no); default no

Parameters

VECTOR = *variates* and *factors*	Original data vectors
NEWVECTOR = *variates* and *factors*	New vectors containing the within-group summaries

Description

FBETWEENGROUPVECTORS is useful when you have replicated observations on a set of groups. It allows you to form variates, with a unit for each group, containing a summary of the observations within that group. You can also form factors, again with a unit for each group, to define the characteristics of the groups. You can then use these to perform a between-group analysis, with one of the variates acting as the response variate, and the factors and other variates defining the model to be fitted.

The factors defining the groups are specified by the CLASSIFICATION option. The METHOD option specifies how to form summaries of the variates, and you can specify weights by using the WEIGHTS option. For more information about the summaries, see the TABULATE directive, which is used to do the calculations. Factors are summarized by taking the level that occurs most frequently within each group, using the TABMODE procedure.

The PERCENTQUANTILES option specifies the percentage point to use for quantiles. The default is 50 (i.e. the median).

The OMITEMPTYCELLS option indicates whether to omit units arising from empty cells in the summary table. By default, these are included.

The SETLEVELS option allows you to redefine the levels of the new factors, to exclude any that do not occur in the data set. By default, they are not redefined.

The VECTOR parameter specifies the variates and factors to be summarized, and the NEWVECTOR parameter saves the variates and factors containing the summaries. You can also use the COUNTS to save a variate containing counts of the number of units in each group if WEIGHTS is unset, or the sum of their weights if it is set.

Options: CLASSIFICATION, COUNTS, WEIGHTS, METHOD, PERCENTQUANTILES, OMITEMPTYCELLS, SETLEVELS.
Parameters: OLDVECTOR, NEWVECTOR.

Action with `RESTRICT`
If any `VECTOR`, or the `WEIGHTS` variate, or any of the classifying factors is restricted, the summaries will be form using only the restricted subset of units. If more than one variate or factor is restricted, the restrictions must be the same.

See also
Directive: `EQUATE`.
Procedures: `FWITHINTERMS`, `UNSTACK`.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FCOMPLEMENT

Forms the complement of an incomplete block design (W. van den Berg).

Option

PRINT = *string token*	Controls whether or not to print a plan of the design (design); default desi

Parameters

TREATMENTS = *factors*	Specifies the treatment factor of the original design
REPLICATES = *factors*	Specifies the replicate factor of the original design when this is a resolvable design
BLOCKS = *factors*	Specifies the block factor of the original design
NEWTREATMENTS = *factors*	Saves the treatment factor of the complement design
NEWREPLICATES = *factors*	Saves the replicate factor of the complement design when this is a resolvable design
NEWBLOCKS = *factors*	Saves the block factor of the complement design
NEWUNITS = *factors*	Saves the treatment factor of the complement design
SEED = *scalars*	Seed for the random-numbers to randomize the design; default 0

Description

An incomplete-block design is a design that is set out in blocks that do not each contain a plot with every treatment. Examples within Genstat include balanced-incomplete-block designs (see procedure AGBIB), cyclic designs (see AFCYCLIC and AGCYCLIC), and alpha designs (see AFALPHA and AGALPHA). The complement design is formed by taking each block of the incomplete-block design and replacing its treatments by those not in the block.

This extends Genstat's repertoire of designs. For example AGCYCLIC provides a cyclic design for 7 treatments levels, 7 blocks and 3 units per block, but not one with 7 treatments levels, 7 blocks and 4 units per block. However, the second design is the complement of the first design, and can thus be constructed using FCOMPLEMENT. One useful feature is that, if the original design is balanced, the complement design will also be balanced.

The block and treatment factors of the original design are specified by the TREATMENTS and BLOCKS parameters. The corresponding factors of the complement design can be saved with the NEWBLOCKS and NEWTREATMENTS parameters. If the original design is a resolvable design (i.e. one where the blocks can be grouped together into replicates), you can use the REPLICATES parameters to supply the replicate factor for the original design, and the NEWREPLICATES parameter to save the replicate factor for the complement design.

The SEED parameter allows you to specify a seed for the random numbers that are used by the RANDOMIZE directive, inside FCOMPLEMENT, to randomize the design. The default of zero continues the existing sequence of random numbers if RANDOMIZE has already been used in the current Genstat job. If RANDOMIZE has not yet been used, Genstat picks a seed at random. Note that you can set SEED=-1 if you want to avoid any randomization.

By default a plan of the complement design is printed, but you can set option PRINT=* to suppress this.

Option: PRINT.

Parameters: TREATMENTS, REPLICATES, BLOCKS, NEWTREATMENTS, NEWREPLICATES, NEWBLOCKS, SEED.

Method
From a variate containing all the treatment levels, a subset is made for each block, discarding the levels present in the original design. These are then appended together to form the full design.

See also
Procedures: AFALPHA, AFCYCLIC, AGALPHA, AGBIB, AGCYCLIC.
Genstat Reference Manual 1 Summary section on: Design of experiments.

FCONTRASTS

Modifies a model formula to contain contrasts of factors (R.W. Payne).

Options

FORMULA = *formula*	Formula to modify to contain contrasts
NEWFORMULA = *formula structure*	Modified formula; if unset, the modified formula replaces FORMULA
FACTORIAL = *scalar*	Limit on the number of variates or factors in terms generated from FORMULA; default 3

Parameters

FACTOR = *factors*	Factors over which to define contrasts
CONTRASTTYPE = *string tokens*	Type of contrast (polynomial, regression); default poly
ORDER = *scalars*	Number of contrasts to define for each FACTOR
XCONTRASTS = *variates* or *matrices*	X-values defining the contrasts for each FACTOR
DEVIATIONS = *string tokens*	Whether to include deviations (yes, no); default no
ORTHOGONALIZE = *string tokens*	Whether to orthogonalize the contrasts (yes, no); default no
SAVECONTRASTS = *pointers*	Pointer to save the contrast variates defined for each FACTOR

Description

FCONTRASTS provides a way of fitting contrasts amongst the levels of factors in analyses such as REML that currently do not allow the use of the Genstat contrast functions like POL and REG. To fit contrasts here, you need to fit x-variates that represent the contrasts explicitly during the analysis. So, if you wanted to fit linear and quadratic contrasts of a factor N, you would need to calculate variates Nlin and Nquad, representing the linear and quadratic contrasts of N, and fit those in the analysis instead of N: i.e. instead of setting option

```
FIXED=POL(N; 2)
```

in the VCOMPONENTS directive, you need to calculate Nlin and Nquad by

```
CALCULATE Nlin = N
& Nquad = Nlin * Nlin
```

and then set

```
FIXED=Nlin+Nquad
```

FCONTRASTS allows you to form the contrast variates and include them in a formula automatically. The formula to be modified is specified, as a formula data structure, by the FORMULA option. The new formula is saved by the NEWFORMULA option. If this is not set, the new formula replaces the original formula specified by the FORMULA option. The formula needs to be expanded into its individual model terms in order to make the modification. The FACTORIAL option sets a limit on the number of factors or variates in the terms that are formed.

The FACTOR parameter specifies the factors that are to be replaced in the formula by contrasts. The CONTRASTTYPE parameter specified the type of contrast and the ORDER parameter specifies the number of contrasts to be generated for each factor. For polynomial contrasts, the XCONTRASTS parameter can specify the x-value to be used for each level of the factor; if this is not set, the factor's existing levels are used.

The DEVIATIONS parameter can be set to yes to include contrasts to represent the deviations from the fitted contrasts that are required to explain all the effects of the factor. These are orthogonalized to the other contrasts and so, if only the main effect of the factor is fitted, the estimates other contrasts will be unaffected by the fitting of the deviations. This

orthogonalization may not work for interactions between the factor and other factors. So, if you find that the deviations are not needed in the model, you should then refit the model without the deviations in order to get the estimates of the contrasts themselves. (Of course, if you find that the deviations are needed, then the contrasts are not really of any interest.) You can set the parameter `ORTHOGONALIZE=yes` to orthogonalize the contrasts themselves. Again, this orthogonalization may work only for the main effect of the factor, and not for interactions with other factors.

The contrasts variates are defined with extra texts, using the `EXTRA` parameter of the `VARIATE` directive, which give the name of the factor and of the individual contrast (e.g. `lin` or `quad`). The `IPRINT` option is set to `extra` at the same time so that these texts are used to label the contrasts in the analysis. If you want to use the contrast variates in any other way, you need to save them a pointer, which can be supplied by the `SAVECONTRASTS` parameter.

Options: `FORMULA`, `NEWFORMULA`, `FACTORIAL`.
Parameters: `FACTOR`, `CONTRASTTYPE`, `ORDER`, `XCONTRASTS`, `DEVIATIONS`, `ORTHOGONALIZE`, `SAVECONTRASTS`.

Action with `RESTRICT`
`FCONTRASTS` takes account of any restrictions on the factors or variates in the `FORMULA`.

See also
Directives: `TERMS`, `TREATMENTSTRUCTURE`, `VCOMPONENTS`.
Procedures: `RCOMPARISONS`, `RTCOMPARISONS`, `VTCOMPARISONS`.
Functions: `COMPARISON`, `POL`, `POLND`, `REG`, `REGND`.
Genstat Reference Manual 1 Summary sections on: Regression analysis, Analysis of variance, REML analysis of linear mixed models.

FCORRELATION

Forms the correlation matrix for a list of variates (R.W. Payne).

Options

PRINT = *string tokens*	Printed output (correlations, test); default corr
METHOD = *string token*	Type of test to make (against zero) for the correlations (twosided, greater, lessthan); default twos
WEIGHTS = *variate*	Provides weights for the units of the variates; default * assumes that they all have weight one
CORRELATIONS = *symmetric matrix*	Saves the correlations
PROBABILITIES = *symmetric matrix*	Saves the test probabilities
NOBSERVATIONS = *scalars*	Saves the number of observations from which the correlations have been calculated

Parameter

DATA = *variates*	Variates for which the matrix is to be calculated

Method

FCORRELATION forms the correlation matrix for a set of variates which should be listed by the DATA parameter. The WEIGHTS option can provide a variate of weights for the units of the variates; by default these are all assumed to have weight one.

Printed output is controlled by the PRINT option with settings:

correlations	prints the correlation matrix;
tests	prints tests for the correlations.

By default PRINT=correlation. The METHOD option indicates the type of test to be done, with settings:

twosided	for a two-sided test of the null hupothesis that that the correlation is zero;
greaterthan	for a one-sided test of the null hypothesis that the correlation is not greater than zero;
lessthan	for a one-sided test of the null hypothesis that the correlation is not less than zero.

Tests cannot be produced if there are fewer than two observations.

The correlation matrix can be saved using the CORRELATIONS option, the (symmetric) matrix of test probabilities can be saved using the PROBABILITIES option, and the number of observations upon which it is based can be saved using NOBSERVATIONS option.

Options: PRINT, METHOD, WEIGHTS, CORRELATIONS, PROBABILITIES, NOBSERVATIONS.
Parameter: DATA.

Method

A SSPM structure is set up for the DATA variates, and its values are formed using the FSSPM directive. The corrected sums of squares and products are divided by the residual degrees of freedom to give the variance covariance matrix, and the CORRMAT function is used to convert this to a correlation matrix. The test probabilities are obtained using the PRCORRELATION procedure.

Action with RESTRICT

FCORRELATION takes account of restrictions on any of the variates.

See also

Directive: CORRELATE.

Procedures: DCORRELATION, FVCOVARIANCE, PARTIALCORRELATIONS, PRCORRELATION, SCORRELATION.

Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Calculations and manipulation.

FDESIGNFILE

Forms a backing-store file of information for AGDESIGN (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (catalogue, data, filestructure); default * i.e. none

Parameters

DATAFILE = *texts*	Name of the data file containing the information required to form each backing-store subfile
BSFILE = *texts*	Name of the backing-store file
SUBFILE = *identifiers*	Identifier of the backing-store subfile

Description

Procedure AGDESIGN generates the factors and pseudo-factors required to define a generally balanced design. To do this it uses a backing-store subfile that contains the repertoire of available designs, together with the information required to form them.

FDESIGNFILE can be used to form this subfile. The DATAFILE parameter supplies the name of a file containing the necessary information, and the BSFILE and SUBFILE parameters specify the name of the backing-store file and subfile where it is to be stored. Details of the format and contents of the data file can be obtained by setting option PRINT=filestructure. PRINT also has a setting catalogue which produces a catalogue of the backing-store file, and a setting data which prints the lines of data as they are read.

Option: PRINT. Parameters: DATAFILE, BSFILE, SUBFILE.

Method

FDESIGNFILE uses the standard Genstat directives for input and output.

See also

Procedure: AGDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

FDIALLEL

Forms the components of a diallel model for REML or regression (R.W. Payne).

No options

Parameters

MALEPARENTS = *factors*	Specifies the male parents
FEMALEPARENTS = *factors*	Specifies the female parents
PARENTS = *matrices*	Saves design matrices for the overall parental effects
COMPPARENTS = *matrices*	Saves comparison matrices for overall parental effects
PUREVSCROSS = *factors*	Saves factors to represent the comparison between pure and crossed lines
CROSSPAIR = *factors*	Saves factors to represent the comparison between types of pairs of parent (ignoring the individual genders)

Description

FDIALLEL forms the factors and matrices that are needed to specify and fit a diallel model using Genstat REML or regression.

The factors identifying the male and female parent of each line are specified by the MALEPARENTS and FEMALEPARENTS parameters, respectively. The PARENTS parameter saves a design matrix that can be used in REML to represent the overall effects of each parental line, and the COMPPARENTS parameter saves the transpose of the matrix. You can use COMPPARENTS as the third argument of the COMPARISON function to fit the parental effects in a Genstat regression model. The PUREVSCROSS parameter saves a factor to represent the comparison between pure and crossed lines, and the CROSSPAIR parameter saves a factor representing the comparison between types of pairs of parent (ignoring their individual genders).

The examples for FDIALLEL (which can be accessed by using the LIBEXAMPLE procedure or the Examples menu in Genstat *for Windows*) show how these factors and matrices can be used in Genstat REML and regression to generate the analyses of Hayman (1954) and Jones (1965), provided by the DIALLEL procedure. The terms in the DIALLEL analysis correspond to those in the FDIALLEL analysis as follows.

- a: variation between mean effects of each parental line; this corresponds to PARENTS in REML, or COMP(Vdum; np; COMPPARENTS) in regression (where vdum is a dummy variate, containing any values, and np is the number of different types of parental line).
- b1: assesses whether dominance is largely uni-directional; corresponds to PUREVSCROSS.
- b2: estimates "asymmetry" i.e. if alleles at any one locus are not equally frequent; corresponds to PARENTS.PUREVSCROSS in REML, or COMP(Vdum; np; COMPPARENTS).PUREVSCROSS in regression.
- b3: signifies that some dominance is peculiar to individual crosses; corresponds to CROSSPAIR.
- c: variation between average maternal effects of each parental line; corresponds to FEMALEPARENT.
- d: variation in the reciprocal differences not attributable to c; corresponds to MALEPARENT.FEMALEPARENT.

Options: none.

Parameters: MALEPARENTS, FEMALEPARENTS, PARENTS, COMPPARENTS, PUREVSCROSS, CROSSPAIR.

References

Hayman, B.I. (1954). The Analysis of Variance of Diallel Tables. *Biometrics*, **10**, 235-244.
Jones, R.M. (1965). Analysis of Variance of the Half Diallel Table. *Heredity*, **20**, 117-121.

Action with `RESTRICT`

`FDIALLEL` ignores restrictions i.e. it forms the factors and matrices using all the units of `MALEPARENTS` and `FEMALEPARENTS`.

See also

Procedures: `DIALLEL`, `FCONTRASTS`.
Genstat Reference Manual 1 Summary sections on: Regression analysis, REML analysis of linear mixed models.

FDISTINCTFACTORS

Checks sets of factors to remove any that define duplicate classifications (R.W. Payne).

No options

Parameters

SET1 = *pointers*	First set of factors
SET2 = *pointers*	Second set of factors
DISTINCTSET = *pointers*	Saves the distinct factors

Description

FDISTINCTFACTORS checks sets of factors to remove any that divide the data units into identical groups. The levels of the factors need not be in the same order – it is the composition of the groups that they define that is important. Also, any null groups (containing no units) are ignored.

The SET1 and SET2 parameters supply pointers containing sets of factors. The DISTINCTSET parameter saves the set of distinct factors (i.e. those that all define different groupings). If only SET1 is set, DISTINCTSET saves the set of factors from SET1 that are distinct from each other. Alternatively, if both SET1 and SET2 are set, DISTINCTSET saves the factors in SET1, plus the factors in SET2 that are distinct from each other and from the factors in SET1. Thus, if SET2 is specified, it is assumed the factors in SET1 are already distinct from each other (so this provides a way of augmenting an already distinct set).

Options: none.
Parameters: SET1, SET2, DISTINCTSET.

Action with RESTRICT

Any restrictions are ignored.

See also

Directives: FACTOR, SETCALCULATE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FDRBONFERRONI

Estimates false discovery rates by a Bonferroni-type procedure (A.I. Glaser).

Options

PRINT = *string token*	Controls printed output (pi0); default pi0
METHOD = *string token*	Controls the method used for calculating π_0 (smoother, bootstrap); default smoo
LOGP = *string token*	Whether to take logs of π_0 when METHOD=smoother (yes, no); default no
DF = *scalar*	Degrees of freedom for smoothing spline; default 3
PLOT = *string token*	Controls plots (phistogram, qhistogram, pi0vslambda, qvsp, tests, expfalsepositive, inference, loginference); default phis, qhis, pi0v, qvsp, test, expf, infe, logi
WINDOW = *scalar*	Window for the graphs; default 1
KEYWINDOW = *scalar*	Window for the key (zero for none); default 2

Parameters

PROBABILITIES = *variates*	Significance values, must lie between 0 and 1
LAMBDA = *scalars* or *variates*	Values of tuning parameter λ, equivalent to significance levels at which to test the PROBABILITIES; default !(0, 0.05....0.9)
FDR = *variates*	Saves the False Discovery Rates (i.e. *q*-values) at the sorted *p*-values in PROBABILITIES
FRR = *variates*	Saves the False Rejection Rates at the sorted *p*-values in PROBABILITIES
POWER = *variates*	Saves the power estimates as a function of the sorted *p*-values in PROBABILITIES
PI0 = *scalars*	Saves the value of π_0, i.e. the maximum value of the FDR
LOWER = *scalars*	Lower bound of *q*-values to use with PLOT settings qvsp, tests and expfalsepositive; default 0
UPPER = *scalar*	Upper bound of *q*-values to to use with PLOT settings qvsp, tests and expfalsepositive; default 1, which indicates maximum *q*-value

Description

When testing m multiple hypotheses there are various outcomes that can occur, summarized in the table below

Decision on null hypothesis:	Accept	Reject	Total
Situation: null true	U	V	m_0
alternative true	T	S	m_1
Total	W	R	m

where R is the total number of rejected hypotheses and $W = m - R$. The proportion of tests that are truly null, π_0, is m_0 divided by m. The false discovery rate (FDR), also known as the *q*-value of a test, is a commonly used error measure in multiple-hypotheses, defined as

$$\text{FDR} = \text{E}(V/R \mid R > 0) \times \Pr(R > 0)$$

i.e. the expected proportion of false positives findings among all the rejected hypotheses multiplied by the probability of making at least one rejection; the FDR is zero when $R = 0$. Similarly the false rejection rate (FRR) is defined as

$$\text{FDR} = \text{E}(T/W \mid W > 0) \times \Pr(W > 0),$$

i.e. the expected proportion of false negatives findings among all the accepted hypotheses times the probability of accepting at least one test. We also define the power to be equal to $\text{E}(S/m_1 \mid m_1 > 0) \times \Pr(m_1 > 0)$.

The *p*-values from the multiple hypotheses are supplied, in a variate, using the PROBABILITIES parameter. The analysis uses a Bonferroni-type multiple-testing procedure to calculate the corresponding *q*-values. The *p*-values are assumed independent, or may be weakly dependent if there are many of them. The parameter π_0 is calculated using the method of Storey (2002). This involves a tuning parameter λ, which can be set using the LAMBDA parameter; the default is a variate containing the numbers 0, 0.05, ... 0.9. Λ can be thought of as the value beyond which the individual *p*-values are considered null. As λ gets larger the bias of π_0 gets smaller, but its variance increases. If you set LAMBDA to a scalar, π_0 is estimated by dividing the number of null tests (i.e. the number of *p*-values greater than λ) by the expected number of null tests $m \times (1 - \Lambda)$. If you set LAMBDA to a variate with several values, two methods are available, selected by the following settings of the METHOD option:

smoother	fits a smoothing spline of λ onto initial estimates of π_0 calculated as for a single λ value, and takes the estimate of π_0 as the value corresponding to the largest value of λ;
bootstrap	estimates π_0 by bootstrap sampling from the variate of *p*-values.

The default is smoother, as the bootstrap method may be time-consuming when there are many *p*-values. The number of degrees of freedom to use in the smoothing is specified by the DF option (default 3). Also, you can set option LOGP=yes to do the smoothing on log-transformed π_0 values.

The PRINT option controls printed output, using the settings:

pi0	prints π_0, the estimate of the expected proportion of null *p*-values corresponding to the largest *q*-value.

By default PRINT=pi0.

Various graphs can be selected by the following settings of the PLOT option:

phistogram	histogram of *p*-values;
qhistogram	histogram of *q*-values (i.e. the FDR values);
pi0vslambda	λ against π_0 (when only one value of λ is specified the default values of LAMBDA are used);
qvsp	*q*-values against *p*-values;
tests	plot of the sorted tests against *q*-values;
expfalsepositive	plot of the number of expected false positives against the sorted tests;
inference	plot of the FDR, FRR and power statistics against the sorted *p*-values; and
loginference	plots the FDR, FRR and power statistics on log scales, against the sorted *p*-values restricted to $p<0.5$, with a background grid to enable estimates to be read for specific probability values. Due to the small numbers used in this plot the *p*-values and FDR, FRR & power statistics are displayed as percentages.

By default all the plots are produced. The WINDOW option specifies the window for the graphs, and the KEYWINDOW option species the window for keys.

Options: PRINT, METHOD, LOGP, ROBUST, DF, PLOT, WINDOW, KEYWINDOW.
Parameters: PROBABILITIES, LAMBDA, FDR, FRR, POWER, PI0, LOWER, UPPER.

Method
FDRBONFERRONI uses the method of Storey (2002), with the definitions of FRR and power given in Genovese & Wasserman (2002).

Action with RESTRICT
The PROBABILITIES parameter can be restricted. All output estimates will then be based only on those unrestricted units.

References
Storey, J.D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society Series B*, **64**, 479-498.
Genovese, C. & Wasserman, L. (2002). Operating characteristics and extensions of the false discovery rate procedure. *Journal of the Royal Statistical Society Series B*, **64**, 499-518.

See also
Procedures: FDRMIXTURE, AMCOMPARISON, QTHRESHOLD.
Genstat Reference Manual 1 Summary section on: Microarray data.

FDRMIXTURE

Estimates false discovery rates using mixture distributions (J.W. McNicol & D.B. Baird).

Options

`PRINT` = *string token*	What to print (`monitoring`, `estimates`); default `esti`
`DISTRIBUTION` = *string token*	Which distribution to mix with Uniform (`beta`, `gamma`); default `beta`
`INITIAL` = *variate*	Initial values for mixing proportion (φ) and Beta or Gamma parameters (*A* and *B*); default `!(0.90, 0.30, 2)`
`LOWER` = *variate*	Lower limits for parameters; default `!(0.00001, 0.001, 0.001)`
`UPPER` = *variate*	Upper values for parameters; default `!(0.99999, 5, 1000)`
`PLOT` = *string token*	What to plot (`histogram`, `density`, `logdensity`, `inference`, `loginference`); default `hist`, `dens`, `logd`, `infe`, `logi`
`WINDOW` = *scalar*	Window for graphs; default 1
`KEYWINDOW` = *scalar*	Key window for Inference plot; default 2
`MAXCYCLE` = *scalar*	Maximum iteration cycles; default 50
`TOLERANCE` = *scalar* or *variate*	Tolerance for convergence of parameters; default 0.01 for Beta, and 0.001 for Gamma

Parameters

`PROBABILITIES` = *variates*	Significance values, must lie between 0 and 1
`ESTIMATES` = *variates*	Saves the estimates of mixture parameters φ, *A* and *B*
`FDR` = *variates*	Saves the False Discovery Rates at the *p*-values in `PROBABILITIES` i.e. *q*-values
`FRR` = *variates*	Saves the False Rejection Rates at the *p*-values in `PROBABILITIES`
`POWER` = *variates*	Saves the power estimates as a function of the *p*-values in `PROBABILITIES`
`POSTHA` = *variates*	Saves the Posterior Probability of H_a at the *p*-values in `PROBABILITIES`
`LOGLIKELIHOOD` = *scalars*	Value of the loglikelihood at end of the iteration process
`NCYCLES` = *scalars*	Number of iterations taken to convergence

Description

`FDRMIXTURE` estimates the false discovery rate (FDR), false rejection rate (FRR) and power of a test by modelling significance values as a 2-component mixture of Uniform and Beta or Gamma densities, Allison *et al.* (2002). The context is multiple testing, with data from any situation where the same simple hypothesis, H_0, is tested many times, such as in transcriptomics (microarrays), metabolomics and proteomics. These tests generate a large number of significance values which, under H_0, have a Uniform distribution and, under H_a, can be modelled as a Beta or truncated Gamma density. `FDRMIXTURE` estimates the parameters of this mixture distribution to derive the False Discovery Rate, Prob(H_0/D_a), the False Rejection Rate, Prob(H_a/D_0) and the Power of the test, Prob(D_a/H_a), each as a function of p_{crit}. Here Da denotes the event "p<$p$crit". The procedure also calculates the posterior probability of Ha, Prob(Ha/p), (POSTHa) from the mixture distribution. The significance values are provided by the `PROBABILITIES` parameter and the choice of distribution (Beta or Gamma) by the `DISTRIBUTION` option. The `FDR`, `FRR`, `POWER` and `POSTHA` parameters return estimates at the corresponding values of `PROBABILITIES`. Thus `FDR` contains the q-values of Storey & Tibshirani (2003). An EM

algorithm is used to estimate the mixture parameters which are returned in the parameter ESTIMATES.

The mixture model parameterization takes a proportion φ from the Uniform distribution, and (1 – φ) from either a Beta or Gamma distribution. The Gamma parameterization is

$$f(x) = (1/b)^A / \text{Gamma}(A) \times \exp(-x/B) \times x^{(A-1)}$$

truncated at x=1, and the Beta parameterization is

$$f(x) = x^{(A-1)} \times (1-x)^{(B-1)} / \text{Beta}(A; B).$$

Details of the estimation process are returned in the parameters NCYCLES and LOGLIKELIHOOD. Initial values, lower and upper mixture parameter limits are set by the INITIAL, LOWER and UPPER options. Convergence can be controlled by a single tolerance for all three parameters or for each parameter separately using the TOLERANCE option, and the number of iterations by the MAXCYCLE option. A warning is printed when the parameter estimates imply a Beta or Gamma density which is unimodal rather than reverse J-shaped. The former would give rise to situations where $Pr(H_0/D_a) > Pr(H_a/D_a)$ for very small *p*.

Printed output is controlled by the PRINT option with settings:

estimates	for estimates of the parameters, and
monitoring	for monitoring information from the fitting process.

By default PRINT=estimates.

Graphical output is controlled by the PLOT option with settings:

histogram	for a plot of the fitted mixture against the histogram of probabilities,
density	for a plot of the fitted mixture against the kernel density estimate of the probabilities on a logit scale (this allows a more detailed comparison at small probability value),
logdensity	gives even greater detail, by putting the density on a log scale (note that greater variation is expected around small density values on the log scale),
inference	generates a plot of FDR, FRR and POWER against *p*, and
loginference	plots these statistics on log scales, restricted to $p<0.5$, with a background grid, to enable estimates to be read for specific probability values.

By default all the plots are produced.

The WINDOW option controls where the plots go and the KEYWINDOW option can be used to position the key in the inference plots.

Options: PRINT, DISTRIBUTION, LOWER, UPPER, PLOT, WINDOW, KEYWINDOW, MAXCYCLE, TOLERANCE.

Parameters: PROBABILITIES, ESTIMATES, FDR, FRR, POWER, POSTHA, LOGLIKELIHOOD, NCYCLES.

Method

In the context of hypothesis testing the false discovery rate, FDR, can be defined as the probability of H_0 being true when the result of the statistical test leads us to accept H_a:

$$\text{FDR} = \text{Prob}(H_0/D_a).$$

By Bayes theorem

$$\text{Prob}(H_0/D_a) = \text{Prob}(D_a/H_0) \times \text{Prob}(H_0) / \text{Prob}(D_a)$$

and

$$\text{Prob}(D_a) = \text{Prob}(D_a/H_0) \times \text{Prob}(H_0) + \text{Prob}(D_a/H_a) \times \text{Prob}(H_a).$$

Further, in the context of multiple testing, where there are many *p*-values available, all the terms in these expressions can be derived by modelling the *p*-values as a 2-component mixture distribution. The *p*-values, under H_0, have a Uniform density and, under H_a, can be modelled as

a Beta or truncated Gamma density. The mixing proportions are Prob(H_0) and Prob(H_a) respectively. Prob(D_a/H_a) is CLBETA(*p*; *A*; *B*) or CLGAMMA(*p*; *A*; *B*). The False Rejection Rate,

FRR = Prob(H_a/Do)

is derived similarly. The posterior probability of H_a,

$PostH_a$ = Prob(H_a/p)

= Prob(p/H_a) × Prob(H_a) / (Prob(p/H_a) × Prob(H_a) + Prob(p/H_0) × Prob(H_0))

and each term is estimated by the mixture model parameters.

Action with RESTRICT

The PROBABILITIES parameter can be restricted. All output estimates will then be based only on those unrestricted units.

References

Allison, D.B., Gadbury. G.L., Heo, M., Fernandez, J.R., Lee, C.-K., Prolla, T.A., & Weindruch R. (2002). A mixture model approach for the analysis of microarray gene expression data. *Computational Statistics & Data Analysis*, **39**, 1-16.

Storey J.D. & Tibshirani R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Science*, **100**, 9440-9445.

See also

Procedures: FDRBONFERRONI, QTHRESHOLD.

Genstat Reference Manual 1 Summary section on: Microarray data.

FEXACT2X2

Does Fisher's exact test for 2×2 tables (M.S. Ridout & M.W. Patefield).

Option

PRINT = *string tokens*	Controls printed output (probabilities, tables); default prob

Parameters

TABLE = *tables* or *variates*	The numbers in each 2×2 table, ordered row by row or column by column
PROBABILITIES = *variates*	Saves the probabilities for each table in a variate of length 6 (to store in positions 1, 3 and 5 one-tailed, two-tailed calculated as twice the one-tailed probability, and as the sum of the probabilities of all tables with probability less than that of the observed table with the corresponding mid-p values stored in positions 2, 4 and 6)

Description

The ostensibly simple problem of testing for association in a 2×2 contingency table has generated a large and disputative literature. Yates (1984) and Hirji, Tan & Elashoff (1991) give excellent reviews. Controversy has centred on Fisher's exact test which conditions on both margins of the 2×2 table; some have argued that this conditioning is appropriate only if both margins of the table are fixed by the sampling design whereas others, Yates included, advocate use of the test irrespective of the sampling design. Consensus of opinion seems to favour the latter viewpoint.

Procedure FEXACT2X2 does the calculations for Fisher's exact test. The TABLE parameter is used to supply to the procedure the four numbers that comprise the 2×2 table, either as a 2×2 Genstat table, with no margins, or as a variate consisting of the four numbers ordered either row by row or column by column.

The procedure calculates the one-tailed significance level that is produced by the exact test. The mid-p value, which includes only half the probability of the observed table, is also calculated. See Hirji, Tan & Elashoff (1991) for a discussion of mid-p values. Several methods have been proposed for calculating a two-tailed significance level, two of which are implemented in the procedure. The first method simply doubles the one-tailed significance level whereas the second method calculates the cumulative probability of all outcomes that are no more probable than the observed table. See Yates (1984) for discussion of these and other methods. The procedure also calculates mid-p values corresponding to each of the two-tailed significance levels. The various probabilities can be saved, in a variate of length six, using the PROBABILITIES parameter.

The procedure has a single option PRINT to control printed output. By default PRINT=probabilities. There is also another setting tables which causes the procedure to display all 2×2 tables with margins that are the same as the observed table together with their probabilities of occurrence under the null hypothesis of no association and the cumulative probabilities calculated from both tails. This display was proposed by Hill (1984).

Option: PRINT.
Parameters: TABLE, PROBABILITIES.

Method
The procedure evaluates all 2×2 tables with the same margins as the observed table. The hypergeometric probabilities that are calculated as inversely proportional to the product of factorials of the table elements and then standardized to sum to one.

Action with `RESTRICT`
If the values of the 2×2 table are specified as a variate, this must not be restricted.

References
Hirji, K.F., Tan, S. & Elashoff, R.M. (1991). A quasi-exact test for comparing two binomial proportions. *Statistics in Medicine*, **10**, 1137-1153.
Yates, F. (1984). Tests of significance for 2x2 contingency tables. *Journal of the Royal Statistical Society, Series A*, **147**, 426-463.
Hill, I.D. (1984). Contribution to the discussion of Yates (1984). *Journal of the Royal Statistical Society, Series A*, **147**, 452-453.

See also
Procedures: `CHIPERMTEST`, `APERMTEST`, `RPERMTEST`.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

FFRAME

Forms multiple windows in a plot-matrix for high-resolution graphics (P.W. Goedhart).

Options

PRINT = *string tokens*	Whether to display the layout and numbering of the plot-matrix in a table or in a high-resolution test-graph on the current device (table, testgraph); default *
ARRANGEMENT = *string token*	Type of plot-matrix (rectangle, square, lowersymmetric, uppersymmetric, diagonal); default rectangle
ROWS = *scalar*	Number of rows of plot-matrix; default 3
COLUMNS = *scalar*	Number of columns of plot-matrix; default 3
DIAGONALWINDOWS = *string token*	Whether to include or exclude the diagonal in symmetric plot-matrices (include, exclude); default include
SQUARESHAPES = *string token*	Whether to force the individual windows, excluding margins for annotation, to be square (yes, no); default no
STARTWINDOW = *scalar*	Specifies the number of the first window; default 1
TESTGRAPH = *variate*	Specifies windows to be displayed in a test-graph (if this option is set, only a test-graph is produced and all other settings are ignored); default *
NUMBERING = *string token*	Controls the way in which the individual windows are numbered (rowwise, columnwise); default rowwise
DEFINE = *string token*	Whether to define the windows within the procedure (windows, nothing); default wind
CLEARWINDOW = *scalar* or *variate*	Defines the windows for which the screen should be cleared; i.e. specifies the elements of the SCREEN pointer which are set to the single-values text 'clear', other element of SCREEN are set to 'keep'; default 1
RLOWER = *scalar*	Lowest y device coordinate; default 0
RUPPER = *scalar*	Highest y device coordinate; default 1
CLOWER = *scalar*	Lowest x device coordinate; default 0
CUPPER = *scalar*	Highest x device coordinate; default 1
RSKIP = *scalar*	Space between windows along the y-axis; default 0
CSKIP = *scalar*	Space between windows along the x-axis; default 0
MARGIN = *string tokens*	Sets the size of the margins for labels and titles (xtitle, ytitle, none, small); default *
YMLOWER = *scalar*	Size of bottom margin (x-axis labelling) in each window; default *
YMUPPER = *scalar*	Size of upper margin (overall title) in each window; default *
XMLOWER = *scalar*	Size of left-hand margin (y-axis labelling) in each window; default *
XMUPPER = *scalar*	Size of right-hand margin in each window; default *
RMLOWER = *scalar*	Additional size of bottom margin (x-axis labelling) in windows at the bottom of the plot-matrix; default 0
RMUPPER = *scalar*	Additional size of upper margin (overall title) in windows at the top of the plot-matrix; default 0
CMLOWER = *scalar*	Additional size of left-hand margin (y-axis labelling) windows at the left of the plot-matrix; default 0
CMUPPER = *scalar*	Additional size of right-hand margin in windows at the

	right of the plot-matrix; default 0
BACKGROUND = *text* or *scalar*	Specifies the colour to be used for the background in each window (where allowed by the graphics device); default 'background'

Parameters

NGRAPHS = *scalar*	To save the number of windows in the plot-matrix
SWINDOW = *pointer*	Pointer to save scalars with window numbers
SYLOWER = *pointer*	Pointer to save scalars with lower y device coordinates for each window
SYUPPER = *pointer*	Pointer to save scalars with upper y device coordinates for each window
SXLOWER = *pointer*	Pointer to save scalars with lower x device coordinates for each window
SXUPPER = *pointer*	Pointer to save scalars with upper x device coordinates for each window
SSCREEN = *pointer*	Pointer to save single-valued texts with value 'clear' or 'keep'; this depends only on the setting of the CLEARWINDOW option
SMYLOWER = *pointer*	Pointer to save scalars with size of bottom margins for each window
SMYUPPER = *pointer*	Pointer to save scalars with size of upper margins for each window
SMXLOWER = *pointer*	Pointer to save scalars with size of left-hand margin for each window
SMXUPPER = *pointer*	Pointer to save scalars with size of right-hand margin for each window

Description

Procedure FFRAME supplements the FRAME directive with automatic definition of windows in a so-called plot-matrix. The ARRANGEMENT option defines the arrangement of the plot-matrix which can be in either a rectangle, square, lowersymmetric, uppersymmetric or diagonal. The number of rows and columns of the plot-matrix can be specified by options ROWS and COLUMNS. The COLUMNS option is only relevant when ARRANGEMENT=rectangle. The DIAGONALWINDOWS option defines whether or not the diagonal windows should be included for symmetric plot-matrices. The option setting SQUARESHAPES=yes forces each window, excluding margins for annotation, to be square.

By default the positions of the windows in the plot-matrix are defined within the procedure by means of a FRAME statement using windows 1, 2 ... Alternatively, you can use the STARTWINDOW option to start the window numbering at a different value. The windows are numbered by rows unless the NUMBERING option is set to columnwise. When the number of windows is larger than the maximum allowable number, the windows are not defined and a warning message is printed. In that case the parameters SYLOWER, SYUPPER, SXLOWER and SXUPPER can be used to save the y and x device coordinates of the windows for subsequent use in a FRAME statement. Likewise parameters SMYLOWER, SMYUPPER, SMXLOWER and SMXUPPER can be used to save the margins for each plot. Also, setting DEFINE=nothing does not define the windows. The NGRAPHS parameter saves the number of windows in the plot-matrix, and the SWINDOW parameter saves the window numbers.

Typically the screen should be cleared only for the first window in a plot-matrix. The SSCREEN parameter can be used to save the single-valued text 'clear' for the first window and 'keep' for all other windows. These texts can then be used to set the SCREEN option of the plot

directives as is shown in the example. The CLEARWINDOW option can be employed in case the screen should be cleared for other windows.

By default the unit plot-square [0,1] x [0,1] is employed. Options RLOWER, RUPPER, CLOWER and CUPPER can be used to define a different plot-square. The RSKIP and CSKIP options can be used to increase the space between windows which is by default 0. The user must ensure that these options are set to sensible values.

The space used for labelling of axis and an overall title for each window can be controlled with options MARGIN, YMLOWER, YMUPPER, XMLOWER and XMUPPER options. Default values of these options ensure that an overall title and labels (not longer than 4 characters) along both axes are displayed when using the standard character size. However, the default values normally prohibit the display of titles along the axes. The settings xtitle and/or ytitle of the MARGIN option generate space for titles along the axes. The MARGIN option can have the following settings:

MARGIN=*	YMLOW,YMUP,XMLOW,XMUP	= 0.04, 0.04, 0.05, 0.01
MARGIN=xtit	YMLOW,YMUP,XMLOW,XMUP	= 0.09, 0.04, 0.05, 0.01
MARGIN=ytit	YMLOW,YMUP,XMLOW,XMUP	= 0.04, 0.04, 0.10, 0.01
MARGIN=xtit,ytit	YMLOW,YMUP,XMLOW,XMUP	= 0.09, 0.04, 0.10, 0.01
MARGIN=none	YMLOW,YMUP,XMLOW,XMUP	= 0
MARGIN=small	YMLOW,YMUP,XMLOW,XMUP	= 0.015

These values can be overridden by setting options YMLOWER, YMUPPER, XMLOWER, XMUPPER explicitly. The outer margins of the border plots in the plot-matrix
can be increased by specifying options RMLOWER, CMLOWER, RMUPPER and RMLOWER. For example, you could expand these, as needed for a trellis-style plot, by

```
FFRAME [ROWS=nwrows; COLUMNS=nwcols; MARGIN=none;\
       CMLOWER=0.04; YMLOWER=0.09; RMUPPER=0.04]
```

The background colour for all windows may be modified by the BACKGROUND option.

The PRINT option can be used to display the layout and numbering of the plot-matrix in a table or in a high-resolution test-graph on the current device. A test-graph can also be requested by setting the TESTGRAPH option to a variate with the window numbers to be displayed; all other settings are then ignored.

Options: PRINT, ARRANGEMENT, ROWS, COLUMNS, DIAGONALWINDOWS, SQUARESHAPES, STARTWINDOW, TESTGRAPH, NUMBERING, DEFINE, CLEARWINDOW, RLOWER, RUPPER, CLOWER, CUPPER, RSKIP, CSKIP, MARGIN, YMLOWER, YMUPPER, XMLOWER, XMUPPER, RMLOWER, RMUPPER, CMLOWER, MUPPER, BACKGROUND.

Parameters: NGRAPHS, SWINDOW, SYLOWER, SYUPPER, SXLOWER, SXUPPER, SSCREEN SMYLOWER, SMYUPPER, SMXLOWER, SMXUPPER.

Method

The relevant part of the calculations for the lower and upper x device coordinates are

```
CALCULATE xrange = (CUPPER - CLOWER + CSKIP)/COLUMNS - CSKIP
CALCULATE XLOWER = CLOWER + (!(1...#COLUMNS) - 1)\
                   * (xrange + CSKIP)
CALCULATE XUPPER = XLOWER + xrange
```

The y device coordinates are calculated similarly.

Action with RESTRICT

Restrictions on CLEARWINDOW and TESTGRAPH are ignored.

See also
Directive: FRAME.
Genstat Reference Manual 1 Summary section on: Graphics.

FFREERESPONSEFACTOR

Forms multiple-response factors from free-response data (R.W. Payne).

Options

MRESPONSE = *pointer*	Pointer with a factor for each RESPONSECODE, indicating which of the DATA texts contain that response
RESPONSECODES = *text*	Specifies the codes to look for in the DATA texts
LABELCODES = *text*	Strings to label the factors within the MRESPONSE pointer; default RESPONSECODES
DUPLICATECODES = *factor*	Defines groupings of duplicate or alternative codes within the RESPONSECODES text
EXCLUDENULL = *string token*	Whether to exclude the factor recording which DATA contain none of the RESPONSECODES (yes, no); default no
SUFFIXNULL = *scalars*	Suffix to use to represent the null factor in MRESPONSE; default 0
LABELNULL = *text*	Label to use to represent a the null factor in MRESPONSE; default 'none'
DATAFORMAT = *string token*	Whether the data for the respondents is given line-by-line within the DATA text(s) or whether there is a separate text for each respondent (linebyline, textbytext); default line
CASE = *string token*	Whether to treat the case of letters (small or capital) as significant when searching for the codes (significant, ignored); default igno
MULTISPACES = *string token*	Whether to treat differences between multiple spaces and single spaces as significant, or to treat them all like a single space (significant, ignored); default igno
DISTINCT = *string tokens*	Whether to require each RESPONSECODE to have one or more separators to its left or right within each DATA text (left, right); default left, righ
SEPARATOR = *text*	Characters to use as separators; default ' ,;:.'

Parameter

DATA = *texts*	Information from the respondents

Description

Multiple responses occur in surveys as the result of open-ended questions like "Which plants grow in your garden" or "What languages do you speak?". Results from a multi-response question are represented in Genstat by a pointer containing a factor for each possible response. Each factor has two levels (numbered 0 and 1, and labelled 'absent' and 'present') to indicate where the corresponding response was recorded.

The data for FFREERESPONSEFACTOR are specified using the DATA parameter. If option DATAFORMAT=linebyline (the default), the data are specified in a single text with a line for each respondent. You can also give a second text, again with a line for each respondent, if you cannot fit all the information for any of the respondents into a single line. Likewise you can specify a third text if you need more than two lines, and so on. Alternatively, if option DATAFORMAT=textbytext, you supply the data for each respondent in a separate text. The information consists of free-form text in which the responses of interest are to be found. For example, in a survey of garden plants, the text might contain the lines

```
'I grow carrots, cabbages and lettuces. I also have an apple
```

```
tree.'
```

The codes to find within the texts are supplied, in a text, by the RESPONSECODES option. If you want to supply alternative codes (for example, synonyms or singlar and plural codes), you should put all the alternatives into the RESPONSECODES text, and set the DUPLICATECODES option to a factor to indicate how the codes are grouped together. For example, Codes below contains singular and plural codes for various plants, and Alternatives indicates how these belong together

```
TEXT [VALUES=carrot,cabbage,lettuce,potato,tomato,\
     carrots,cabbages,lettuces,potatoes,tomatoes,\
     apple,rose,magnolia,sycamore,'silver birch',\
     apples,roses,magnolias,sycamores,'silver birches'] Codes
FACTOR [LEVELS=10; VALUES=(1...5)2,(6...10)2] Alternatives
```

The pointer of multiple-response factors is saved using the MRESPONSE option. By default, the pointer is labelled by the names of the codes (or by the first of each set of codes if there are alternatives). However, you can use the LABELCODES option to supply other labels if these are unsuitable (e.g. too long).

The EXCLUDENULL option controls whether or not the pointer contains a factor to make an explicit record of the people that gave none of the codes (default 'no'). This will be needed if the later tables are to contain a line for "no response". The SUFFIXNULL option specifies the suffix to be used for this factor in the pointer while, the LABELNULL option specifies its label.

FFREERESPONSEFACTOR usually ignores the case of letters (small or capital) when looking for the codes. So for example 'Apple' would be the same as 'apple'. However, you can set option CASE=significant to recognise these differences in case. FFREERESPONSEFACTOR usually also treats multiple spaces as the same as a single space, but you can set option MULTISPACE=significant to treat these differences as important.

By default, FFREERESPONSEFACTOR requires each code to begin either at the start of the DATA text or to be preceded in the text by a separator (such as a space or comma). Similarly, it requires each code to end within the text with a separator (or to be at the end of the text). This is requested by the DISTINCT option, with its default DISTINCT=left,right. However, for example, you can set DISTINCT=left if the codes must be separated from other text only to the left (i.e. at the start), or DISTINCT=* if they need not be separated at all. The separators are specified by the SEPARATOR option.

Options: MRESPONSE, RESPONSECODES, LABELCODES, DUPLICATECODES, EXCLUDENULL, SUFFIXNULL, LABELNULL, DATAFORMAT, CASE, MULTISPACES, DISTINCT, SEPARATOR.
Parameter: DATA.

Method

FFREERESPONSEFACTOR uses the TXFIND directive to search for the response codes.

Action with RESTRICT

FFREERESPONSEFACTOR ignores any restrictions on the DATA texts.

See also

Procedures: FMFACTORS, MTABULATE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Survey analysis.

FHADAMARDMATRIX

Forms Hadamard matrices (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (monitoring); default * i.e. none
METHOD = *string token*	Method of construction (firstpaley, secondpaley, stored, sylvestre, tensorproduct, turyn, williamson); default * i.e. determined automatically

Parameters

NROWS = *scalars*	Number of rows of the matrices
HADAMARDMATRIX = *matrices*	Saves the Hadamard matrices
ERROR = *scalars*	Returns 0 if the matrix has been formed successfully and 1 if not

Description

A Hadamard matrix is a matrix containing values −1 and +1 whose rows are orthogonal: i.e. if H is a Hadamard matrix

```
H *+ T(H) = n * IDENTITY(n)
```

where *n* is the number of rows of H, which must be 2 or a multiple of 4.

FHADAMARDMATRIX provides several methods for forming the matrices, described in Hedayat, Sloane & Stufken (1999, Chapter 7). These work for all sizes of matrix up to 200, and various other sizes above that. The METHOD option can be used to specify the method. If this is unset FHADAMARDMATRIX selects an appropriate method automatically (you may then want to set the PRINT option to monitoring to record what method has been used). The settings of METHOD, in the order in which they will be selected, are as follows.

sylvestre	uses a tensor product construction building on the 2 × 2 matrix to form matrices of size $n = 2^m$ for any positive integer m.
williamson	builds the matrix from four circulant matrices; FHADAMARDMATRIX has a repertoire covering 20, 28, 36, 44, 52, 60, 68, 76, 84, 92, 100, 108, 116, 124, 132, 148, 156 and 172 rows.
turyn	builds the matrix from a circulant and seven retrocirculant matrices; this method is used for the matrix with 188 rows.
tensorproduct	uses a tensor product between the matrix with two rows, and a Hadamard matrix with $n/2$ rows.
secondpaley	forms matrices with $n = 2 \times (s + 1)$, where s is a prime power, using properties of the Galois field of order s.
firstpaley	forms matrices with $n = s + 1$, where s is a prime power, using properties of the Galois field of order s.
stored	takes a matrix from a stored repertoire with rows from four to 32.

The number of rows of the matrix is specified by the NROWS parameter, the HADAMARDMATRIX parameter saves the matrix, and the ERROR parameter can be set to a scalar which returns zero if the matrix has been formed successfully and one if not.

Options: PRINT, METHOD.
Parameters: NROWS, HADAMARDMATRIX, ERROR.

Method
The methods are described in Chapter 7 of Hedayat, Sloane & Stufken (1999).

Reference
Hedayat, A.S., Sloane, N.J.A., & Stufken, J. (1999). *Orthogonal Arrays, Theory & Applications*. Springer-Verlag, New York.

See also
Procedures: AGBIB, AGMAINEFFECT.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Design of experiments.

FHAT

Calculates an estimate of the F nearest-neighbour distribution function (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

Y1 = *variates*	Vertical coordinates of the first spatial point patterns; no default – this parameter must be set
X1 = *variates*	Horizontal coordinates of the first spatial point patterns; no default – this parameter must be set
Y2 = *variates*	Vertical coordinates of the second spatial point patterns; no default – this parameter must be set
X2 = *variates*	Horizontal coordinates of the second spatial point patterns; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
FVALUES = *variates*	Variates to receive the estimated F nearest-neighbour distribution functions
NNDISTANCES = *variates*	Variates to receive the nearest-neighbour distances

Description

The F nearest-neighbour distribution function relates to the distribution of distances from each of a set of sample points covering the region of interest to the nearest event of an observed spatial point pattern (see Diggle 1983). Other names for this function are the point-nearest event distribution function and the empty-space distribution function. An estimate of F can be obtained by generating a grid of points (for example, using the PTGRID procedure) and then calculating the empirical distribution function (EDF) FHAT(s) which is defined as the proportion of grid points for which the nearest event in the observed pattern is within distance *s*. The larger the number of grid points used, the better the approximation to the true distribution, F. For preliminary analysis, Diggle (1983) recommends using the same number of grid points as there are events in the observed pattern.

The term complete spatial randomness (CSR) is used to represent the hypothesis that the overall density of events in a spatial point pattern is constant throughout the study region, and that the events are distributed independently and uniformly. Under CSR, the F nearest-neighbour distribution function is given by

$$F(s) = 1 - \exp(-\pi \times density \times (s^2)),$$

where *density* is the overall density of events per unit area. (The procedure FZERO can be used to calculate values of this function for a pattern with a given density.) The F nearest-neighbour distribution function for a clustered (regular) pattern will tend to be smaller (larger) than the corresponding function for a completely random pattern, at least for small distances.

FHAT requires the coordinates of an observed spatial point pattern (specified by the parameters X1 and Y1), the coordinates of a set of sample points (specified by the parameters X2 and Y2), and a vector of distances at which to calculate the EDF of F (specified by the parameter S). The primary output of the procedure is a vector of estimates of F corresponding to the distances in S. The estimated F function can be saved using the parameter FVALUES. The nearest-neighbour distances can be saved using the parameter NNDISTANCES.

Printed output is controlled using the PRINT option. The default setting of summary prints the distances at which the F function is estimated and the estimates themselves under the headings S and FVALUES.

Option: PRINT.
Parameters: Y1, X1, Y2, X2, S, FVALUES, NNDISTANCES.

Method
A procedure PTCHECKXY is called to check that X1 and Y1 have identical restrictions. A similar check is made on X2 and Y2. The procedure then calls a procedure PTPASS to call a Fortran program to calculate the F nearest-neighbour distances. No corrections are made for edge effects. The EDF of the nearest-neighbour distances relative to the distances specified by the parameter S is obtained using the CALCULATE directive.

Action with RESTRICT
The variates X1, Y1, X2, Y2, and S may be restricted, as long as X1 has the same restriction as Y1, and X2 has the same restriction as Y2. Only the subset of values specified by each restriction will be included in the calculations.

Reference
Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

See also
Procedures: FZERO, GHAT, KHAT, KSTHAT, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

FIELLER

Calculates effective doses or relative potencies (P.W. Lane).

Options

PRINT = *string token*	What to output (value); default valu
ESTIMATES = *variate*	Parameter estimates; default extracts these with RKEEP
VCOVARIANCE = *symmetric matrix*	Variances and covariances; default extracts these with RKEEP
%LIMIT = *scalar*	Percentage points for limits; default 95, thus giving 95% confidence limits
RELATIVE = *string token*	Whether to calculate relative potencies (no, yes); default no
LINK = *string token*	Which link function to assume when forming effective doses (probit, logit, complementaryloglog); default obtained using RKEEP, if the ESTIMATES or VARIANCES are obtained in that way, otherwise prob
LOGBASE = *string token*	Base of antilog transformation to be applied to value and limits, (ten, e); default * i.e. none
DF = *scalar*	If this has a non-missing value, a t-distribution is used instead of a Normal distribution to calculate the confidence limits; default obtained using RKEEP if the ESTIMATES or VARIANCES are obtained in that way (setting DF to the number of residual d.f. when the dispersion factor is estimated, or a missing value when it is fixed), otherwise the default is a missing value

Parameters

TREATMENT = *variates*, *scalars* or *texts*	Positions of intercept parameters in list of estimates; default first estimate
SLOPE = *variates*, *scalars* or *texts*	Positions of slope parameters in list of estimates; default last estimate
%DOSE = *variates* or *scalars*	Percentage doses; default 50, thus giving LD50
VALUE = *variates* or *scalars*	To store estimated values
LOWER = *variates* or *scalars*	To store lower limits
UPPER = *variates* or *scalars*	To store upper limits
SE = *variates* or *scalars*	To store approximate s.e.s of values

Description

Quantal data from bioassay experiments can be analysed with the regression directives by fitting a generalized linear model with a binomial distribution and a probit, logit or complementary log-log link function. The coefficients estimated by the regression directives are the intercepts and slopes of the lines fitted on the transformed scale. However, you often need LD50s: that is, estimates of the median effective doses (or LD90s, and so on). These quantities are ratios of the estimated coefficients, and fiducial limits and approximate standard errors can be derived for their estimates using Fieller's theorem.

By default, FIELLER assumes that parameter estimates and their covariances can be extracted using the RKEEP directive from a fit already done using the regression directives. If this is not the case, these quantities must be supplied by setting the ESTIMATES and VCOVARIANCE options.

FIELLER can be used to estimate either effective doses of individual treatments (parallel or

non-parallel regression lines) or relative potencies compared to a standard treatment (parallel regression lines only). For effective doses, you specify the percentage point with the option %DOSE: so the default, %DOSE=50, gives LD50s. The link function to be assumed in the calculations can be specified with the LINK option. If this is not specified, and the estimates or variances are being extracted using RKEEP, the LINK will be set (using the OMODEL option RKEEP) to the same link as in the fitted generalized linear model; otherwise the default is to use a probit link. For relative potencies, you should set option RELATIVE=yes; %DOSE and LINK are then not relevant. The DF parameter can be set to use a t-distribution instead of a Normal distribution when calculating the confidence limits; this would be relevant if the dispersion factor in the generalized linear model was being estimated instead of being fixed at 1 (i.e. if a heterogeneity factor is being used). If DF is unset and the estimates or variances are being extracted using RKEEP, FIELLER uses RKEEP to see whether the dispersion is fixed and, if so, sets DF to the number of residual d.f. in the analysis; otherwise the default setting for DF is a missing value, which indicates that a Normal distribution should be used.

The SLOPE and TREATMENT parameters should be set to indicate the estimates of the slope and intercept parameters to be used. You can do this either by supplying a scalar or variate giving the position or positions in the list of estimates, or by giving a text containing their labels (as used in the tables of estimates printed by FIT &c). For effective doses, any model that includes a treatment effect should be fitted without an intercept (i.e. setting option FULL=yes in the TERMS statement and CONSTANT=omit in the FIT statement) so that the estimates produced by the regression directives are absolute intercepts rather than differences. For relative potencies, an intercept should be included, and the standard treatment represented as the first level of the treatments factor so that the estimates are differences of intercepts.

The procedure prints the estimate with lower and upper fiducial limits. The range of the limits can be set by the %LIMIT option: the default is 95% limits. Printing can be turned off by setting PRINT=*. The results of the procedure can be stored in scalars or variates using the VALUE, LOWER and UPPER parameters; also an approximate standard error of the estimated value (before back transformation, if relevant) can be stored using parameter SE.

Options: PRINT, ESTIMATES, VCOVARIANCE, %LIMIT, RELATIVE, LINK, LOGBASE, DF.
Parameters: TREATMENT, SLOPE, %DOSE, VALUE, LOWER, UPPER, SE.

Method
The fiducial limits are calculated using Fieller's Theorem; see, for example,
Finney (1971, page 78).

Reference
Finney, D.J. (1971). *Probit Analysis (third edition)*. Cambridge University Press, Cambridge.

See also
Procedure: PROBITANALYSIS.
Genstat Reference Manual 1 Summary section on: Regression analysis.

FILEREAD

Reads data from a file (P.W. Lane).

Options

PRINT = *string tokens*	What output to display (summary, groups, comments, firstline); default summ, grou, comm, firs
NAME = *text*	External name of the data file; no default in batch mode, name is prompted for in interactive mode
END = *text*	What string terminates data; default ':' (the end of file also terminates data for any setting); the setting END=* is not allowed
MISSING = *text*	What character represents missing values; default '*'
SKIP = *scalar* or *text*	Number of lines to skip at the start of the file, or string to indicate the record before the first record of data; default 0
MAXCATEGORY = *number*	The maximum number of categories for which a structure is defined to be a factor unless otherwise specified by FGROUPS; default 10
COMMENTSYMBOLS = *text*	What characters to treat as introducing comments if found in the first column at the start of the file; default double-quote character (")
IMETHOD = *string token*	How identifiers are to be specified for the data structures to be read (supply, read, none); default supp
ISAVE = *pointer*	To store the identifiers, whether read or supplied, and to provide suffixed identifiers for data with no specified identifiers
SEPARATOR = *text*	What (single) character separates successive values; default is the space character

Parameters

IDENTIFIER = *identifiers*	Names for the data structures that are to be read; these are prompted for if this is unset when running interactively with IMETHOD=supply; identifiers are redefined if they have been used previously
FGROUPS = *string tokens*	Whether to form each data structure into a factor (check, form, leave); default chec, which causes FILEREAD when running interactively to ask about any structure whose number of distinct values is less than or equal to MAXCATEGORY, and when running in batch to define as factors all structures with MAXCATEGORY or fewer distinct values (note: for compatibility with earlier releases, yes and no can be used as synonyms of form and leave)
REPRESENTATION = *string tokens*	What representation to assume for each data structure (numbers, characters); default unset – representation is determined by whether the first value is a number; if set for one structure, this parameter must be set for all structures

Description

FILEREAD reads data from a file into suitable structures determined from the data. It can deal with values laid out as follows.

(1) A character file: that is, a normal readable file, or flat file.
(2) Maximum record length of 200 characters.
(3) Contents consist of values for one or more data structures – usually presented as a single rectangular data matrix.
(4) The values for the data structures are recorded in parallel – that is, the first values of all the structures, followed by the second values of all, and so on; usually, each record of the file contains one value of each structure, but multiple values per record and multiple records for each unit can also be dealt with.
(5) Values in a record are separated from each other by the same separator – usually one or more spaces.
(6) Text values must be enclosed in single quotes if they contain a space, comma, backslash, or double-quote; single-quotes must be used only to enclose textual values, or be duplicated as part of a value which is also enclosed in single quotes.
(7) Comments are allowed at the start of the file only if every record to be treated as a comment starts with a double quote, or other specified symbol. Alternatively, a specified number of records at the start of the file can be skipped, or any number of records up to and including a specified string.
(8) Identifiers for the columns of the matrix can be read from the first row of data, as long as they are valid, unsuffixed, Genstat identifiers. An exclamation mark after an identifier signals that the structure is to be set up as a factor.

Information may be numerical or textual. Numerical values are read as variates, and textual as texts, determined by the values in the first complete record or by the REPRESENTATION parameter. If this parameter is unset, FILEREAD searches for the first record in the file with no missing values, and fails if there is no such record. If the REPRESENTATION parameter is set, it determines whether the values of each structure are to be treated as numbers or characters; if set for any structure, this parameter must be set for all of them.

The NAME option of the procedure supplies the name of the file, enclosed in single quotes. In batch mode the name must be supplied, but in interactive mode, FILEREAD will prompt for the name if it is not supplied.

The IMETHOD option controls the specification of identifiers for the structures to be read. With the default, IMETHOD=supply, the identifiers can be listed using the IDENTIFIER parameter, one for each column of the data matrix. If IDENTIFIER is not set when running in interactive mode, FILEREAD will prompt for identifiers; if it is unset when running in batch mode, FILEREAD just reports on the contents of the file, unless option ISAVE is set (see below). If IMETHOD=read, FILEREAD will attempt to read identifiers for the data structures from the first complete record in the file (and the IDENTIFIER parameter is ignored). They must be valid Genstat identifiers, and must not include suffixes. If an exclamation mark is found after (or in) an identifier, then the structure will be set up as a factor unless the FGROUPS parameter is set to leave. (This convention matches that used when data is read into a Genstat spreadsheet using menus.) If IMETHOD=none, FILEREAD just reports on the contents of the file without assigning identifiers unless option ISAVE is set.

The ISAVE option can be set to a pointer to store the identifiers read from the file (if IMETHOD=read) or supplied interactively (if IMETHOD=supply). If IMETHOD=none in either mode, or IMETHOD=supply and the IDENTIFIER parameter is unset in batch mode, the data structures can be referred to using the pointer.

Values on the same record of a file must be separated from each other by at least one space unless the SEPARATOR option is set. This option can nominate any single character to be treated as data separator. The MISSING and END options specify the missing-value and end-of-file

symbols.

If the number of identifiers is not specified, the number of data structures is taken to be the number of values on the first record with no missing values. But if identifiers are supplied in the IDENTIFIER parameter, or read from the data file, it is possible to read several units of data from each record, or each unit from several records. If there are more values on the first record of data than there are identifiers, the type of each data structure can be determined only by its first value: FILEREAD will fail if any first value is missing, unless the REPRESENTATION parameter is set. If there are fewer values on the first record of data than there are identifiers, FILEREAD will fail regardless of the absence of missing values unless the REPRESENTATION parameter is set.

By default, FILEREAD reports what structures are set up and tabulates the number of values in each category for structures that have MAXCATEGORY or less distinct values. It also displays any comments that it identifies before the start of the data, and the first record of data that contains no missing values. These four reports are controlled by the PRINT option.

The FGROUPS parameter allows structures to be formed automatically into factors. The default setting is check: in interactive mode, FILEREAD then prompts for a decision about any structure where the number of distinct values is less than or equal to the setting of the MAXCATEGORY option; in batch mode, all structures with these few distinct values become factors automatically. FGROUPS can also be set to form or leave to specify explicitly whether each structure should or should not be defined automatically as a factor. (The settings form or leave were introduced in Procedure Library PL21 to remove the confusion arising from the fact that other options and parameters that have no as a setting, use no as their default. However, for compatibility with earlier programs, the settings yes and no are still recognised as synonyms for form and leave.)

The COMMMENTSYMBOLS option can be set to a list of single characters, in quotes. If any of these characters is found at the start of a record, before any data has been read, that record will be treated as a comment. By default, the double-quote symbol is the only comment symbol, but it must appear at the start of every record to be treated as a comment.

The SKIP option allows records at the start of the file to be skipped altogether. It can be set either to the number of records to be skipped, or to a string, indicating that all records are to be skipped up to and including the first record containing that string.

Options: PRINT, NAME, END, MISSING, SKIP, MAXCATEGORY, COMMENTSYMBOLS, IMETHOD, ISAVE, SEPARATOR.
Parameters: IDENTIFIER, FGROUPS, REPRESENTATION.

Method

The file is opened on the first free input channel. The first record is read as a single string, and then individual items are read from the string into a text. This is tested, and the process repeated until a record has been found that is not blank or a comment, and has no missing items. Items are tested to determine if they are valid numbers, and then the whole file is read into variates and texts as appropriate. Each structure is grouped to provide information about numbers of categories.

See also

Directive: READ.
Procedures: IMPORT, DBIMPORT, SPLOAD.
Genstat Reference Manual 1 Summary section on: Input and output.

FITINDIVIDUALLY

Fits regression models one term at a time (R.W. Payne).

Options

PRINT = *string tokens*	What to print (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, confidence); default mode, summ, esti
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalar*	Limit for expansion of model terms; default 3
POOL = *string token*	Whether to pool ss in accumulated summary between all terms fitted in a linear model (yes, no); default no
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default *
FPROBABILITY = *string token*	Printing of probabilities for variance and deviance ratios (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-statistics (yes, no); default no
SELECTION = *string tokens*	Statistics to be displayed in the summary of analysis produced by PRINT=summary, seobservations is relevant only for a Normally distributed response, and %cv only for a gamma-distributed response (%variance, %ss, adjustedr2, r2, seobservations, dispersion, %cv, %meandeviance, %deviance, aic, bic, sic); default %var, seob if DIST=normal, %cv if DIST=gamma, and disp for other distributions
PROBABILITY = *scalar*	Probability level for confidence intervals for parameter estimates; default 0.95
DEVIANCE = *scalar*	Saves the residual deviance
DF = *scalar*	Saves the residual d.f.
LACKOFFIT = *string token*	Whether to use observations with replicated values of the explanatory variables to split the final residual term into a 'true' residual and lack of fit (estimate, omit); default omit

Parameter

TERMS = *formula*	Terms to be fitted

Description

FITINDIVIDUALLY is provided as an alternative to the FIT directive for use, in particular, with generalized linear models. With these models, for efficiency, the entire model is fitted at once rather than one term at a time as in ordinary regression models. As a result the terms of the generalized linear model are pooled into a single line in the analysis of deviance table. However, if you want to see the contributions of the individual terms in the analysis of deviance table, you can use FITINDIVIDUALLY instead of FIT.

FITINDIVIDUALLY is used exactly like FIT. It must be preceded by a MODEL statement, and can be followed by RCHECK, RDISPLAY, RGRAPH, RKEEP, ADD, DROP, SWITCH and so on. It has a TERMS parameter to specify the terms to be fitted, like the parameter of FIT. It also has options PRINT, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION and PROBABILITY which operate like those of FIT.

If you have observations with replicated values of the explanatory variables, you can set option LACKOFFIT=estimate to split the final residual term into a "true" residual (measured by the variation amongst the replicate observations) and lack of fit. FITINDIVIDUALLY then sets the dispersion parameter and its number of degrees of freedom in the regression save structure to the "true" residual deviance and its degrees of freedom, so that these will be used for standard errors and probabilities etc. in future output. (These are the aspects that you can set using the DISPERSION and DFDISPERSION options of MODEL.) The DEVIANCE option allows you to save the residual deviance, and the DF option saves the residual number of degrees of freedom.

Options: PRINT, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION, PROBABILITY, DEVIANCE, DF, LACKOFFIT.
Parameter: TERMS.

Method

FITINDIVIDUALLY uses FCLASSIFICATION to break the TERMS formula up into individual terms. It fits these individually using ADD, and then calls RDISPLAY to display the output. It uses procedure FACCOMBINATIONS to identify the observations with replicated values of the explanatory variables so that it can calculate the lack of fit. It calls an auxiliary procedure _FITIRSET for setting the dispersion parameter and its number of degrees of freedom in the regression save structure (this uses inside knowledge of the structure of the structure).

Action with RESTRICT

As in FIT, the y-variate (specified in an earlier MODEL directive) can be restricted to analyse a subset of the data.

See also

Directive: ADD, FIT.
Genstat Reference Manual 1 Summary section on: Regression analysis.

FITMULTINOMIAL

Fits generalized linear models with multinomial distribution (R.W. Payne).

Options

PRINT = *string tokens*	What to print (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, confidence); default mode, summ, esti
RESPONSEFACTOR = *factor*	Factor representing the response categories of the multinomial distribution
CLASSIFICATION = *factors*	Factors classifying the subjects; default uses the factors in TERMS
FACTORIAL = *scalar*	Limit for expansion of model terms from TERMS; default 3
POOL = *string token*	Whether to pool ss in accumulated summary between all terms fitted in a linear model (yes, no); default no
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default *
FPROBABILITY = *string token*	Printing of probabilities for variance and deviance ratios (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-statistics (yes, no); default no
SELECTION = *string tokens*	Statistics to be displayed in the summary of analysis produced by PRINT=summary (%variance, %ss, adjustedr2, r2, dispersion, %meandeviance, %deviance, aic, bic, sic); default disp
PROBABILITY = *scalar*	Probability level for confidence intervals for parameter estimates; default 0.95
FULL = *string token*	Whether to assign all possible parameters to factors and interactions (yes, no); default no

Parameter

TERMS = *formula*	Terms to be fitted

Description

FITMULTINOMIAL provides an automatic way of fitting generalized linear models with the multinomial distribution. These models can be fitted with the ordinary generalized linear models commands, by using the fact that a multinomial distribution can be generated by taking the sum of several Poisson variables (one for each outcome of the multinomial), and then constraining their sum to be equal to the multinomial total (see McCullagh & Nelder 1989, or any book on probability distributions).

The data for the model are counts of numbers of subjects observed in the various categories of the multinomial distribution. The counts may also be classified by various treatment factors, and the interest is in seeing how distribution of the subjects varies according to the treatments or any variates that differ over the levels of the treatments. These observations must be put into a single variate, and specified beforehand using Y parameter of the MODEL directive. The DISTRIBUTION option of MODEL should be set to Poisson, and the LINK option to logarithm

(or left as the default `canonical` for the canonical link, which is `logarithm` for the Poisson); this gives a logit link in the multinomial.

You also need to form a factor to identify the response category of the multinomial recorded in each unit of the `Y` variate. This is then input to `FITMULTINOMIAL` using the `RESPONSEFACTOR` option. `FITMULTINOMIAL` also has a `CLASSIFICATION` option that can be used to specify the factors that classify the subjects. The other options have the same purpose as those in the `FIT` directive. The model to be fitted is specified by the `TERMS` parameter (like the first parameter of `FIT`). If `CLASSIFICATION` is unset, `FITMULTINOMIAL` will use the set of factors that occur in `TERMS`. Usually these will contain all the factors that classify the subjects. However, if you have a classification factor with numerical levels, you might for example want to fit a variate calculated as some function of the levels rather that a effect for every level of the factor. You could then specify the factors in the list for the `CLASSIFICATION` option, and use the variate in `TERMS`.

`FITMULTINOMIAL` first fits a model defined as all factorial combinations of the `CLASSIFICATION` factors. This imposes the constraint that the Poisson variables sum to the totals of the multinomial distribution. The effects of these terms assess how the design has been set up – i.e. how the subjects have been allocated to the treatments – but they have no information on the effects of the treatments on the response.

It then fits `RESPONSEFACTOR`. This represents the overall distribution of the response categories across the subjects, and is analogous to the grand mean in an ordinary analysis. (This must be fitted, and so `FITMULTINOMIAL` has no `CONSTANT` option.)

Finally it fits the interactions of the terms in `TERMS` with `RESPONSEFACTOR`. These show how the distribution of subjects to response categories is affected by the treatment terms – which is the main interest of the analysis. The `FACTORIAL` option sets a limit on the number of factors and/or variates in the model terms that are generated from the `TERMS` formula. (Note, though, that the `RESPONSEFACTOR` is ignored in interpreting this limit). By default these terms are fitted individually, so they will each have their own line in an accumulated analysis of deviance (option `PRINT=accumulated`). However, you can set option `POOL=yes` to fit them all at once.

After `FITMULTINOMIAL` you can use the standard regression output commands, `RDISPLAY`, `RKEEP` and so on, in the usual way.

If you have a large model, you can set the GROUPS option in the earlier `MODEL` statement to the response factor to save space. Note, though, that if you want to use the `PREDICT` directive after `FITMULTINOMIAL`, you will then only be able to predict values within one response category at a time.

Options: `PRINT`, `RESPONSEFACTOR`, `CLASSIFICATION`, `FACTORIAL`, `POOL`, `DENOMINATOR`, `NOMESSAGE`, `FPROBABILITY`, `TPROBABILITY`, `SELECTION`, `PROBABILITY`, `FULL`.
Parameter: `TERMS`.

Method

`FITMULTINOMIAL` uses the standard generalized linear models commands, as explained in the *Description*.

Action with `RESTRICT`

As in `FIT`, the y-variate (specified in an earlier `MODEL` directive) can be restricted to analyse a subset of the data.

Reference

McCullagh, P. & Nelder, J.A. (1989). *Generalized Linear Models (second edition)*. Chapman & Hall, London.

See also
Directive: MODEL.
Genstat Reference Manual 1 Summary section on: Regression analysis.

FMEGAENVIRONMENTS

Forms mega-environments based on winning genotypes from an AMMI-2 model (D.A. Murray & M. Malosetti).

Option

PRINT = *string tokens*	What to print (summary); default summ

Parameters

DATA = *variates*	Provides the data to be analysed
GENOTYPES = *factors*	Specifies the genotypes
ENVIRONMENTS = *factors*	Specifies the environments (or locations when years are supplied)
YEARS = *factors*	Specifies years within locations
MEGAENVIRONMENTS = *factors*	Saves the mega-environments

Description

FMEGAENVIRONMENTS forms mega-environments based on the winning genotype from each environment using an AMMI-2 model.

The data to be analysed must be supplied in a variate, using the DATA parameter. The associated genotype and environment factors are specified using the GENOTYPES and ENVIRONMENTS parameters, respectively.

For environments that contain data from several years, FMEGAENVIRONMENTS can form mega-environments on the basis of locations and not years. The locations should then be supplied in a factor using the ENVIRONMENTS parameter, and the years within the locations should be supplied in a factor using the YEARS parameter.

The MEGAENVIRONMENTS parameter saves the mega-environments in a factor.

You can set option PRINT=summary to display a summary of the winning genotype in each environment and the mega-environment allocation.

Option: PRINT.
Parameters: DATA, GENOTYPES, ENVIRONMENTS, YEARS, MEGAENVIRONMENTS.

Method

The AMMI procedure is used to fit an AMMI-2 model to the genotype-by-environment data table, or the genotype-by-location data table when years within locations have been supplied. The fitted values are then used to identify the winning genotype within each environment.

Action with RESTRICT

FMEGAENVIRONMENTS takes account of any restrictions on DATA, GENOTYPES, ENVIRONMENTS and YEARS.

See also

Procedure: AMMI.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

FMFACTORS

Forms a pointer of factors representing a multiple-response (R.W. Payne).

Options

MRESPONSE = *pointer*	Pointer with a factor for each code, indicating the units where it occurs in the CODE texts or variates
RESPONSECODES = *text* or *variate*	Saves the set of distinct multiple-response codes
CODENULL = *text* or *variate*	Code(s) used to represent a null value in the CODE texts or variates; default * or ''
EXCLUDENULL = *string token*	Whether to exclude the null factor recording the respondents that made no reply (yes, no); default no
SUFFIXNULL = *scalar*	Suffix to use to represent the null factor in MRESPONSE; default 0
LABELNULL = *text*	Label to use to represent the null factor in MRESPONSE; default 'none'
LDIRECTION = *string token*	How to order the labels from textual codes (ascending, given); default asce

Parameter

CODE = *texts*, *variates* or *factors*	Codes from the respondents

Description

Multiple responses occur in surveys as the result of open-ended questions like "Which cities have you visited this year?" or "What languages do you speak?". The easiest way to input these into Genstat is in a set of text vectors. Each text has a unit for every respondent, and the set contains as many texts as the maximum number of the replies from any respondent. Alternatively, if the responses are numerical, they can be input in a set of variates. The MTABULATE procedure can form tables with multiple responses. However, these raw codes must first be converted by FMFACTORS into a set of factors.

The texts or variates containing the raw data are listed using the CODE parameter. You can also supply the raw data in factors. If CODE specifies a mixture of texts and factors, FMFACTORS uses the labels of the factors (and they must all have labels). Alternatively, if CODE specifies a mixture of variates and factors, FMFACTORS uses the factor levels. Finally, if CODE specifies only factors, FMFACTORS will use their labels if they all have labels; otherwise their levels. FMFACTORS will give a fault if you specify a mixture of texts and variates.

The multiple-response factors are saved, in a pointer, using the MRESPONSE option. The pointer contains a factor for every recorded code, with levels 0 and 1, and corresponding labels 'absent' and 'present'. If the codes are textual, the various strings are used as labels of the pointer; while if they are numerical, the numbers are used as the pointer suffixes.

By default, the texts or variates are assumed to contain a missing values for any null response: for example these would occur in the third and fourth text, if there were four CODE texts and the respondent concerned had made only two replies. However, you can use the CODENULL option to supply alternative codings (for example '-' for textual responses).

The EXCLUDENULL option controls whether or not the pointer contains a factor to make an explicit record of the respondents that made no replies at all (default no). This will be needed if the later tables are to contain a line for "no response". The SUFFIXLNULL option specifies the suffix to be used for this factor in the pointer while, for textual codes, the LABELNULL option specifies its label in the pointer.

The LDIRECTION option controls whether the labels defined from textual codes are put into ascending alphabetic order (the default), or kept in the order in which they occur in the CODE texts.

Options: MRESPONSE, RESPONSECODES, CODENULL, EXCLUDENULL, SUFFIXNULL, LABELNULL, LDIRECTION.
Parameter: CODE.

Method
FMFACTORS uses the standard Genstat calculation commands.

Action with RESTRICT
FMFACTORS takes account of any restrictions on the CODE texts or variates.

See also
Procedures: FFREERESPONSEFACTOR, MTABULATE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Survey analysis.

FNCORRELATION

Calculates correlations from variances and covariances, together with their variances and covariances (S.A. Gezan).

Options

PRINT = *string token*	Output required (summary); default summ
IVARIANCES = *variate*	Indexes of the two variances in the ESTIMATES variate; no default – must be set
ICOVARIANCE = *scalar*	Index of the covariance in the ESTIMATES variate; no default – must be set

Parameters

ESTIMATES = *variates*	Estimated values of the variances and covariances
VCOVARIANCE = *symmetric matrices*	Variance-covariance matrix of the variances and covariances
FUNCTIONESTIMATE = *scalars*	Saves the estimated value of the function
SE = *scalars*	Saves the standard error of the function estimate
NEWESTIMATES = *variates*	Saves new vectors of estimates, including the estimated value of the function
NEWVCOVARIANCE = *symmetric matrices*	Saves variance-covariance matrices for the new vectors (including the function estimate)

Description

FNCORRELATION estimates correlations from variances and covariances. The estimated values of the variances and covariances, are contained in a variate supplied by the ESTIMATES parameter. The positions of the two variances in the ESTIMATES variate are specified (in a variate of length two) by the IVARIANCES option, and the position of the covariance is specified (in a scalar) by the ICOVARIANCES option. The variances and covariances of the ESTIMATES are supplied (in a symmetric matrix) by the VCOVARIANCE parameter.

The estimated correlation can be saved by the FUNCTIONESTIMATE parameter, and its standard error can be saved by the SE option (both in scalars). The NEWESTIMATES parameter can save a new variate of estimates, containing first the original ESTIMATES variate and then the function estimate. The corresponding variance-covariance matrix can be saved (in a symmetric matrix) by the NEWVCOVARIANCE parameter.

Options: PRINT, IVARIANCES, ICOVARIANCE.
Parameters: ESTIMATES, VCOVARIANCE, FUNCTIONESTIMATE, SE, NEWESTIMATES, NEWVCOVARIANCE.

Method

The correlation function w is calculated from the random variances f and g, and covariance h by the expression:

$$w = h / \sqrt{(f \times g)}$$

The variance of the estimated correlation is approximated using a first-order Taylor series expansion (i.e. the *delta* method); see Holland (2006).

$$\begin{aligned}\mathrm{var}(w) &= \mathrm{var}(h / \mathrm{sqrt}(f \times g)) \\ &= \mathrm{E}(w)^2 \times \{ \mathrm{var}(f) / (4 \times \mathrm{E}(f)^2) + \mathrm{var}(g) / (4 \times \mathrm{E}(g)^2) + \mathrm{var}(h) / \mathrm{E}(h)^2 \\ &\quad + \mathrm{cov}(f,g) / (2 \times \mathrm{E}(f) \times \mathrm{E}(g)) - \mathrm{cov}(f,h) / (\mathrm{E}(f) \times \mathrm{E}(h)) - \mathrm{cov}(h,g) / (\mathrm{E}(h) \times \mathrm{E}(g))\}\end{aligned}$$

Reference

Holland, J.B. (2006). Estimating genotypic correlations and their standard errors using multivariate restricted maximum likelihood estimation with SAS Proc MIXED. *Crop Sci.*, **46**, 642-654.

See also

Procedures: FNLINEAR, FNPOWER.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FNLINEAR

Estimates linear functions of one or more random variables, and calculates their variances and covariances (S.A. Gezan).

Options

`PRINT` = *string token*	Output required (`summary`); default `summ`
`CONSTANTVALUE` = *scalar*	Constant value for the function; default 0
`COEFFICIENTS` = *scalar*	Linear coefficients for the random variables in the function; no default – must be set

Parameters

`ESTIMATES` = *variates*	Estimated values of the random variables
`VCOVARIANCE` = *symmetric matrices*	Variance-covariance matrix of the random variable estimates
`FUNCTIONESTIMATE` = *scalars*	Saves the estimated value of the function
`SE` = *scalars*	Saves the standard error of the function estimate
`NEWESTIMATES` = *variates*	Saves new vectors of estimates, including the estimated value of the function
`NEWVCOVARIANCE` = *symmetric matrices*	Saves variance-covariance matrices for the `NEWESTIMATES`

Description

`FNLINEAR` estimates linear functions of one or more random variables. The estimated values of the random variables, from which the function value is calculated, are supplied (in a variate) by the `ESTIMATES` parameter. Their variances and covariances are supplied (in a symmetric matrix) by the `VCOVARIANCE` parameter. The linear coefficients for the function are supplied (again in a variate) by the `COEFFICIENTS`, and the constant is supplied (in a scalar) by the `CONSTANTVALUE` option. So the function value is given by

```
SUM(ESTIMATES * COEFFICIENTS) + CONSTANTVALUE
```

The value can be saved by the `FUNCTIONESTIMATE` parameter, and its standard error can be saved by the `SE` option (both in scalars). The `NEWESTIMATES` parameter can save a new variate of estimates, containing the original `ESTIMATES` variate and then the function value inserted at the end. The corresponding variance-covariance matrix can be saved (in a symmetric matrix) by the `NEWVCOVARIANCE` parameter.

Options: `PRINT`, `CONSTANTVALUE`, `COEFFICIENTS`.
Parameters: `ESTIMATES`, `VCOVARIANCE`, `FUNCTIONESTIMATE`, `SE`, `NEWESTIMATES`, `NEWVCOVARIANCE`.

Method

The linear function w of the random variables f, g, h etc. is defined by the expression:

$$w = a_0 + a_1 \times f + a_2 \times g + a_3 \times h + ...$$

where a_0, a_1, a_2 etc. are (known) coefficients. The estimated means and variances of the random variables, supplied by the `ESTIMATES` and `VCOVARIANCE` parameter, are used to calculate the estimated value of the function w and to calculate its variance. If the original random variables are Normally distributed, the random variable w is also Normally distributed and the variance calculation is exact.

Action with `RESTRICT`
Any restrictions are ignored.

See also
Procedures: `FNCORRELATION`, `FNPOWER`.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FNPOWER

Estimates products of powers of two random variables, and calculates their variances and covariances (S.A. Gezan).

Options

PRINT = *string token*	Output required (summary); default summ
CONSTANTVALUE = *scalar*	Constant value for the function; default 0
POWERS = *variate*	Specifies the powers of the two random variables
INDEXES = *variate*	Specifies the locations of the random variables corresponding to the elements of the POWERS variate
CORRECTION = *string token*	Whether to apply an additional correction to the variance of a product, using terms from the second-order approximation; default no

Parameters

ESTIMATES = *variates*	Estimated values of the random variables
VCOVARIANCE = *symmetric matrices*	Variance-covariance matrix of the random variable estimates
FUNCTIONESTIMATE = *scalars*	Saves the estimated value of the function
SE = *scalars*	Saves the standard error of the function estimate
NEWESTIMATES = *variates*	Saves new vectors of estimates, including the estimated value of the function
NEWVCOVARIANCE = *symmetric matrices*	Saves variance-covariance matrices for the new vestors (including the function estimate)

Description

FNPOWER estimates products of powers of two random variables. The estimated values of the random variables, from which the function estimate is calculated, are supplied (in a variate) by the ESTIMATES parameter. Their variances and covariances are supplied (in a symmetric matrix) by the VCOVARIANCE parameter. The positions of the random variables in the ESTIMATES variate are specified by the INDEXES option, and their powers are specified by the POWERS option (both in variates of length two).

The estimate can be saved by the FUNCTIONESTIMATE parameter, and its standard error can be saved by the SE option (both in scalars). The NEWESTIMATES parameter can save a new variate of estimates, containing the original ESTIMATES variate and then the function estimate inserted at the end. The corresponding variance-covariance matrix can be saved (in a symmetric matrix) by the NEWVCOVARIANCE parameter.

The variance and covariances are calculated using a first-order Taylor expension. You can obtain a more accurate value for the variance of an ordinary product by setting option CORRECTION=yes. (FNPOWER then uses a second-order Taylor expansion.)

Options: PRINT, CONSTANTVALUE, POWERS, INDEXES, CORRECTION.

Parameters: ESTIMATES, VCOVARIANCE, FUNCTIONESTIMATE, SE, NEWESTIMATES, NEWVCOVARIANCE.

Method

The power function w, of the random variables f and g, is defined by the expression:

$$w = f^p \times g^q$$

for the real-valued coefficients p and q (defined by the POWERS parameter). The functions that

can be defined thus include:

single power	$w = f^p$ (i.e. $q = 0$),
square root	$w = \sqrt{f}$ (i.e. $p = 0.5$, $q = 0$),
product	$w = f \times g$ (i.e. $p = q = 1$),
ratio	$w = f / g$ (i.e. $p = q = -1$).

The variances and covariances of the function are approximated using a first-order Taylor series expansion (i.e. the *delta* method); see Kendall & Stuart (1963). For example the expressions for the variance of the product and ratio functions are as follows:

product	$\text{var}(w) = \text{var}(f \times g)$ $= \text{E}(f)^2 \times \text{var}(g) + \text{E}(g)^2 \times \text{var}(f) + 2 \times \text{E}(f) \times \text{E}(g) \times \text{cov}(f,g)$
ratio	$\text{var}(w) = \text{var}(f / g)$ $= (1 / \text{E}(g)^2) \times \{ \text{var}(f) - 2 \times \text{E}(w) \times \text{cov}(f,g) + \text{E}(w)^2 \times \text{var}(g) \}$

The quality of this approximation depends on the linearity of the function near the estimate. For a product, you can request an additional correction for the product function based on a second-order Taylor expansion. A correction factor *cf* is then added to the expression above, where

$cf = \text{var}(f) \times \text{var}(g) + \text{cov}(f,g)^2$.

Reference

Kendall, M. & Stuart, A. (1963). *The Advanced Theory of Statistics, Volume 1*. Griffin, London.

See also

Procedures: FNCORRELATION, FNLINEAR.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FOCCURRENCES

Counts how often each pair of treatments occurs in the same block (W. van den Berg).

Options

`PRINT` = *string tokens*	Controls printed output (`concurrences`, `efficiency`); default `conc`, `effi`
`DIAGONAL` = *string token*	What to store on the diagonal of the concurrence matrix (`missingvalues`, `replication`); default `repl`

Parameters

`TREATMENTS` = *factors*	Supplies the treatment factor
`REPLICATES` = *factors*	Supplies the replicates factor
`BLOCKS` = *factors*	Supplies the block factor
`CONCURRENCES` = *symmetric matrices*	Saves the concurrence matrix, recording the number of times each pair of treatments occurs together in a block
`EFFICIENCY` = *scalars*	Save the efficiency of the design

Description

`FOCCURRENCES` forms a symmetric "concurrence" matrix recording the number of times that each pair of treatments occurs together in the same block. If the treatments all have the same replication, it can also calculate the efficiency of the design, namely the average efficiency factor of the treatment contrasts after eliminating blocks.

The treatment and block factors are supplied by the `TREATMENTS` and `BLOCKS` factors, respectively, and the concurrence matrix and the efficiency can be saved by the `CONCURRENCES` and `EFFICIENCY` parameters, respectively. For resolvable designs the replicate factor can be supplied using the `REPLICATES` parameter.

Printed output is controlled by the `PRINT` option, with settings:

`concurrences`	to print the concurrence matrix, and
`efficiency`	to print the efficiency.

By default, both are printed.

The diagonal of the concurrence matrix usually contains the replication of each treatment i.e. its concurrence with itself. Alternatively, if you are interested only in the concurrences of pairs of different treatments, you can put missing values in the diagonal by setting option `DIAGONAL=missingvalues`.

Options: `PRINT`, `DIAGONAL`.
Parameters: `TREATMENTS`, `REPLICATES`, `BLOCKS`, `CONCURRENCES`, `EFFICIENCY`.

Method

First the treatments-by-blocks incidence matrix `N` is formed. This contains one in row *i* and column *j* if treatment *i* occurs in block *j*, otherwise it contains zero. The symmetric matrix of concurrences can then be calculated as

```
N *+ T(N)
```

See John & Williams (1995).

The efficiency is calculated by analysing a y-variate of Normally-distributed random numbers, using `REML`, with fixed model

```
BLOCKS + TREATMENTS
```

The efficiency of the design is then calculated as

$(\sigma^2 \times 2) / (r \times meanv)$

where σ^2 is the residual variance,
r is the replication of the treatments, and
meanv is the mean of the squares of the standard errors of differences of the treatment effects.

For resolvable designs the block factor Blocks is used and constructed using

```
FACPRODUCT !P(REPLICATES, BLOCKS); Blocks.
```

Action with RESTRICT

FOCCURRENCES takes account of restrictions on BLOCKS or TREATMENTS.

Reference

John, J.A. & Williams, E.R. (1995). *Cyclic and Computer Generated Designs, 2nd edition*. Chapman & Hall, London.

See also

Procedures: AFCYCLIC, AGBIB, AGCYCLIC.
Genstat Reference Manual 1 Summary section on: Design of experiments.

FPARETOSET

Forms the Pareto optimal set of non-dominated groups (W. van den Berg).

Options

PRINT = *string token*	Controls whether to print the groups (groups); default grou
PLOT = *string token*	Controls whether to plot the data, using different coloured points to indicate the groups (data); default data
NGROUPS = *scalar*	Number of groups to form; default 1
GROUPS = *factor*	Saves the group allocations
TITLE = *text*	Title for the plot; default * i.e. none
LABELS = *text*, *variate* or *factor*	Labels for the items; default * i.e. none

Parameters

DATA = *variates*	Data variates, defining the properties of the items
SIGN = *scalars*	Value by which to multiply each DATA variate: for example, this can be set to set to −1 if the variate is to be minimized instead of being maximized; default 1
TITLE = *texts*	Title to use for the axis of each DATA variate in the plot; if unset, its identifier is used

Description

FPARETOSET is useful when you want to select an item according to several different properties. Often there will be no uniquely optimal item: one item may be best according to one property, while others are best according to other properties. One way to assist with the decision is to form the set of *non-dominated* items (i.e. those that are not dominated by any other item). An item is said to *dominate* another item, if it has at least one property that is better than that item, while its other properties are at least as good. The non-dominated items form the *Pareto optimal set*, which includes the item that would be selected as best according to any weighting that may be chosen for the properties. See, for example, Goldberg (1989) pages 197-201 for more details.

The process can be extended to form a hierarchy over the items. You can form a second set by removing the non-dominated items, and finding the items that are now non-dominated. You can then form a third set of items, consisting of those that are non-dominated once the first two sets have been removed, and so on, until you reach the set of items that do not dominate any other item.

The properties of the items are specified in variates (one for each property) by the DATA parameter. By default, it is assumed that the maximum value of each property is best, but you can set the SIGN parameter to minus one (or any other negative value) if you want to minimize it instead. The NGROUPS option defines the number of non-dominated sets to form, and the GROUPS option can save a factor indicating the set to which each item is allocated. (The GROUPS factor will thus have NGROUPS plus one levels.)

By default, FPARETOSET prints the DATA variates and group allocations, but you can suppress that by setting option PRINT=*. It also plots the data in a scatter-plot matrix, using different coloured points to indicate the groups; this can be suppressed by setting PLOT=*. The LABELS option can supply labels for the items to use in the printed output and graph. The TITLE option can supply a title for the graph, and the TITLE parameter can supply a title to use for the axis of each DATA variate in the plot.

Options: PRINT, PLOT, NGROUPS, GROUPS, TITLE, LABELS.
Parameters: DATA, SIGN, TITLE.

Action with RESTRICT
Restricted units of the DATA variates are ignored.

Reference
Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley.

See also
Procedures: DEMC, MINIMIZE, MIN1DIMENSION, SIMPLEX.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation.

FPLOTNUMBER

Forms plot numbers for a row-by-column design (K. Punyawaew).

Options

FIRSTPLOT = *string token*	Defines the starting location for numbering the plots (lowleft, lowright, upleft, upright); default uple
PLOTORDER = *string token*	Defines the order in which the numbers are allocated (colserpentine, colbycol, rowserpentine, rowbyrow); default rowb

Parameters

NROWS = *scalars*	Number of rows in the design
NCOLUMNS = *scalars*	Number of columns in the design
PLOTNUMBER = *factors*	Saves the plot numbers

Description

FPLOTNUMBER generates unique numbers that can be used to identify the plots in a row-by-column design. These are often used to define the order in which various husbandry operations will be carried out. Several ways are therefore provided to specify the order.

The FIRSTPLOT option specifies where the numbering should start, as follows:

lowleft	left-hand plot at the bottom of the design i.e. in the final row and first column;
lowright	right-hand plot at the bottom of the design i.e. in the final row and final column;
upleft	left-hand plot at the top of the design i.e. in the first row and first column (default);
upright	right-hand plot at the top of the design i.e. in the first row and final column.

The PLOTORDER option defines the order in which the plots are then allocated:

colserpentine	column-by-column in a serpentine way e.g. top-to-bottom, then bottom to top, and so on;
colbycol	column-by-column taking the same direction for every column;
rowserpentine	in a serpentine way e.g. left-to-right, then right-to-left, and so on;
rowbyrow	row-by-row taking the same direction for every row (default).

The numbers of rows and columns defined by the NROWS and NCOLUMNS parameters, respectively. The PLOTNUMBER parameter saves the numbers, in a factor.

Options: FIRSTPLOT, PLOTORDER.
Parameters: NROWS, NCOLUMNS, PLOTORDER.

Method

The procedure first forms row and column factors using the GENERATE directive, with the FIRSTPLOT option indicating whether they are generated with their levels in ascending or descending order. The plot numbers are then calculated from those factors. With the serpentine orders, a different calculation is required for the even- and the odd-numbered rows or columns.

See also
Genstat Reference Manual 1 Summary section on: Design of experiments.

FPROJECTIONMATRIX

Forms a projection matrix for a set of model terms (R.W. Payne).

No options

Parameters

TERMS = *formula structures*	Defines the model terms corresponding to the design matrices whose projection matrices are required
PROJECTION = *symmetric matrices*	Saves the projection matrix for each formula structure

Description

This procedure forms the projection matrix for the design matrix of a set of model terms. The terms are specified in a model formula using the TERMS parameter, and the projection matrix is saved (as a symmetric matrix) by the PROJECTION parameter.

Options: none.
Parameters: TERMS, PROJECTION.

Method

The projection matrix is

```
X *+ GINVERSE( TRANSPOSE(X) *+ X ) *+ TRANSPOSE(X)
```

The design matrix X is formed using the TERMS directive, and the generalized inverse is formed using the GINVERSE procedure.

Action with RESTRICT

The factors must not be restricted, nor may they contain missing values.

See also

Procedures: ASWEEP, FRTPRODUCTDESIGNMATRIX.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, Calculations and manipulation.

FREGULAR

Expands vectors onto a regular two-dimensional grid (R.W. Payne).

Options

ROWS = *factor*	Original row factor
COLUMNS = *factor*	Original column factor
NEWROWS = *factor*	New row factor expanded onto the full grid
NEWCOLUMNS = *factor*	New column factor expanded onto the full grid
SORT = *string token*	Whether to sort the new values into row × column order (yes, no); default no

Parameters

OLDVECTOR = *variates*, *factors* or *texts*
Original data vectors

NEWVECTOR = *variates*, *factors* or *texts*
New vector with values, provided by the VALUES parameter, inserted in the units added to complete the grid

VALUES = *variates*, *scalars* or *texts*
Values to insert in the units added to complete the grid; default is to insert missing values

Description

FREGULAR is useful when you have data that have been recorded on a regular but incomplete two-dimensional grid: i.e. a you have rectangular grid of positions where observations could have been made but, at some of the points, nothing has been recorded.

An incomplete grid has disadvantages, for example, if you want to use REML to fit a spatial covariance model to one of the data variates. With data on a complete grid, you can fit various types of separable covariance model but, with an incomplete grid, only power distance-models are available (see VSTRUCTURE for details). So, if there are only a few empty positions, you might want to fill them in with missing values, and then set option MVINCLUDE = explanatory, yvariate in the REML command to include (and adjust for) these units in the analysis. Missing values are the default. Alternatively, if you know the values to insert, you can supply them using the VALUES parameter. You can supply a single value (in a scalar or a single-valued text, according to the type of the OLDVECTOR) if the new values are all the same. If they differ over the added units, they should be specified in row-column order

The ROWS and COLUMNS options supply factors identifying the row and column coordinates of the original data values. Note that, if there is more than one data value at any pair of coordinates, additional units are added so that there is the same number of units at each point. The NEWROWS and NEWCOLUMNS options can save factors with the new row and column coordinates, where additional units have been added to complete the grid.

The OLDVECTOR parameter specifies the original data vectors (variates, factors or texts). The NEWVECTOR parameter can supply the identifiers of vectors to store the new values (with missing values in the units added for the previously empty positions). If no NEWVECTOR is specified for an OLDVECTOR, the new values replace those in the OLDVECTOR.

By default, the units are left in their original order, with the new units added at the end. However, you can set option SORT=yes to sort them into ascending row × column order.

Options: ROWS, COLUMNS, NEWROWS, NEWCOLUMNS.
Parameters: OLDVECTOR, NEWVECTOR, VALUES.

Action with `RESTRICT`
Any restrictions are ignored.

See also
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FRESTRICTEDSET

Forms vectors with the restricted subset of a list of vectors (R.W. Payne).

Options

METHOD = *string token*	Whether to form the new vectors only when the old vectors are restricted or aways (always, whenrestricted); default alwa
RESTRICTED = *scalar*	Scalar set to 1 or 0 according to whether or not the old vectors are found to be restricted
VRESTRICTED = *variate*	Variate with each unit set to 1 or 0 according to whether or not that unit is restricted in any of the OLDVECTORS
SETLEVELS = *string token*	Whether to reform the levels (and labels) of factors to exclude those that do not occur in the restricted subset (yes, no); default no

Parameters

OLDVECTOR = *factors*, *variates* or *texts*	List of vectors, one or more of which may be restricted
NEWVECTOR = *factors*, *variates* or *texts*	New vectors which will contain only the unrestricted units of the old vectors

Description

The RESTRICT directive can be used in Genstat to associate a "restriction" with a vector, so that subsequent directives operate on only a subset of its units (see the *Guide to the Genstat Command Language*, Part 1, Section 4.4.1). This is very convenient, as the full set of data is still available and can be reinstated by cancelling the restriction. However, there are also occasions when it may be preferable to form a vector that contains only the restricted units. This is particularly true within procedures that may themselves need to apply restrictions, in addition to those already applied to their input vectors.

The OLDVECTOR parameter specifies the list of possibly-restricted vectors, and the NEWVECTOR parameter specifies a list of vectors to store the values from their unrestricted units. FRESTRICTEDSET first checks whether any of the old vectors is restricted and, if more than one is restricted, it gives a fault if the restrictions are not the same. The unrestricted set is defined to contain the units that are not restricted on any of the vectors. The RESTRICTED option can be set to a scalar which will be set to one or zero according to whether or not any OLDVECTOR was restricted, and the VRESTRICTED option can save a variate containing 0 in the restricted units and 1 in the unrestricted units. The METHOD option controls whether the new vectors are formed irrespective of whether or not the old vectors are restricted, or only if they are restricted. If the restricted subset excludes some of the levels of a factor, a new reduced set of levels (and labels) can be requested by setting option SETLEVELS=yes.

Options: METHOD, RESTRICTED, VRESTRICTED, SETLEVELS.
Parameters: OLDVECTOR, NEWVECTOR.

Method

FRESTRICTEDSET first uses directives such as GETATTRIBUTE, CALCULATE and GROUPS to check whether any vector is restricted, whether the restrictions are compatible and to form a logical variable to indicate which units are not excluded by the restriction. It then calls SUBSET to form the new vectors.

Action with `RESTRICT`
Restrictions on the old vectors are used to determine which of their units should be copied to the new vectors.

See also
Directive: `RESTRICT`.
Procedure: `SUBSET`.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Program control.

FRIEDMAN

Performs Friedman's nonparametric analysis of variance (S. Langton).

Options

PRINT = *string tokens*	Output required (test, ranks); default test
TREATMENTS = *factor*	Treatment factor
BLOCKS = *factor*	Block factor

Parameters

DATA = *variates*	Identifier of the variate holding the data values
RANKS = *variates*	Saves the ranks
STATISTIC = *scalars*	Saves the test statistic
DF = *scalars*	Saves the degrees of freedom for the chi-square approximation
PROBABILITY = *scalars*	Saves the probability value for the chi-square statistic

Description

Friedman's test is a nonparametric test for analysing a randomized complete block design. That is, the data consists of observations on k treatments assessed under n different conditions (blocks). The variate of observations is specified using the DATA parameter, whilst options TREATMENTS and BLOCKS supply the treatment and blocking factors. FRIEDMAN calculates the test statistic together with a probability value based on a chi-square approximation. If sample sizes are small, stored tabulated values are printed in addition.

The PRINT option controls printed output, with settings test to print the various test statistics, and ranks to print the ranks (together with the BLOCKS, TREATMENTS and DATA).

Parameters RANKS, STATISTIC, DF and PROBABILITY can be used to save the ranks, the test statistic (adjusted for ties), the degrees of freedom for the chi-square approximation, and the probability value for the chi-square approximation.

Options: PRINT, TREATMENTS, BLOCKS.
Parameters: DATA, RANKS, STATISTIC, DF, PROBABILITY.

Method

The Friedman test is a test for treatment differences in a randomized complete block design: i.e. a test of the null hypothesis that the samples arise from distributions with the same mean versus the alternative that the distribution means differ. Each block is checked in turn to ensure that it consists of exactly one replicate of each treatment, after excluding any units which are restricted out or which have missing values for DATA, TREATMENTS or BLOCKS. Any block not meeting this condition is excluded from analysis and a warning is printed. The treatments are ranked within each block and the sum of the ranks is calculated for each treatment group over all valid blocks. The sum of the squared values of these rank sums is calculated, as is the sum of the cubed sizes of all groups of ranks (i.e. 1 for an untied observation, 2^3=8 for pairs of ties, etc.).

The test statistic is formed using the equation (Siegel & Castellan 1988):

$$Fr = 12 \times R / (n \times k \times (k + 1)) - 3 \times n \times (k + 1)$$

where R is the sum of the squared rank sums,

k is the number of treatments, and

n is the number of blocks.

A version adjusted for ties is also formed, and this version is used for calculating the significance level:

$$Fr = \{12 \times R - 3 \times n^2 \times k \times (k + 1)^2\} / \{n \times k \times (k + 1) + (n \times k - T) / (k - 1)\}$$

where T is the sum of the cubed sizes of rank groups.

Action with RESTRICT

Any units that are restricted for DATA, BLOCKS or TREATMENTS (or which have missing values) are excluded from the analysis. Any block which no longer has one replicate of each treatment as a result of such restrictions is excluded in its entirety.

Reference

Siegel, S. & Castellan, N.J. (1988). *Nonparametric Statistics for the Behavioural Sciences (second edition)*. McGraw-Hill, New York.

See also

Procedures: APERMTEST, A2WAY, KRUSKAL.

Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Analysis of variance.

FROWCANONICALMATRIX

Puts a matrix into row canonical, or reduced row echelon, form (C.J. Brien).

Option

PRINT = *string token*	Controls printed output (rowcanonicalmatrix); default * i.e. none

Parameters

MATRIX = *matrices*	Matrix to be put into row canonical form
ROWCANONICALMATRIX = *identifiers*	Matrix in row canonical form

Description

A matrix is in row canonical, or reduced row echelon, form if the following conditions apply:

1) the leading coefficient in each row (i.e. its first non-zero element) is one,
2) the other elements in the column of each leading coefficient are zero, and
3) all rows that contain only zeros are at the bottom of the matrix.

The matrix to be put into row canonical form is specified by the MATRIX parameter. The resulting matrix, in that form, can be saved by the ROWCANONICALMATRIX parameter. It can be printed by setting option PRINT=rowcanonicalmatrix.

You can use the procedure to calculate the rank of a matrix, by counting the number of non-zero elements on the diagonal of its equivalent row canonical matrix. You can solve a set of consistent linear equations, by defining a matrix with the coefficients on the left-hand side of the assignments in its left-hand columns, and the values on the right-hand side of the assignments in its final column. You can also invert a square matrix, by appending the identity matrix to its right. These uses are demonstrated in the Example.

Option: PRINT.
Parameters: MATRIX, ROWCANONICALMATRIX.

Method

The row canonical form is produced by Gauss-Jordan elimination, which involves performing elementary row operations on the matrix. These operations are:

1) swap two rows,
2) multiply a row by a non-zero number, and
3) add a multiple of one row to another row.

The algorithm is based on that provided online by Rosetta Code (see http://rosettacode.org/wiki/Reduced_row_echelon_form).

See also

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FRTPRODUCTDESIGNMATRIX

Forms summation, or relationship, matrices for model terms (C.J. Brien).

No options

Parameters

TERM = *formula structures*	Model terms corresponding to design matrices whose summation matrices are required
MATRIX = *symmetric matrices*	Saves the summation or relationship matrix for each term

Description

This procedure forms the *summation*, or *relationship*, matrix for the design matrix of a model term. This is similar to the *projection* matrix, constructed by the FPROJECTIONMATRIX procedure. FPROJECTIONMATRIX calculates the mean of the units of a data variate with each effect of the term, and then forms a new data variate where each unit contains the mean calculated for the relevant effect. FRTPRODUCTDESIGNMATRIX calculates the sums of the data values instead of the means. The result is also a relationship matrix that indicates, using one and zero whether two units have the same levels of the factors in a term. One use for these matrices in synthesizing a variance matrix that involves a linear combination of variance components.

The term is specified (as a model formula) using the TERM parameter, and the matrix is saved (as a symmetric matrix) by the MATRIX parameter.

Options: none.
Parameters: TERM, MATRIX.

Method

The *summation* or *relationship* matrix is

```
D *+ TRANSPOSE(D)
```

The design matrix D is formed using the TERMS directive.

Action with RESTRICT

The factors must not be restricted, nor may they contain missing values.

See also

Procedure: FPROJECTIONMATRIX.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, Calculations and manipulation.

FSPREADSHEET

Creates a Genstat spreadsheet file (GWB or GSH) from specified data structures, PC Windows only (D.B. Baird).

Options

OUTFILE = *text*	Name of GSH file to store data in
SHEET = *number*	Sequence number of existing sheet, if this is set to 0 the data will be added to the first compatible spreadsheet open in the Windows interface
METHOD = *string token*	What to do with any existing columns with the same names as the new columns (replace, rename); default rena
READONLY = *string token*	Whether to make the complete sheet read-only (yes, no); default no
TITLE = *text*	The title associated with the spreadsheet
POINTER = *pointer* or *text*	A pointer or a name of a pointer to the columns in the spreadsheet
ANALYSIS = *text*	Genstat directives to analyse columns in the spreadsheet
ASETUP = *text*	Genstat directives to be run once before the analysis of any columns in the spreadsheet
ADUMMY = *text*	The name of the dummy (if any) used in the ANALYSIS directives
CURSOR = *variate*	A variate of length 2 giving the active cell position (x,y) when the spreadsheet is first displayed
NOUNITS = *string token*	Whether to stop the inclusion of a units column in the spreadsheet (yes, no); default no
BOOK = *number*	Window number of existing book, if this is set to 0 the sheet will be created in a new book, if to −1 it will be created in the last book formed with BOOK=0, and if set to −2 it will be created in the last book created in the Windows interface.
PAGENAME = *text*	The 32 character text to be displayed on the sheet tab
ROWCOLOURS = *factor*	The factor to be used for colouring the rows (the factor must have colours defined by the FACCOLOURS parameter)
TABLEFORMAT = *string token*	The format to use when displaying tables with two or more classifying factors (page, column); default page
FILEFORMAT = *string token*	The format to use for the spreadsheet file (GWB, GSH); default GWB
MARGINNAME = *text*	The 60 character text to be displayed for the margin labels
FROZENCOLUMNS = *scalar*	The number of columns to freeze on the left hand side of the spreadsheet; default 0 i.e. none

Parameters

DATA = *identifiers*	Data to write to the spreadsheet
PROTECT = *string tokens*	Whether to protect each data column by making it read-only (yes, no); default no
FACCOLOURS = *variates, texts* or *pointers*	Specifies background colours for factor columns
FOREGROUND = *variate, text, scalar* or *pointer*	

BACKGROUND = *variate, text, scalar or pointer*	Specifies foreground colours for columns Specifies background colours for columns
HIDDEN = *string tokens*	Whether to hide each DATA column (yes, no); default no

Description

FSPREADSHEET can be used to create a new spreadsheet or spreadsheet file, or to update a spreadsheet already open in the windows interface.

The DATA parameter lists the data structures to put into the spreadsheet. FSPREADSHEET regards the following structures as compatible:

- variates, factors or texts with identical lengths and restrictions,
- one-way tables with the same classifying factor, and
- multi-way tables with the same classifying factors (when option TABLEFORMAT=column), and
- scalars.

Structures that are compatible with each other are put into a single page of the spreadsheet. Matrices, diagonal matrices, symmetric matrices and incompatible structures go into separate pages. Multi-way tables (with two or more dimensions) also use separate pages when option TABLEFORMAT=page (see below). If the spreadsheet is being displayed in the Genstat Client, the structures are sorted into compatible groups each of which is displayed on a separate page.

The TABLEFORMAT option controls how tables with two or more classifying factors are displayed, with settings:

page	to put each table onto a separate page, with the last classifying factor displayed across the columns, and
column	to put each table into a single column, so that several tables are displayed on a single page.

The default is TABLEFORMAT=page.

By default, FSPREADSHEET writes the spreadsheet in the more recent GWB format, but you can set option FILEFORMAT=gsh to use the older GSH format instead.

If vectors unit labels, they will be included by default as the initial column of the spreadsheet. However, you can set option NOUNITS=yes to exclude them.

You can set option READONLY=yes to make the entire spreadsheet read-only (so that its contents cannot be changed). Alternatively, you can use the PROTECT parameter to protect any individual column (by making it read-only). Settings of the PROTECT parameter override the setting of the READONLY option. You can use set the HIDDEN parameter to yes to hide columns. The FROZENCOLUMNS option allows you to specify a number of columns on the left, that must not scroll off the screen when you scroll to the right.

If OUTFILE is set, the output is sent to the specified file; otherwise a new spreadsheet containing the data is formed and displayed in the Genstat Client. The SHEET and METHOD options are for updating open spreadsheets.

The number provided by SHEET is the position of the spreadsheet in the list of currently open spreadsheets. Thus SHEET=1 will add or update data in the first spreadsheet in the window list, SHEET=2 the second etc. Setting SHEET=0 will cause Genstat to update the first sheet with matching stuctures (i.e. for a variate this will be a VECTOR sheet with the same number of rows). The Genstat interface uses internal pointers to the spreadsheet structures which appear as large integers starting from 100000, and these should not be re-used in your saved code as they depend on how many spreadsheets have been opened from the start of the session. The BOOK option allows you to specify the particular book that the sheet is to be created in (if SHEET is not set). Setting BOOK=0 will cause the sheet to be placed in a new book, even if the default option in the Windows interface is to add sheets from the server to the last book with focus, and

setting BOOK=-1 will cause the sheet to be added to the last book created by FSPREADSHEET with BOOK=0. Setting BOOK=-2 will cause the sheet to be added to the last created spreadsheet. The PAGENAME option allows the tab name displayed when multiple sheets are in a book to be specified, and the MARGINNAME option specifies the labels to use for the margins of tables.

The METHOD option controls what happens when new columns have the same names as existing columns of the spreadsheet, with settings:

replace	to replace existing columns by new columns that have the same name, or
rename	to retain the old columns, and rename new columns whose names are the same as those of existing columns.

By default METHOD=rename.

The TITLE option can supply textual information (e.g. a title or a description) to be stored with the spreadsheet. The CURSOR option specifies the current cell to have focus when the spreadsheet is opened in a window. The POINTER, ANALYSIS, ASETUP and ADUMMY options allow Genstat directives to be attatched to the spreadsheet for use in the Spread > Sheet > Analysis menu.

The colours displayed in the cells of a spreadsheet can be controlled by using the FOREGROUND and BACKGROUND parameters to specify the foreground and background colours of the cells in each column. The setting can contain either colour names or RGB values (see the PEN directive for details). You can specify a scalar or a text of length one if all the cells in a column have the same colour. You can specify a variate or text with several values to define different a colour for each cell. Finally you can specify a single pointer to a set of variates or texts if the corresponding DATA setting will need several columns in the spreadsheet. Alternatively, you can specify the background columns for factor columns using the FACCOLOURS parameter. This should be set to a variate or text with the same number of values as the number of levels of the factor. You can apply the colours defined for background of each cell of a factor to the cell's complete row by setting the ROWCOLOURS option to the identifier of the factor. A missing value, empty string or undefined setting for any of these parameters will retain the default colour for the foreground or background.

FSPREADSHEET cannot create GWB files containing more than one sheet. To add further sheets to a GWB file, you can use EXPORT with option METHOD=add.

Options: OUTFILE, SHEET, METHOD, READONLY, TITLE, POINTER, ANALYSIS, ASETUP, ADUMMY, CURSOR, NOUNITS, BOOK, PAGENAME, ROWCOLOURS, TABLEFORMAT, FILEFORMAT, MARGINNAME, FROZENCOLUMNS.

Parameters: DATA, PROTECT, FACCOLOURS, FOREGROUND, BACKGROUND, HIDDEN.

Method

Internal directives are used to write the data to a GWB or GSH file which holds the data in a binary format. A message is sent to the Windows interface to read this file if OUTFILE is not set.

Action with RESTRICT

Restrictions on the structures are included in the spreadsheet created. If the restrictions on the structures are not consistent, a fault will occur.

See also

Directive: SPLOAD.

Procedures: TABTABLE, EXPORT, IMPORT, DDEEXPORT, DBEXPORT.

Genstat Reference Manual 1 Summary section on: Input and output.

FSTRING

Forms a single string from a list of strings in a text (R.W. Payne).

No options

Parameters

TEXT = *texts*	Texts containing the lists of strings to put into single strings
STRING = *texts*	Text to store the strings in each TEXT
SEPARATOR = *texts*	Characters to separate all except last two strings of each TEXT; default ', '
LASTSEPARATOR = *texts*	Characters to separate last two strings of each TEXT; default SEPARATOR
PREFIX = *texts*	Characters to insert at the start of each STRING; default '' (i.e. none)
END = *texts*	Characters to put at the end of each STRING; default '' (i.e. none)

Description

This procedure forms a string from a list of strings. The strings are input in a text specified by the TEXT parameter, and the string is saved using the STRING parameter. The SEPARATOR parameter allows you to specify the characters to separate the strings. The default is to use the characters ', '. The LASTSEPARATOR parameter allows you to supply a different set of characters to separate the last pair of strings. The PREFIX parameter can supply characters to put at the start of the STRING, and the END parameter can supply characters to put at the end.

Options: none.
Parameters: TEXT, STRING, SEPARATOR, LASTSEPARATOR, PREFIX, END.

Method

The output STRING is formed from the input TEXT using CONCATENATE.

Action with RESTRICT

If the TEXT is restricted, the STRING will be formed only with the units not excluded by the restriction.

See also

Directive: TXCONSTRUCT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FTEXT

Forms a text structure from any Genstat data structure (A. Keen & J.T.N.M. Thissen).

Option

MISSING = *text*	What to print for missing value; default '*'

Parameters

STRUCTURE = *identifiers*	Structure (scalar, variate, factor, text, table, matrix, symmetricmatrix, diagonalmatrix, pointer) from which the text structure is to be formed
TEXT = *texts*	Saves the text structure
DECIMALS = *scalars*	Number of decimals to use when forming the text structure; default * uses the number required to provide 4 significant figures, but unnecessary trailing zeros are ignored
FREPRESENTATION = *string tokens*	How factor values are to be represented in the text structure (labels, levels, ordinals); default is to use labels if available and levels otherwise

Description

Procedure FTEXT can be used to form a text structure from almost any Genstat data structure. The structure from which the text structure is to be formed must be specified using the STRUCTURE parameter, and can be a scalar, factor, variate, text, table, matrix, symmetricmatrix, diagonalmatrix or pointer. The identifier of the text structure must be specified using the TEXT parameter. For numerical structures, the number of decimal places to use when forming the text can be set by means of the DECIMALS parameter. By default the number required to provide four significant figures is used, but unnecessary trailing zeros are ignored. For example, a scalar containing the number 3.250 will be printed with two decimal places, not three. If the STRUCTURE parameter is set to a factor, parameter FREPRESENTATION can be used to control the way in which factor values are represented in the text structure. The default is to use labels if available and levels otherwise. If FREPRESENTATION is set to ordinals, the DECIMALS parameter is always set to zero for the corresponding factor.

The MISSING option allows you to specify a string to be used instead of the default asterisk symbol to represent missing values. For example, you could set MISSING='unknown' or MISSING=' '.

Option: MISSING.
Parameters: STRUCTURE, TEXT, DECIMALS, FREPRESENTATION.

Method

The text is formed using the PRINT directive with option CHANNEL=TEXT. The calculation of a default number of decimals places uses the same method as in the DECIMALS procedure.

Action with RESTRICT

If the STRUCTURE parameter has been restricted (as is possible only for a variate, factor or text) the length of the text structure will be the length of the restricted structure. The number of decimal places is determined from the values not excluded by the restriction.

See also
Directive: TXCONSTRUCT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FUNIQUEVALUES

Redefines a variate or text so that its values are unique (R.W. Payne).

Options

INCREMENT = *scalar*	Increment to use to modify duplicated numbers; default * i.e. a suitable (small) value is determined automatically
ADDTO = *string token*	Whether to add the increment to the value or the absolute value of duplicated numbers (value, absolutevalue); default abso

Parameters

OLDVECTOR = *variates* or *texts*	Vectors whose values are to be made unique
NEWVECTOR = *variates* or *texts*	New vectors with unique values; if unset, the values of the corresponding OLDVECTOR are replaced
CHANGED = *scalars*	Indicates whether the values have changed

Description

FUNIQUEVALUES allows you to redefine a variate or a text so that its values become unique. The variates or texts are listed by the OLDVECTOR parameter, and the NEWVECTOR parameter can save new variates or texts with the redefined values. If no NEWVECTOR is defined for one of the variates or texts in the OLDVECTOR list, the new values are put into the original variate or text. The CHANGED parameter can save a scalar that set to one if the values were changed; otherwise it is set to zero.

FUNIQUEVALUES appends the characters '_1', '_2' and so on to each duplicate value of a text. By default, with a variate, FUNIQUEVALUES adds a small increment to each zero or positive duplicate value, and subtracts that increment from each negative duplicate value. If you would prefer to add the increment to both positive and negative values, you can set option ADDTO=value. (This indicates that the increment is to be added to the value, rather than the absolute value of the duplicate number.) The default increment is taken as the largest power of 10 that is small enough to modify each duplicate value while preserving their numerical order. So, for example, if you had values 1, 2, 3, 2, 5 and 2, the increment would be 0.1; the second instance of 2 would become 2.1, and the third would become 2.2. As another example, if the values were 0.1, 0.2, 0.3, 0.2, 0,5 and 0.2, the increment would be 0.01; the second instance of 0.2 would then become 0.21, and the third would become 0.22. Alternatively, you can supply your own increment using the INCREMENT option.

Options: INCREMENT, ADDTO.
Parameters: OLDVECTOR, NEWVECTOR, CHANGED.

Action with RESTRICT

Any restrictions are ignored.

See also

Procedure: FACUNIQUE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FVCOVARIANCE

Forms the variance-covariance matrix for a list of variates (W. van den Berg).

Options

PRINT = *string tokens*	Printed output (df, vcovariance); default df, vcov
WEIGHTS = *variate*	Provides weights for the units of the variates; default * assumes that they all have weight one
VCOVARIANCE = *symmetric matrix*	Saves the variance-covariance matrix
DF = *scalar*	Saves the number of degrees of freedom of the (co)variances

Parameter

DATA = *variates*	Variates for which the matrix is to be calculated

Description

FVCOVARIANCE calculates vcovariance matrices for a set of variates which should be listed by the DATA parameter. The WEIGHTS option can provide a variate of weights for the units of the variates; by default these are all assumed to have weight one. The vcovariance matrix can be saved using the VCOVARIANCES option, and the degrees of freedom can be saved using DF option.

Printed output is controlled by the PRINT option with settings:

vcovariance	prints the variance-covariance matrix,
df	prints the degrees of freedom.

By default both are printed.

Options: PRINT, WEIGHTS, VCOVARIANCES, DF.
Parameter: DATA.

Method

A SSPM structure is set up for the DATA variates, and its values are formed using the FSSPM directive. The corrected sums of squares and products are divided by the residual degrees of freedom to give the variance covariance matrix.

Action with RESTRICT

FVCOVARIANCE takes account of restrictions on any of the variates.

See also

Directive: FSSPM.
Procedures: FCORRELATION, ROBSSPM.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

FVSTRING

Forms a string listing the identifiers of a set of data structures (R.W. Payne).

Options

STRING = *text*	Saves the string
POINTERNAME = *text*	If all the structures are belong to the same pointer, this saves its name
ELEMENTNAMES = *text* or *variate*	Saves the elements of the pointer, in a text if they have labels, otherwise in a variate

Parameter

DATA = *identifiers*	Data structures to be used to form the string

Description

This procedure forms a string listing the identifiers of the data structures specified by the DATA parameter. This is used, for example, by the AREPMEASURES procedure when it appends variates of observations made at successive times into a combined variate for the analysis. FVSTRING is used to form a string listing the variates, and this is then used as the "extra" text for the combined variate. This extra text is displayed in the output from directives like ANOVA. Users can thus see what is actually being analysed.

If the data structures do not belong to a common pointer, the string simply contains their identifiers separated by commas e.g. `x,y,z` for structures `x`, `y` and `z`. However, if they do belong to a pointer, FVSTRING produces a more succinct string e.g. `p[1,3,5]` for `p[1]`, `p[3]` and `p[5]`, or `p['a','c','e']` for `p['a']`, `p['c']` and `p['e']`. The name of the common pointer can be saved by the POINTERNAME option, and the names (numbers or strings) of the elements can be saved by the ELEMENTNAME option.

Options: STRING, POINTERNAME, ELEMENTNAMES.
Parameter: DATA.

See also

Directive: TXCONSTRUCT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

†FWITHINTERMS

Forms factors to define terms representing the effects of one factor within another factor (R.W. Payne).

Options

LEVNULL = *scalar*	Numerical value to represent the null level assigned to units not involved in the comparison of the levels of one of the factors within a particular level of the other factor; default 0
LABNULL = *text*	String to label the null level; default '-'

Parameters

F1 = *factors*	First factor
F2 = *factors*	Second factor
F1WITHINF2 = *pointers*	Pointer containing a factor for each level of the second factor, used to estimate the effects of the first factor within that level
F2WITHINF1 = *pointers*	Pointer containing a factor for each level of the first factor, used to estimate the effects of the second factor within that level

Description

When there is an interaction between two factors, it may be interesting to compare the effects of one factor within each level of the other factor. FWITHINTERMS forms factors to define terms to enable you to do this.

The two factors are specified by the F1 and F2 parameters. The F1WITHINF2 parameter saves a pointer containing a factor for each level of the F2 factor, to be used to assess the effects of F1 within that level. To do this you can specify a model formula of

```
F2 / F1WITHINF2[]
```

The term F2 represents the main effect of factor F2. Subsequent terms like F2.F1WITHINF2[i] represent the effects of F1 within level i of factor F2. To help make the output clear, the elements of F1WITHINF2 are labelled by the levels of F2 (or its labels if available). Each factor has the levels of F1 for the units with its corresponding level of F2, and a "null" level elsewhere. The numerical value to represent the null level is specified by the LEVNULL option (default 0). If F1 has labels, the label for the null level is specified by the LABNULL option (default '-').

Similarly, the F2WITHINF1 parameter saves a pointer containing factors to assess the effects of F2 within F1.

Options: LEVNULL, LABNULL.
Parameters: F1, F2, F1WITHINF2, F2WITHINF1.

See also

Procedure: FBETWEENGROUPVECTORS.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Analysis of variance.

FZERO

Gives the F function expectation under complete spatial randomness (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

DENSITY = *scalars*	Densities to use i.e. numbers of points per unit area; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
FVALUES = *variates*	Variates to receive the expected values of the F nearest-neighbour distribution function under CSR

Description

The F nearest-neighbour distribution function relates to the distribution of distances from each of a set of sample points covering the region of interest to the nearest event of an observed spatial point pattern (see Diggle 1983). (The procedure FHAT can be used to obtain an estimate of F given an observed pattern and a set of sample points.) The term complete spatial randomness (CSR) is used to represent the hypothesis that the overall density of events in a spatial point pattern is constant throughout the study region, and that the events are distributed independently and uniformly.

Under CSR, and ignoring edge effects, the F nearest-neighbour distribution function is given by

$$F(s) = 1 - \exp(-i \times density \times (s^2)),$$

where *density* is the overall density of events per unit area. (The F nearest-neighbour distribution function for a clustered (regular) pattern will tend to be smaller (larger) than values calculated using the above expression, at least for small distances.) The procedure FZERO evaluates this expression for a given density (specified using the parameter DENSITY) and a vector of distances (specified using the parameter S). (The procedure PTINTENSITY may be used to obtain the density of events in an observed pattern prior to using FZERO.) The output of the procedure is a vector containing the expected values of F under CSR for each distance in S. The values of the F function can be saved using the parameter FVALUES.

Printed output is controlled using the PRINT option. The default setting of summary prints the distances at which the F function is estimated, and the estimates themselves under the headings S and FVALUES.

Another nearest-neighbour distribution function, the G nearest-neighbour distribution function, relates to the distribution of distances from each event of a spatial point pattern to the nearest other event in the pattern (see Diggle 1983). (The procedure GHAT can be used to obtain an estimate of G for an observed pattern.) Under CSR, the F and G nearest-neighbour distribution functions are identical. The output from the procedure FZERO can, therefore, be compared to estimates of the G nearest-neighbour distribution function. (The G nearest-neighbour distribution function for a clustered (regular) pattern will tend to be larger (smaller) than the values given by the above expression, at least for small distances.)

Option: PRINT.
Parameters: DENSITY, S, FVALUES.

Method
The CALCULATE directive is used to evaluate the expression for the expected value of the F function using the density of events specified by the parameter DENSITY and the set of distances in S.

Action with RESTRICT
If S is restricted, only the subset of values specified by the restriction will be included in the calculations.

Reference
Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

See also
Procedures: FHAT, GHAT, KHAT, KSTHAT, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

F2DRESIDUALVARIOGRAM

Calculates and plots a 2-dimensional variogram from a 2-dimensional array of residuals (S.J. Welham).

Options

PLOT = *string token*	What to plot (surface); default surf
ROWS = *factor*	Factor defining the rows of the grid
COLUMNS = *factor*	Factor defining the columns of the grid
REPLICATES = *factor*	Factor defining the replicate grids (if any)
RMAX = *scalar*	Maximum lag to include in variogram in row direction (default determined by procedure)
CMAX = *scalar*	Maximum lag to include in variogram in column direction (default determined by procedure)
RSCALE = *scalar*	Actual distance represented by 1 unit in row direction (default 1)
CSCALE = *scalar*	Actual distance represented by 1 unit in column direction (default 1)
MINREP = *scalar*	Minimum replication required for position to be included in variogram (default 30)
TITLE = *text*	Title for surface/graph; default * i.e. none
WINDOW = *scalar*	Graphics window to be used for plotting; default 1
SCREEN = *string token*	Whether to keep or clear screen before plotting variogram (clear, keep); default clear
METHOD = *text*	Whether to use Fortran DLL or Genstat code to calculate variogram (dll, genstat); default dll
SCALEPLOT = *string token*	Whether to scale variogram to 0-1 (i.e. unit) scale for plotting (unit, none); default unit

Parameters

RESIDUALS = *variates*	Variate of residuals to form variogram
VARIOGRAM = *matrices*	Calculated variogram (trimmed)
FULLVARIOGRAM = *matrices*	Calculated variogram (all values)
COUNTS = *matrices*	Number of comparisons contributing to each variogram position
COMPONENTS = *pointers*	Components used to calculate variogram (only available when METHOD=genstat)

Description

F2DRESIDUALVARIOGRAM calculates and plots a 2-dimensional variogram from a 2-dimensional array of residuals, i.e. residuals that come from an experiment with units arranged in a regular grid.

The data variate is specified by the RESIDUALS parameter, and factors defining the rows and columns of the grid are specified using the ROWS and COLUMNS options. The ROWS and COLUMNS should completely define the layout of the data, except in the case that there are replicates of a grid. In this case, the option REPLICATES should be used to specify a factor defining the replicates of the grid.

The full variogram is calculated for all row lag distances 1 ... NLEVELS(ROWS) and all column lag distances 1 ... NLEVELS(COLUMNS), unless constrained by options RMAX and CMAX, respectively. This full variogram can be saved using parameter FULLVARIOGRAM, and the number of comparisons contributing to each point can be saved in a corresponding matrix using parameter COUNTS. Missing values will be inserted into the full variogram where the number of

comparisons is less than set by option MINREP (default 30). Note that setting MINREP to low values may mean that spurious patterns appear in the variogram.

The full variogram is then trimmed to give a rectangle containing no missing values – this is the variogram that is plotted and can be saved as a matrix using parameter VARIOGRAM.

By default a surface plot is produced for a two-dimensional variogram, i.e. where the trimmed variogram has > 2 rows in both dimensions. If only one dimension has > 2 rows, then a graph will be plotted for that dimension. Note that setting RMAX=1 means that a one-dimensional variogram for COLUMNS can be produced, and vice versa (this can be useful for longitudinal data). Plotting can be suppressed by setting option PLOT=*. Within the plot, difference of scale between row and column directions can be represented by specifying the actual distances represented by each row and column unit using the RSCALE and CSCALE options respectively. The variogram will be scaled onto a 0-1 scale for plotting unless specified by SCALEPLOT=none. A title for the plot can be specifed using the TITLE option. The graphics window used for plotting and whether to keep/clear the screen before plotting are controlled by the WINDOW and SCREEN options as usual.

For large grids, calculation of the variogram using Genstat code can be slow. For the PC Windows implementation a Fortran DLL is available to make the calculations much faster. By default, the procedure finds out if the DLL is available, and if so, uses it. Otherwise Genstat cdoe within the procedure is used. The procedure can be forced to use the Genstat code by setting option METHOD=genstat.

When the Genstat code within the procedure calculates the variogram, the components of the variogram can be saved in a pointer specified by the COMPONENTS option.

Options: PLOT, ROWS, COLUMNS, REPLICATES, RMAX, CMAX, RSCALE, CSCALE, MINREP, TITLE, WINDOW, SCREEN, METHOD, SCALEPLOT.
Parameters: RESIDUALS, VARIOGRAM, FULLVARIOGRAM, COUNTS, COMPONENTS.

Method

The sample variogram position (i, j) is calculated as the average of the half-squared differences between all pairs of residuals i row units and j column units apart.

The variogram is trimmed so that the number of pairs of points contributing to each variogram postion is greater than the setting of MINREP.

Action with RESTRICT

F2DRESIDUALVARIOGRAM takes account of any restriction on RESIDUALS, subject to the factors ROWS, COLUMNS and REPLICATES still defining a valid grid of residuals.

See also

Procedures: DVARIOGRAM, DCOVARIOGRAM, DRESIDUALS, VPLOT.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

GALOIS

Forms addition and multiplication tables for a Galois finite field (I. Wakeling & R.W. Payne).

Option

METHOD = *string token*	Whether to choose the primitive polynomial to generate the Galois field with the least number of higher terms or whether to make a random choice (minimal, random); default rand

Parameters

ORDER = *scalars*	Order of the required Galois field
ADDITION = *symmetric matrices*	Saves the addition table of the field
MULTIPLICATION = *symmetric matrices*	Saves the field's multiplication table
PRIMITIVE = *variates*	Saves the primitive irreducible polynomial
ERROR = *scalars*	Returns 0 or 1 according to whether or not the tables have been formed successfully

Description

Procedure GALOIS generates the addition and multiplication tables for a Galois finite field. The order of the field, p^n, is specified by the ORDER parameter and must be a power of a prime number p. If $n>1$, the elements of the field can be considered as polynomials of degree $(n-1)$ whose coefficients can be any residue modulo p. The multiplication and addition tables can be saved using the ADDITION and MULTIPLICATION parameters. These return pointers to n symmetric matrices containing the coefficients for the polynomials. (The first element of the pointer is the constant term, the last element is the x^{n-1} term). In order to construct the field, a primitive polynomial of degree n is needed (see the Method section for more details). The coefficients of this polynomial can be saved using the PRIMITIVE parameter. Again the first element corresponds to the constant term and the last element to the highest power. The final parameter, ERROR, can be set to a scalar which will return the value one if it has not been possible to form the tables, for example, because ORDER is not a valid prime power. If the tables have been formed successfully, ERROR is set to zero.

Option: METHOD.
Parameters: ORDER, ADDITION, MULTIPLICATION, PRIMITIVE, ERROR.

Method

Procedure PRIMEPOWER is used to decompose ORDER into prime powers (p^n). For prime values of ORDER, i.e. when the exponent n is one, the finite field is equivalent to the operations of addition and multiplication modulo p; in which case the solution is trivial. Otherwise, when ORDER is a prime power ($n>1$ and p prime), a solution is found by representing the elements of the field by ordered n-tuples of integers modulo p, For example the order 4 field has elements (00, 01, 10, 11) which may alternatively be thought of as polynomials e.g. (10) as $x+0$ and (11) as $x+1$. The method first generates a list of these elements and then forms all possible pairwise products between the polynomials (reducing by modulo p where necessary). All polynomials formed in this way may therefore be factorized over an order p field. By elimination, a list of polynomials of degree n that may not be factorized is created. These are said to be irreducible and have been tabulated in the mathematical literature (Williams 1996, Section 2.8). The standard method of construction of the field is to select one of these irreducible polynomials f(x), which is also primitive; meaning that the non-zero elements of the field form a cyclic group under multiplication. See Williams (1996) Section 2.9 for a table of primitive polynomials.

GALOIS either chooses f(x) randomly from the set of all possible primitive polynomials (METHOD=random), or chooses the polynomial having the minimal number of higher terms (METHOD=minimal). The final choice of f(x) is made available to the user through the parameter PRIMITIVE. For further information on finite fields in a statistical context the reader should consult Street & Street (1987).

References

Street, A.P. & Street, D.J. (1987). *The Combinatorics of Experimental Design*. Clarenden Press, Oxford.

Williams, H. (1996). Number theory and finite fields. In: *The CRC Handbook of Combinatorial Designs*, (ed. C.J. Colburn & J.H. Dinitz), 615-644. CRC Press, Boca Raton, Florida.

See also

Procedures: AGLATIN, FHADAMARDMATRIX, PRIMEPOWER.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Design of experiments.

GBGRIDCONVERSION

Converts GB grid references to or from latitudes and longitudes or to or from UTM coordinates (R.W. Payne).

Options

INPUTSOURCE = *string token*	Which of the coordinate systems if acting as input for conversion to either of the other two systems (gridreference, geographical, utm); default geog
GRIDREFERENCES = *texts*	Grid references
LATITUDES = *scalar*s or *variates*	Latitudes
LONGITUDES = *scalars* or *variates*	Longitudes
EASTINGS = *scalars* or *variates*	UTM easting references
NORTHINGS = *scalars* or *variates*	UTM northing references
GRIDACCURACY = *string token*	The accuracy for saving grid references (kilometres, hectometres, dekametres, metres); default hect

No parameters

Description

GBGRIDCONVERSION can convert between three coordinate systems: firstly British grid references, secondly geographical latitude and longitude coordinates, and thirdly eastings and northings in the Universal Transverse Mercator (UTM) coordinate system; see UTMCONVERSION procedure for details. The grid references are supplied or saved, in a text, by the GRIDREFERENCES option. The other coordinates are supplied or saved by the LATITUDES, LONGITUDES, EASTINGS and NORTHINGS options. These can be scalars for a single location, or variates for several.

The INPUTSOURCE option specifies which of these is to provide the input for conversion to the other two, with settings:

gridrefercences	for GRIDREFERENCES,
geographical	for LATITUDES and LONGITUDES, and
utm	for EASTINGS and NORTHINGS.

The GRIDACCURACY option controls the accuracy of the grid references that are formed i.e. how may digits are used. The available settings are:

kilometres	saves references with two letters and four digits, giving accuracy to the nearest kilometre;
hectmetres	saves references with two letters and six digits, giving accuracy to the nearest hectometre (i.e. 100 metres);
dekametres	saves references with two letters and eight digits, giving accuracy to the nearest dekametre (i.e. 10 metres);
metres	saves references with two letters and ten digits, giving accuracy to the nearest metre.

The default is hectmetres, giving the most common style, e.g. TL059066. A space is inserted in the middle of the references with eight and ten digits, e.g. TL05921 06642.

Options: INPUTSOURCE, GRIDREFERENCES, LATITUDES, LONGITUDES, EASTINGS, NORTHINGS. GRIDACCURACY.
Parameters: none.

Method

UKGRIDCONVERSION uses UTMCONVERSION to convert latitudes and longitudes to or from eastings and northings. The eastings and northings are converted to or from grid references using

the standard Genstat manipulation commands.

Action with `RESTRICT`
If any of the options `GRIDREFERENCES`, `LATITUDES`, `LONGITUDES`, `EASTINGS` or `NORTHINGS` is restricted, only the units not excluded by the restriction will be analysed.

See also
Procedure: `UTMCONVERSION`.

GEE

Fits models to longitudinal data by generalized estimating equations (D.M. Smith & M.G.Kenward).

Options

PRINT = *string token*	What to display (estimates, correlations, scalefactor, wald, monitoring); default esti, corr, scal
DISTRIBUTION = *string token*	Distribution of response (normal, Poisson, binomial, gamma, inversenormal, negativebinomial); default *
LINK = *string token*	Link function (identity, logarithm, logit, reciprocal, power, squareroot, probit, complementaryloglog, logratio); default *
EXPONENT = *scalar*	Exponent for power link; default −2
TERMS = *formula*	Explanatory variates, factors etc
CONSTANT = *string token*	How to treat constant (estimate, omit); default esti
FACTORIAL = *scalar*	Limit for expansion of model terms; default 3
AGGREGATION = *scalar*	Fixed parameter for negative binomial distribution (parameter k as in variance function var = mean + mean^2/k); default 1
KLOGRATIO = *scalar*	Parameter for logratio link, in form log(mean / (mean + k)); default as set in AGGREGATION option
QUADESTIMATION = *string token*	Whether to use quadratic estimation (used, notused); default used
SCALEFACTOR = *string token*	How to calculate the scale factor (fixed, constant, varytime); default varies with distribution, fixed for Poisson, binomial and negative binomial, constant for rest
SFVALUE = *scalar*	Value for scale factor when SCALEFACTOR=fixed; default 1.0 for Poisson and binomial, missing for rest
CRTYPE = *string token*	Form of correlation matrix (independence, unstructured, exchangeable, autoregressive, dependence, antedependence); default *
ORDER = *scalar*	Order in dependence and ante-dependence form of correlation matrix; default 1
TIMEDEPENDENT = *string token*	Whether correlation in dependence model changes with time (no, yes); default no

Parameters

Y = *variates*	Response variate for each analysis
NBINOMIAL = *variates* or *scalars*	Denominator in binomial
FITTEDVALUES = *variates*	To store fitted values
RESIDUALS = *variates*	To store residuals
SUBJECT = *factors*	Identifier of subjects
OUTCOME = *factors*	Identifier of outcomes
COUNT = *variates*	Variate of counts of no. outcomes
TIME = *factors*	Times of repeated measures variate
WEIGHT = *variates*	Weight variate
OFFSET = *variates*	Offset variate
SAVE = *pointers*	Structure to save output variables

Description

GEE implements the General Estimating Equation (GEE) methodology of Liang & Zeger (1986) with quadratic estimation for the covariance structure. In the terminology of Liang *et al*. (1992) the methodology implemented is a form of GEE1. Full details of the implementation are given in Kenward & Smith (1995a). GEE, as implemented here, is a comparatively simple non-likelihood method for fitting marginal models to repeated measurements that can be used when the response has a distribution in the exponential family. This includes the Gaussian distribution, for which the procedure implemented here reduces to a form of the EM algorithm, and then produces exact ML or REML estimates, or a close approximation to these depending on the particular correlation structure chosen. For other distributions the resulting estimates are not maximum likelihood but can be shown to have asymptotic properties familiar from quasi-likelihood, such as consistency and asymptotic normality.

The standard range of generalized linear models (as in procedure GLM) can be fitted involving a variety of covariance or correlation structures over the times of the repeated measurements. The standard links and distributions can be chosen by setting the options DISTRIBUTION, LINK, EXPONENT, AGGREGATION and KLOGRATIO, as in the MODEL directive. Non-standard ones require the definition of auxiliary procedures to carry out the necessary calculations (see the *Method* section). The terms in the fitted model are specified by the TERMS option, which may be set to a formula or left unset to fit a null model. The FACTORIAL option (default 3) sets a limit on the number of factors and variates in the terms that are fitted, as in the FIT directive. The CONSTANT option can be used to omit a constant term. Setting the QUADESTIMATION option to used requests the use of quadratic estimation for the data-based covariance or correlation matrix (see Kenward & Smith 1995a). The SCALEFACTOR option specifies the form of scalefactor to be used (fixed to a value specified by the SFVALUE option, constant over times of repeated measurements, or varying over times of repeated measurements). The CRTYPE option specifies the structure of the covariance or correlation matrix over the times of the repeated measurements. The ORDER option specifies the order of the covariance or correlation structures for the dependence and ante-dependence cases, with option TIMEDEPENDENT specifying whether the correlation in a dependence structure changes with the time of the repeated measurement.

The Y parameter must be set to specify the response variate. For a binomial distribution the NBINOMIAL parameter must also be set; this may be either a variate or a scalar (if it is a scalar, GEE maps it out automatically to a variate with the same number of values as Y). The SUBJECT parameter specifies a factor to identify the subjects. Alternatively, where the data consist of outcomes and numbers with those outcomes, the parameter OUTCOME must be set to the identifier of the outcome and the parameter COUNT to the number with the outcome. The parameter TIME must be set to the times of the repeated measurements. The parameters WEIGHT and OFFSET specify weight and offset variates that may be involved.

The output from the procedure is controlled by the PRINT option; by default estimates, their standard errors, covariances or correlations and scalefactors are given. Two sets of standard errors are provided for the estimates. One is the naive estimate which assumes the specified covariance or correlation structure holds. The other is the sandwich estimate which makes no such assumption. When PRINT=wald, Wald tests are produced using both sets of standard errors and correlations.

The fitted values and residuals can be obtained by setting the parameters FITTEDVALUES and RESIDUALS. The residuals are the Pearson residuals as defined in the *Guide to the Genstat Command Language*, Part 2, Section 3.1.1.

The SAVE parameter can save various details of the analysis, in a pointer with the following suffixes and labels:

1 or 'scalefactors' scalefactor(s),
2 or 'correlation' or 'covariance'

	correlations or covariances, according to the type of model (and labelled appropriately),
3 or 'estimates'	the estimates of the linear predictor parameters,
4 or 'naive covariances'	naive variance-covariance matrix for the estimates,
5 or 'sandwich covariances'	sandwich variance-covariance matrix for the estimates,
6 or 'naive Wald'	Wald tests calculated using the naive variance-covariance matrix, and
7 or 'sandwich Wald'	Wald tests calculated using the sandwich variance-covariance matrix.

The algorithms in the procedure have been set up assuming that the data contain a complete set of observations for each subject. Where there are missing values these must be included explicitly (using the missing value symbol *) to create a complete set of observations. Missing values are allowed in both the Y variate and the explanatory variates in TERMS.

In the case of the Gaussian distribution, a working covariance matrix, rather than correlation matrix, is used. This provides considerable simplification within the algorithm.

This is a complicated algorithm and some examples may take a while to run. If necessary, however, you can set option PRINT=monitoring to see what is happening.

Options: PRINT, DISTRIBUTION, LINK, EXPONENT, TERMS, CONSTANT, FACTORIAL, AGGREGATION, KLOGRATIO, QUADESTIMATION, SCALEFACTOR, SFVALUE, CRTYPE, ORDER, TIMEDEPENDENT.

Parameters: Y, NBINOMIAL, FITTEDVALUES, RESIDUALS, SUBJECT, OUTCOME, COUNT, TIME, WEIGHT, OFFSET, SAVE.

Method

For full details of the method implemented in this procedure see Kenward & Smith (1995a). A generalized linear model is formulated for the marginal distribution of the observations at each time point using an appropriate link function and error distribution. If the repeated measurements could be assumed to be independent, the well-known iterative weighted least squares fitting procedure could be used to obtain ML estimates of the marginal model parameters. However this ignores the dependence among the repeated measurements. Full likelihood is in general very awkward in this setting so, to avoid a formal introduction of dependence into the model, a working correlation matrix is introduced into the iterative procedure, changing the least squares from a weighted to a generalized form. The correlation matrix can be introduced in various ways. It can be held constant throughout the iterative procedure. An example of this is the use of the identity matrix, leading to the so-called independence estimating equations for which the process reduces back to that of fitting a univariate generalized linear model. Alternatively an estimated correlation matrix can be introduced into the algorithm which is updated at each cycle using quadratic estimation: essentially the correlation structure is estimated from the residuals using the equations that would be appropriate were the residuals normally distributed. On convergence consistent estimates of the marginal linear model parameters are obtained and, if the correlation structure chosen is appropriate, then this will be consistently estimated as well. It is not necessary for the correlation structure to be correct for the consistency of the marginal parameter estimates, at least when the correlation structure is fixed; indeed the common choice of independence is almost certain not to be appropriate. However the estimates of precision of the marginal parameter estimates do need to be adjusted to allow for the true correlation structure. This correction is done in the so-called "sandwich" estimator provided by the procedure.

The procedures have been written so that it is possible to fit models other than the standard ones. An important example of such a model is the application of the GEE methodology to ordinal categorical data. This application requires the data to be arranged in a particular form (as cummulative logits) and a particular correlation matrix (specified in _GEECORRELATION). The

type of analyses are explained in Kenward *et al*. (1994) and the methodology described in that paper has been duplicated. Further details are given in Kenward & Smith (1995b).

An option (SCALEFACTOR) has been included that allows the user to decide whether or not the scale factor is fixed at its independence distributional default, or is estimated from the scaled residuals as in Liang & Zeger (1986), or is treated as a vector varying over time.

GEE calls three subsidiary procedures, _GEECODI, _GEECALLIN and _GEECALDIS to assist with the analysis. There is no need for the user to modify these procedures:

_GEECODI	prints out the results of the iterative processes;
_GEECALLIN	calculates the fitted values and derivatives for various links;
_GEECALDIS	calculates the variance function and deviance for various distributions.

There are also four other procedures, which can be re-written or replaced, to cater for further user-defined distributions, links and correlation structures:

_GEEINIT	calculates initial estimates of the linear predictor in the generalized linear model;
_GEELINK	calculates fitted values and derivatives;
_GEEDISTRIBUTION	calculates the variance function and deviance;
_GEECORRELATION	calculates the correlation matrix and the sandwich matrix involving the residuals. (For the normal distribution the variance-covariance matrices are used not the correlation matrices.)

If the LINK option is unset, the procedure will call _GEEINIT and _GEELINK instead of using those for the various standard link functions. For a logit link function _GEEINIT and _GEELINK should be defined as follows.

```
PROCEDURE '_GEEINIT'
          "Calculation of initial estimate of linear predictor,
           link unset"
PARAMETER NAME = \
          'Y',              "I: variate; response variate"\
          'LINEARPREDICTOR',"O: variate; linear predictor"\
          'OFFSET',         "I: variate; offset"\
          'NBINOMIAL';      "I: variate; denominator of binomial"\
          SET=3(yes),no;TYPE=4('variate'); \
          COMPATIBLE=*,3(!T(type,nvalues,restriction));\
          PRESENT=yes,no,2(yes)

CALC LINEARPREDICTOR = LOG((Y+0.5)/(NBINOMIAL-Y+0.5)) - OFFSET
ENDPROCEDURE

PROCEDURE '_GEELINK'
          "Calculation of fitted values and derivatives"
PARAMETER NAME = \
          'LINEARPREDICTOR', "I: variate; linear predictor"\
          'FITTEDVALUES',    "O: variate; estimate of fitted values"\
          'DERIVATIVES',     "O: variate; estimate of derivatives"\
          'OFFSET',          "I: variate; offset"\
          'NBINOMIAL';       "I: variate; denominator of binomial"\
          SET=4(yes),no;TYPE=5('variate'); \
          COMPATIBLE=*,4(!T(type,nvalues,restriction));\
          PRESENT=yes,2(no),2(yes)

GETATTRIBUTE [ATTRIBUTE=NVALUES] LINEARPREDICTOR; SAVE=!P(nobs)

CALC  FITTEDVALUES = NBINOMIAL/(1+EXP(-LINEARPREDICTOR - OFFSET))
&     DERIVATIVES = 1/FITTEDVALUES+1/(NBINOMIAL-FITTEDVALUES)
ENDPROCEDURE
```

If the DISTRIBUTION option is unset, the procedure will call _GEEDISTRIBUTION instead of

using one of the various standard distributions. For a binomial error distribution _GEEDISTRIBUTION should be defined as follows.

```
PROCEDURE '_GEEDISTRIBUTION'
          " Calculation of variance function and deviance"
PARAMETER NAME = \
          'Y',           "I: variate; response variate"\
          'FITTEDVALUES',"I: variate; fitted values"\
          'VARIANCE',    "O: variate; variance"\
          'DEVIANCE',    "O: scalar; total deviance"\
          'NBINOMIAL';   "I: variate; denominator of binomial"\
          SET=4(yes),no;TYPE=3('variate'),'scalar','variate'; \
          COMPATIBLE=*,2(!T(type,nvalues,restriction)),*,\
                     !T(type,nvalues,restriction); \
          PRESENT=2(yes),2(no),yes

CALC VARIANCE = FITTEDVALUES*(NBINOMIAL-FITTEDVALUES)/NBINOMIAL
&    DEVIANCE = -2*LLB(Y;NBINOMIAL;(FITTEDVALUES/NBINOMIAL))
ENDPROCEDURE
```

If the CRTYPE option is unset, the procedure will call _GEECORRELATION instead of using one of the various standard correlation models. For the independence model _GEECORRELATION should be defined as follows. Kenward & Smith (1995b) describe how _GEECORRELATION should be set up for analysing repeated ordinal categorical data.

```
PROCEDURE   '_GEECORRELATION'
            "
              Calculation of correlation matrix

              For SANDWICH = NO
                  input is the R matrix as for UNSPECIFIED
                  output is the desired R matrix.
              For SANDWICH = YES
                  input is the (Y-MU)*T(Y-MU) matrix
                  output is the desired modified (Y-MU)*T(Y-MU) matrix.

              N.B. For the normal distribution both the input and
                   output R's should be variance/covariance matrices
                   not correlation matrices.
            "
OPTION      NAME = \
            'CONSTANT', "I: text; how to treat constant (estimate,
                        omit); default e"\
            'SANDWICH'; "I; text; whether the sandwich central matrix
                        product or not) (no,yes); default no"\
            MODE=2(T); NVALUES=2(1); SET=yes;\
            VALUES=!T(ESTIMATE,OMIT),!T(NO,YES); \
            DEFAULT=!T(ESTIMATE),!T(NO);

PARAMETER NAME = \
          'CORRELATIONS',"I/O: matrix; the correlation matrix"\
          'ESTIMATES',   "I: variate; estimates of parameters in
                         model"\
          'Y',           "I: variate; response variate"\
          'RESIDUALS',   "I: variate; residuals"\
          'FITTEDVALUES',"I: variate; fitted values"\
          'TIME',        "I: variate; times of repeated measures"\
          'MARKER',      "I: factor; identifier of subject or outcome"\
          'DISTRIBUTION',"I: text; identifier of distribution"\
          'SCALEFACTOR', "I: text; scalefactor option in use"\
          'SFVALUE';     "I: scalar; value of scalefactor if FIXED"\
          SET=10(yes);DECLARED=10(yes); \
          TYPE='symmetric',5('variate'),'factor',2('text'),'scalar'; \
          PRESENT=9(yes),no

GETATTRIBUTE [ATTRIBUTE=NVALUES] ESTIMATES; SAVE=!P(ncol)
 &           [ATTRIBUTE=NROWS] CORRELATIONS; SAVE=!P(ntime)
```

```
DIAGONALMATRIX [ROWS=ntime;MODIFY=yes] done,wkdm; \
               VALUES=!(#ntime(1)),*

CALC const = 'ESTIMATE' .IN. CONSTANT
 &   sandw = 'NO' .IN. SANDWICH

IF sandw
"
  SCALEFACTOR is as in GEE i.e. FIXED means fixed to SFVALUE
  CONSTANT means the scalefactor is estimated but constant
  across time, and VARYTIME means the scalefactor is estimated
  and varies across time.

  The variate TIME in this PROCEDURE represents the 1...ntime
  distinct times, it is not a FACTOR of length nobs as in GEE.
  It is the levels of the parameter TIME of GEE.
"
  IF DISTRIBUTION.EQS.'NORMAL'
    IF SCALEFACTOR.NES.'VARYTIME'
      IF SCALEFACTOR.EQS.'FIXED'
        CALC wkdm = SFVALUE
      ELSE
        CALC wkdm = TRACE(CORRELATIONS)/ntime
      ENDIF
    ELSE
      CALC wkdm = CORRELATIONS
    ENDIF
    CALC CORRELATIONS = 0 + wkdm
  ELSE
    CALC CORRELATIONS = done
  ENDIF
ENDIF
ENDPROCEDURE
```

If `LINK`, `DISTRIBUTION` or `CRTYPE` are unset, but no user routines are given for `_GEEINIT`, `_GEELINK`, `_GEEDISTRIBUTION` and `_GEECORRELATION`, then those given here (for logit link, binomial error distribution and independence) will be used.

Action with `RESTRICT`

Input structures must not be restricted, and any existing restrictions will be cancelled.

References

Kenward, M.G., Lesaffre, E. & Molenberghs, G. (1994). An application of maximum likelihood and generalized estimating equations to the analysis of ordinal data from a longitudinal study with cases missing at random. *Biometrics*, **50**, 945-953.

Kenward, M.G. & Smith, D.M. (1995a). Computing the generalized estimating equations for repeated measurements. *Genstat Newsletter*, **32**, 50-62.

Kenward, M.G. & Smith, D.M. (1995b). Computing the generalized estimating equations for repeated ordinal, categorical measurements. *Genstat Newsletter*, **32**, 63-70.

Liang, K.-Y. & Zeger,S.L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, **73**, 13-22.

Liang, K.-Y., Zeger, S.L. & Qaqish, B. (1992). Multivariate regression analyses for categorical data (with discussion). *Journal of the Royal Statistical Society, Series B*, **54**, 3-40.

See also

Procedures: `GLMM`, `HGANALYSE`.

Genstat Reference Manual 1 Summary sections on: Regression analysis, Repeated measurements.

GENPROCRUSTES

Performs a generalized Procrustes analysis (G.M. Arnold & R.W. Payne).

Options

PRINT = *string tokens*	Printed output required (analysis, centroid, column, individual, monitoring); default anal, cent
SCALING = *string token*	Type of scaling to use (none, isotropic, separate); default none
METHOD = *string token*	Method to be used (Gower, TenBerge); default Gowe
NROOTS = *scalar*	Number of roots (i.e. dimensions) to print for the output configurations, consensus and rotation matrices, and number of dimensions to save with the XOUTPUT, CONSENSUS and ROTATIONS paramaters if their matrices have alread not been defined; default is to print and save all the dimensions
PLOT = *string tokens*	Controls which graphs to display (consensus, individuals, projections); default * i.e. none
NDROOTS = *scalar*	Number of dimensions to display in the consensus and individuals plots; default 3
TOLERANCE = *scalar*	The algorithm is assumed to have converged when (last residual sum of squares) − (current residual sum of squares) < TOLERANCE × (number of configurations); default 0.00001
MAXCYCLE = *scalar*	Limit on number of iterations; default 50

Parameters

XINPUT = *pointers*	Each pointer points to a set of matrices holding the original input configurations
XOUTPUT = *pointers*	Each pointer points to a set of matrices to store a set of final (output) configurations
CONSENSUS = *matrices*	Stores the final consensus configuration from each analysis
ROTATIONS = *pointers*	Each pointer points to a set of matrices to store the rotations required to transform each set of XINPUT configurations to their final (scaled) XOUTPUT configurations
RESIDUALS = *pointers*	Each pointer points to a set of matrices to store the distances of a set of scaled XINPUT configurations from its consensus
RSS = *scalars*	Stores the residual sum of squares from each analysis
ROOTS = *diagonal matrices*	Stores the latent roots from referring the centroid configuration to its principal axis form (consensus) for each analysis
WSS = *scalars*	Stores the initial within-configuration sum of squares from each analysis
SCALINGFACTOR = *variates*	Stores the isotropic scaling factors for configurations from each analysis
PROJECTIONS = *pointers*	Each pointer points to a set of matrices to store a set of projection matrices

Description

An $N \times V$ matrix represents a configuration of N points in V dimensions. Given a set of M such matrices (XINPUT), a generalized Procrustes analysis iteratively matches them to a common centroid configuration by the operations of translation to a common origin, rotation/reflection of axes and possibly also scale changes. This matching seeks to minimise the sum of the squared distances between the centroid and each individual configuration summed over all points (the Procrustes statistic for each configuration and the centroid, summed over all configurations). The final centroid is referred to principal axes to give a unique consensus configuration. Two methods of scaling are available (controlled by the SCALING option). Isotropic scaling, which scales the all the dimensions of each configuration by an equal amount, takes place during the Procrustes analysis. The alternative is to scale each configuration prior to the analysis so that the trace of each matrix is one (see Arnold 1992). If this latter method is used, the subsequent residuals represent pure lack-of-fit and the scaling factors given in the results represent differences in relative size/spread of the original (centred) configurations, whereas for overall isotropic scaling the scaling factor contains components of both size and lack-of-fit.

Procedure GENPROCRUSTES carries out a generalized Procrustes analysis and has parameters for saving various results for future use (XOUTPUT, CONSENSUS, ROTATIONS, RESIDUALS, RSS, ROOTS, WSS, SCALINGFACTOR, PROJECTIONS). There are options for different methods to use for the matching (SCALING, METHOD), control of convergence (TOLERANCE, MAXCYCLE) and printing and plotting of results (PRINT, PLOT, NROOTS and NDROOTS).

Note that the special case of M=2 corresponds to the classical pairwise Procrustes matching (ROTATE directive) except that by fitting each configuration to a common centroid the requirement to regard one of the initial configurations as fixed is obviated.

Options: PRINT, SCALING, METHOD, NROOTS, PLOT, NDROOTS, TOLERANCE, MAXCYCLE.
Parameters: XINPUT, XOUTPUT, CONSENSUS, ROTATIONS, RESIDUALS, RSS, ROOTS, WSS, SCALINGFACTOR, PROJECTIONS.

Method

The default method used for generalized Procrustes analysis in GENPROCRUSTES is that described by Gower (1975). Each input configuration (XINPUT – referred to henceforth as X_i, i=1...M) is initially column-centred, with the individual column means for each configuration optionally printed (by including the column setting with the PRINT option). If separate scaling is requested (option SCALING=separate), the matrices are also scaled to have trace one (see Arnold 1992). A constraint is required on the overall sum of squares to prevent the trivial solution of matching by all configurations collapsing to the origin. In this procedure the constraint used is

$$\sum (\text{trace} (X_i' X_i)) = M.$$

An initial estimate of the centroid is found from these centred and scaled configurations; firstly X_2 is rotated to X_1, with the rotated X_2 saved as the new X_2 and the centroid computed as the mean of X_1 and the new X_2; X_3 is rotated to this centroid which is then recalculated as the mean of the three current configurations; and so on until all configurations X_i (i=1...M) have been included. The centroid thus found is taken as the initial centroid estimate Y, with the rotated values as the new X_i. The initial residual sum of squares S_r is calculated as

$$Sr = M \times (1 - \text{trace} (Y' Y)).$$

Each of the current configurations X_i is then rotated to Y and the rotated position saved as the new X_i. The updated estimate of the centroid Y_n is calculated as the mean of the new X_i (i=1...M) and the new residual sum of squares calculated as

$$Sr_n = M \times (1 - \text{trace} (Y_n' Y_n)).$$

If isotropic scaling has been requested (option SCALING=isotropic) new estimates ro_i' of the individual scaling factors ro_i (originally set to 1) are now found by

$ro_i'/ro_i = \sqrt{}(\text{trace}(X_i' Y_n)/(\text{trace}(X_i' X_i) \times \text{trace}(Y_n' Y_n)))$

and each X_i is updated by a factor of ro_i'/ro_i. The centroid is then recalculated as the mean of the new X_i and the new residual sum of squares calculated in a similar manner to before. If the change in residual *Sr* is less than a preset tolerance (controlled by option `TOLERANCE`) the algorithm is taken to have converged. If not, the process is repeated until the tolerance is reached, up to a maximum number of iterations as set by the option `MAXCYCLE` (default 50) after which a message of non-convergence is printed and the procedure terminated. Monitoring information about convergence can be printed by including the `monitoring` setting with the `PRINT` option.

After convergence a unique consensus configuration is found by referring the final centroid to principal axes; the corresponding latent roots may be saved using the `ROOTS` parameter. Final results for the consensus and individual configurations (referred to the same principal axes) may be printed using the `centroid` and `individual` settings of the `PRINT` option, and/or saved using the parameters `XOUTPUT`, `CONSENSUS` and `ROTATIONS`. By default, results are presented and saved for the maximum available dimensionality but the option `NROOTS` allows a reduced number of dimensions to be set. Analysis of variation for the *M* configurations (including the individual scaling factors) and for the *N* points, along with the initial within and between configurations sums of squares (*WSS* and *BSS*), the final residual sum of squares (*RSS*) and number of steps in the iteration process may be printed using the `analysis` setting of the `PRINT` option. The initial within-configuration sum of squares, final residual sum of squares and individual isotropic scaling factors may also be saved using, respectively, the `WSS`, `RSS` and `SCALINGFAC` parameters. (Note that the final results are still scaled by the original factor from the initial overall constraint; to return to the original scale all sums of squares need adjustment by a factor of *WSS*/*M* and configurations by the square root of that factor).

Independently of the choice of dimensionality for printing and saving, the `NDROOTS` option controls the dimensionality of the graphical output requested using the `PLOT` option (default 3). The `consensus` setting plots the consensus solution in the chosen dimensionalty, and the `individual` setting gives the individual final configurations as well as the consensus. The `projection` setting displays the projections (calculated from the individual rotation matrices scaled by the singular values from the consensus solution in principal axis form) as vectors labelled by configuration number and colour-coded for order of column. This projection plot can be particularly helpful in comparing the use of terms/attributes (columns of the configurations) by individual assessors in sensory analysis, both in conventional and free-choice profiling; see Arnold & Collins (1993) for further details.

Modifications to the method described above are given in TenBerge (1975), and may be invoked by the `TenBerge` setting of the `METHOD` option. This may give considerable savings in the time to reach convergence (Arnold 1988).

References

Arnold, G.M. (1988). Comparisons of algorithms for generalized Procrustes analyses. *Genstat Newsletter*, **22**, 7-11.

Arnold, G.M. (1992). Scaling factors in generalized Procrustes analysis. *Computational Statistics, Volume 1, Proceedings of the 10th Symposium on Computational Statistics, COMPSTAT, Neuchatel, Switzerland, August 1992*, 61-66.

Arnold, G.M. & Collins, A.J. (1993). Interpretation of transformed axes in multivariate analysis. *Applied Statistics*, **42**, 381-400.

Gower, J.C. (1975). Generalized Procrustes analysis. *Psychometrika*, **40**, 33-51.

TenBerge, J.M.F. (1977). Orthogonal Procrustes rotation for two or more matrices. *Psychometrika*, **42**, 267-276.

See also
Directives: ROTATE. FACROTATE.
Procedures: PCOPROCRUSTES, SAGRAPES.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

GESTABILITY

Calculates stability coefficients for genotype-by-environment data (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (means, stability, sortedstability, quantiles); default stab, quan
METHOD = *string tokens*	Methods to use to calculate stability (superiority, static, wricke, ranks); default supe
BESTMETHOD = *string token*	How to define the best genotype (minimum, maximum); default maxi
PLOT = *string tokens*	What graphs to plot (stability); default * i.e. none
NBEST = *string tokens*	Number of best genotypes to print in tables of sorted stability coefficients; default * i.e. print all of them
DIRECTION = *string token*	Direction to sort tables of sorted stability coefficients (ascending, descending); default asce
PERCENTQUANTILES = *scalar* or *variate*	Percentage points for which quantiles are required; default !(50,5,1,0.1)
NTIMES = *scalar*	Number of permutations to make; default 999
BLOCKSTRUCTURE = *formula*	Model formula defining any blocking to consider during the permutation test; default none
EXCLUDE = *factors*	Factors in the block formula whose levels are not to be randomized in the permutation test

Parameters

Y = *variates*	Yields (or other measurements) made on the genotypes in the environments
GENOTYPES = *factors*	Genotype corresponding to each yield
ENVIRONMENTS = *factors*	Environment where each yield was recorded
SEED = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically
STABILITY = *tables* or *pointers*	Saves stability coefficients
QUANTILES = *tables* or *pointers*	Saves quantiles of the stability coefficients
TITLE = *texts*	Overall title for the graphs; default * i.e. none

Description

To assess new genotypes of plants, trials are often carried out in a range of environments. Yields and other measurements will then be made, and analyses carried out (e.g. using REML) to see how well the genotypes perform. These analyses allow you to determine which genotypes are best overall, or at a specific site. However, they do not consider how reliable, or stable, their yields may be overall. GESTABILITY allows you to calculate several stability coefficients to assess this. These are selected using the METHOD option.

The superiority setting of the METHOD option calculates the cultivar-superiority measure of Lin & Binns (1988). For each genotype, this is the sum of the squares of the differences between its mean in each environment and the mean of the best genotype there, divided by twice the number of environments. The BESTMETHOD option specifies whether the best genotype is defined to be the one with the maximum mean yield or the one with the minumum mean yield. (You would want to take the minimum as best, for example, if the "yields" were disease scores.) Genotypes with the smallest values of the superiority measure tend to have better yields and to be more stable.

The `ranks` setting gives the mean and variance of the ranks of each genotype across the environments where it occurs, as well as the rank-difference coefficient of Nassar & Huehn (1987). For each genotype, this is the sum of the absolute differences between its ranks in all the pairs of environments where it occurs. This assesses the consistency of the response of each genotype with respect to the other genotypes.

The `static` setting calculates the static stability coefficient. For each genotype, this is defined as the variance between its means in the various environments. This provides a measure of the consistency of the genotype (but without taking account of how good it is).

The `wricke` setting gives Wricke's (1962, 1964) ecovalence stability coefficient. This is the contribution of each genotype, to the genotype-by-environment sum of squares, in an unweighted analysis of the genotype-by-environment means. A low value indicates that the genotype responds in a consistent manner to changes in environment.

The yields (or other measurements) are specified, in a variate, using the `Y` parameter. The `GENOTYPES` parameter specifies a factor to indicate the genotype that supplied each yield, and the `ENVIRONMENTS` parameter specifies a factor to indicate the environment where it was grown. `GESTABILITY` prefers to be given all the data, not just the mean yield. It can then do some permutation tests to help you assess the coefficients.

In the permutation tests, `GESTABILITY` randomly permutes the original data within each environment, and calculates and stores the coefficients. The `NTIMES` option controls how many permutations are done; so its default of 999 gives 1000 sets of coefficients (the set from the original unpermuted data, plus the 999 permuted data sets). `GESTABILITY` constructs a variate `Group` to indicate genotypes that occurred in exactly the same sets of environments: you cannot make comparisons between genotypes that occurred at different sites as these will have been competing with different genotypes across their environments. `GESTABILITY` combines the permuted and original coefficients within each group, and calculates quantiles over the combined set of values. A coefficient can then be taken as significant at a particular level if its coefficient is greater than the corresponding quantile. The `PERCENTQUANTILES` option specifies a variate or scalar to define which quantiles are calculated; the default gives 50%, 5%, 1% and 0.1%. The `SEED` parameter defines the seed used to generate the random numbers used to generate the permutations for each `Y` variate. The default value of zero initializes the seed at random if this is the first time that the Genstat randomization routines have been used in the current job; otherwise it continues the existing sequence of random numbers.

If the data come from a designed experiment, you may need to use the `BLOCKSTRUCTURE` option to specify a block model to define how to do the randomization. The `EXCLUDE` option can then restrict the randomization so that one or more of the factors in the block model is not randomized. See the `RANDOMIZE` directive for further details.

The `PRINT` option controls the printed output. The `means` setting prints the overall means of the genotypes. The `stability` setting prints the stability coefficients. These are accompanied by the quantiles from the permutation tests if the `quantiles` setting is also specified. The `sortedstability` setting prints the stability coefficients in a sorted order, as specified by the the `DIRECTION` option. The default, ascending, order prints the most stable genotypes first. The `NBEST` option can be set to control the number of genotypes that are included; by default they are all printed.

The `PLOT` option can be set to `stability` to plot the stabilities against the mean responses. This provides a way of simultaneously assessing the general effectiveness and stability of the genotypes. You can supply a title for the plots using the `TITLE` parameter.

The `STABILITY` parameter allows you to save the coefficients selected by the `METHOD` option. If only the cultivar-superiority measure has been selected, these are saved in a table. Otherwise a pointer of tables is saved with elements labelled by their names: `'superiority'`, `'static'`, `'wricke'`, `'rankmeandifference'`, `'rankmean'` and `'rankvariance'`. Similarly the `QUANTILES` parameter can save the quantiles. If there is a single percentile, a table is saved for

each coefficient. If there are several, a pointer of tables is saved for each one.

Options: PRINT, METHOD, BESTMETHOD, PLOT, NBEST, DIRECTION, PERCENTQUANTILES, NTIMES, BLOCKSTRUCTURE, EXCLUDE.
Parameters: Y, GENOTYPES, ENVIRONMENTS, SEED, STABILITY, QUANTILES. TITLE.

Action with RESTRICT
GESTABILITY takes account of any restrictions on Y, GENOTYPES or ENVIRONMENTS.

References
Lin, C.S. & Binns, M.R. (1988). A superiority performance measure of cultivar performance for cultivar x location data. *Canadian Journal of Plant Science*, **68**, 193-198.
Nassar, R. & Huehn, M. (1987). Studies on estimation of phenotype stability: tests of significance for nonparametric measures of phenotype stability. *Biometrics*, **43**, 45-53.
Wricke, G. (1962). Uber eine methode zur erfassung der okologischen streubreite in feldversuchen. *Zeitschrift Fur Pflanzenzuchtung*, **47**, 92-96.
Wricke, G (1964) Zur berechnung der okovalenz bei sommerweizen und hafer. *Zeitschrift Fur Pflanzenzuchtung*, **52**, 127-138.

See also
Procedures: AMMI, GGEBIPLOT, RFINLAYWILKINSON, DBIPLOT.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

GETNAME

Forms the name of a structure according to its IPRINT attribute (A.R.G. McLachlan).

No options

Parameters

STRUCTURE = *identifiers*	Structures whose names are to be obtained
NAME = *texts*	Saves the names of the structures
IDENTIFIER = *texts*	Saves the identifiers of the structures
EXTRA = *texts*	Saves the extra texts of the structures
IPRINT = *texts*	Saves (or forms) IPRINT attributes

Description

Procedure GETNAME can be used to obtain the name from any Genstat data structure.

The structure from which the name is to be obtained must be specified using the STRUCTURE parameter, and can be any data structure. The NAME parameter saves the name, which will be a text containing the identifier and/or the extra text associated with a structure depending on the setting of its IPRINT attribute. If the structure is one that does not have an IPRINT attribute (e.g. an LRV), NAME returns the identifier. If the structure is unnamed, NAME returns a missing text.

The IDENTIFIER parameter can save the identifier of the structure, and the EXTRA parameter can save the extra text from the structure (if any). The IPRINT parameter can save the IPRINT attribute if the structure has one. For structures that do not have an IPRINT attribute, IPRINT is set to 'identifier'. For unnamed structures, IPRINT returns a missing text.

Options: none.
Parameters: STRUCTURE, NAME, IDENTIFIER, EXTRA, IPRINT.

Method

GETNAME uses GETATTRIBUTE to get the necessary attributes.

See also

Directives: GETATTRIBUTE, TXCONSTRUCT.
Genstat Reference Manual 1 Summary section on: Data structures.

GETRGB

Gets the RGB values and names of the initial default graphics colours of the Genstat pens (R.W. Payne).

No options

Parameters

COLOUR = *scalars* or *variates*	Colour numbers
RGB = *scalars* or *variates*	RGB values
NAME = *texts*	Names of nearest colours

Description

Genstat supports a standard set of colours for graphics whose names and corresponding RGB values are defined in the help information for the PEN directive. The 256 Genstat pens take their initial default colours from this set, and Genstat procedures such as AGRAPH use the sequence of initial default colours of pens 1, 2... to define default colours for their graphs.

GETRGB provides a convenient way of accessing this sequence of colours. The COLOUR parameter specifies a scalar or variate containing the pen number(s) whose initial default colours are required. The RGB parameter saves a scalar or variate containing the corresponding colours, expressed as RGB values (see PEN). The NAME parameter saves a text containing the name of the nearest colour.

Options: none.
Parameters: COLOUR, RGB, NAME.

Method

GETRGB uses the RESET option and SAVE parameter of the PEN directive to get the initial default colours.

Action with RESTRICT

Restrictions are ignored.

See also

Directive: PEN.
Functions: BLUE, GRAY, GREY, GREEN, RED, RGB.
Genstat Reference Manual 1 Summary section on: Graphics.

GETTEMPFOLDER

Gets gets the location of the folder used by Genstat for temporary files (R.W. Payne).

Option

PRINT = *string token*	Controls printed output (tempfolder); default temp

Parameter

TEMPFOLDER = *text*	Saves the name of the temporary folder

Description

The various parts of Genstat (server, client and graphics viewer etc) pass information using files in the Genstat temporary folder. This can take place, unseen, in the ordinary use of Genstat. However, it may sometimes be useful for developers to know where this is happening.

By default, GETTEMPFOLDER prints the name of the folder, but you can suppress this by setting option PRINT=*. The TEMPFOLDER parameter can save the name of the folder, in a text.

Option: PRINT.
Parameter: TEMPFOLDER.

See also

Genstat Reference Manual 1 Summary section on: Program control.

GGEBIPLOT

Plots displays to assess genotype + genotype-by-environment variation (A.I. Glaser).

Options

PRINT = *string tokens*	What to print (variation); default * i.e. nothing
DIMENSIONS = *scalars*	Which dimensions to display; default 1,2
PLOT = *string token*	Type of plot (scatter, ranking, compare, joint, centred); default scat
METHOD = *string token*	Whether the names in LEV1 (and LEV2) are from the ENVIRONMENTS or GENOTYPES factor (environments, genotypes); default envi
SCPLOT = *string token*	Features to add to a scatter plot (hull, sector, megaenvironment, vector, linear); default * i.e. none
SCALING = *string tokens*	What scaling to use (genotype, environment, symmetric); default envi
NORMALIZE = *string token*	Whether to scale the data using the within-environment standard deviation (yes, no); default no
CULL = *variate* or *text*	Specifies environments at which to examine the performance of the genotypes in order to decide which genotypes to cull
QUANTILE = *scalar*	Proportion at which to calculate quantile for CULL; default 0.5.
DIVISIONS = *scalar*	Number of parallel lines or concentric circles to use when ranking genotypes or environments; default 10
RANKINGLINES = *string token*	Whether the ranking lines drawn with PLOT settings ranking or joint are perpendicular to the biplot axis or projected onto the axis (perpendicular, projection); default perp
GENREVERSE = *string token*	Whether to reverse the order of the genotype scores (yes, no); default no
ENVREVERSE = *string token*	Whether to reverse the order of the environment scores (yes, no); default no
WINDOW = *scalar*	Which graphical window to use; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2

Parameters

DATA = *variates* or *tables*	Provides the data to be analysed
GENOTYPES = *factors*	Specifies the genotypes
ENVIRONMENTS = *factors*	Specifies the environments
LEV1 = *texts* or *scalars*	First environment (or genotype) to use with PLOT settings centred, compare, joint or ranking, or with scatter when SCPLOT=linear
LEV2 = *texts* or *scalars*	Second environment (or genotype) to use with PLOT settings centred, compare or joint
LABGENOTYPES = *texts*	Labels for genotypes
LABENVIRONMENTS = *texts*	Labels for environments
TITLE = *texts*	Titles for the plots; if this is unset, an appropriate title is formed auomatically
MEGAGROUPS = *variates* or *texts*	Specifies or saves the groupings to use for the plot produced by SCPLOT=megaenvironment

Description

GGEBIPLOT provides a range of plots that are useful for assessing the performance of genotypes in different environments. The observed phenotypic variation (*P*) of genotypes across environments is made up of environment variations (*E*), genotype variations (*G*) and genotype-by-environment interaction (*GE*): i.e.

$$P = E + G + GE,$$

Usually *E* is the dominant source of variation, while *G* and *GE* are relatively small. Thus, it is usual to remove the environmental main effect *E*, and focus only on *G* and *GE*.

The data for GGEBIPLOT is a table of data values, classified by genotype and environment factors, and specified by the DATA parameter. The genotype and environment factors are specified by the GENOTYPES and ENVIRONMENTS parameters. You can set DATA to the table itself. Alternately, you can set it to a variate containing the raw data, and GGEBIPLOT will form the table as a table of means.

GGEBIPLOT illustrates the genotype plus genotype-by-environment variation using scores from a principal components analysis, treating the table as a data matrix. The rows (or units) of the data matrix correspond to the genotypes, and the columns (or variates) correspond to the environments. The analysis works on the matrix of variances and covariances between environments. The environment means are automatically removed during the calculation of the variances and covariances. So the analysis automatically ensures that it is only the genotype variation and genotype-by-environment interaction that is examined. You can also scale the columns first, using the within-environment standard deviation, by setting option NORMALIZE=yes. Usually the scores are taken from the first two dimensions of the decomposition, but you can request others by setting the DIMENSIONS option. You can set option PRINT=variation to print the amount of variation explained by these two dimensions; by default, nothing is printed.

GGEBIPLOT plots the scores in a range of different ways, together with biplot axes from the principal components analysis. Essentially these are standard principal-component biplots, but various additional information can be added to the plots, as suggested in the book *GGE Biplot Analysis* by Yan & Kang (2003), to help elucidate the genotype and environment relationships.

The PLOT option controls the plots that are displayed. The setting scatter plots the genotype and environment scores. The SCPLOT option allows further information to be included on the plot, with settings:

hull	to draw an enclosing convex hull around the genotype scores;
sector	to draw lines from the origin perpendicular to each side of the convex hull around the genotype scores, to divide the biplot into sectors;
megaenvironment	to draw an ellipse round those environments which share the same sector;
vector	to draw lines connecting environment scores with the origin;
linear	to draw the same lines as vector, together with a rug plot at the side showing the angles between the environments, the parameter LEV1 must then be set to the label (or level) of an environmental factor which will be used as a "base" factor.

Note that hull, sector and megaenvironment can be used together, but vector and linear must be used individually. For single-trait data, genotypes at the vertex of the convex hull are considered to be the best performers in the environments that occur in the same sector (these are known as the vertex cultivars). The sector setting splits the plots into different sectors. The

genotypes in the same sector as a particular environment should be those with higher yields in that environment. As a general rule, the vertex cultivar will be the highest-yielding genotype in all environments with which it shares a sector. The `megaenvironment` setting draws an ellipse around those environments which share a sector (if the ellipse extends into another sector and sector lines are plotted, the ellipse lines become dashed when they go into a different sector).

The `MEGAGROUPS` parameter can be used to specify or save the groups used for the `megaenvironment` setting. To specify the groups, you can set `MEGAGROUPS` to a variate or text with the same length as the number of levels of the `ENVIRONMENTS` factor; its values indicate the group to which each environment belongs. Alternatively, if `MEGAGROUPS` is set to an undefined data structure, or one with no values, this will be defined as a variate containing the default group definitions.

The `PLOT` setting `ranking` can examine the performances of all the genotypes within a specific environment. Alternatively, you can set option `METHOD=genotype` to examine all the environments for a specific genotype. This draws a biplot axis through the specific environment (or genotype) together with ranking lines to show the best performing genotypes (or environments) in that environment (or genotype). By default the ranking lines are drawn to be perpendicular to the biplot axis, but you can set option `RANKINGLINES=projection` to project lines from the environments (or genotypes) to the biplot axis instead. In the plot, the best performing genotypes (or environments) are those whose projections onto the biplot axis are closest to the environment or genotype). The required genotype (or environment) is specified by setting the parameter `LEV1` to either the label or level of the required environment (or genotype). If `LEV1` is unset or is set to a missing value, an axis is drawn through the "average environment coordinate" (*AEC*), with the appropriate ranking lines. The *AEC* is represented by a circle on the plot.

The `PLOT` setting `compare` can compare the performance of the environments with a specific environment, or you can set option `METHOD=genotype` to compare the genotypes with a specific genotype. The specific environment (or genotype) is viewed as an "ideal" environment (or genotype), and concentric circles are plotted around it. The closer an environment (or genotype) is to the "ideal" environment (or genotype) the more attributes they share. The required environment (or genotype) is specified by setting the parameter `LEV1` to either the label or level of the required environment (or genotype). If `LEV1` is unset or is set to a missing value, `GGEBIPLOT` constructs an "ideal" environment (or genotype), and draws concentric circles from its point. The constructed "ideal" environment (or genotype) lies on the line that joins the origin to the *AEC*, at a distance from the origin equal to the distance from the origin to the environment (or genotype) with the greatest yield. (The "ideal" environment or genotype considers only those environments or genotypes that show greater than average yield.) The "ideal" environment (or genotype) is represented by an arrow on the plot. In practice the "ideal" is unlikely to exist, but can be used as a reference point. It is also possible to see where the *AEC* is in relation to the "ideal" genotype (or environment) by setting `LEV2` to a missing value.

The major difference between `ranking` and `compare` is that `ranking` shows the best performing environments (or genotypes) in a genotype (or environment) in a single dimension, whilst `compare` shows the best performing genotypes (or environments) in comparison to an "ideal" genotype (or environment) in two dimensions. The `DIVISIONS` option specifies the number of lines, or concentric circles, to use when ranking genotypes or environments with `PLOT` settings `ranking` or `compare`; the default is to use 10.

The `PLOT` setting `joint` can be used to compare two environments simultaneously, or you can set option `METHOD=genotype` to compare two genotypes. When comparing two environments, a line is drawn joining the environments. A median point on this line is found, which acts as a virtual trait. A biplot axis is plotted passing through this median and the origin. Ranking lines are also drawn to the biplot axis, as with the `PLOT` setting `ranking`; the `RANKINGLINES` option again controls whether these are perpendicular to the axis or projected onto the axis. The

genotypes that are furthest along the biplot axis (in the direction of the arrow) are considered to be the best performing genotypes in the two environments. Alternatively, when comparing two genotypes, a line is drawn joining the genotypes. An axis is now drawn through the origin perpendicular to this joining line. The environments on the same side of the axis as one of the chosen genotypes are those where that genotype is considered to have a better performance. In some circumstances both genotypes may end up on the same side of the axis. The genotype that is closest to the axis is then considered to have a better performance in the environments on the other side of the perpendicular line. The two environments (or genotypes) are specified by setting `LEV1` and `LEV2` to their levels or labels.

The `PLOT` setting `centred` can produce a scatter plot of the environment-centred data, with the x and y-axes representing two of the environments. In this case only the genotypes are plotted. Alternatively, you can set `METHOD=genotype` to produce a plot of the genotype-centred environment data, with the x and y-axes representing two of the genotypes. The line y=x is also plotted. Genotypes (or environments) below this line perform better in the environment (or genotype) representing the x-axis, and genotypes (or environments) above this line perform better in the environment (or genotype) representing the y-axis. The two environments (or genotypes) are again specified by setting `LEV1` and `LEV2` to their levels or labels.

When there are a large number of genotypes it may be helpful to cull some of them from the biplot. For example, you may want to remove genotypes that have performed badly in some of the environments. To do this you specify `CULL` to a variate or a text containing the levels or labels of the environments that you want to consider. Then, by default, all genotypes with y-values less then the median value at each chosen environment will be removed. Alternatively, you can specify some other quantile at which to cull by using the `QUANTILE` option. Note, however, if you select more than one environment when the y-values at the environments are negatively correlated, there may be very few (or possibly no) genotypes left to plot.

The `GENREVERSE` and `ENVREVERSE` options can reverse the y-direction in the plots of the genotype and environment scores, respectively,

By default, the species scores, site scores and x-variable(s) are labelled by the labels of the `ENVIRONMENTS` and `GENOTYPES` factors, if available, or otherwise by their levels. Alternatively, you can specify other labels using the `LABENVIRONMENTS` and `LABGENOTYPES` parameters.

Options: `PRINT`, `DIMENSIONS`, `PLOT`, `METHOD`, `SCPLOT`, `SCALING`, `NORMALIZE`, `CULL`, `QUANTILE`, `DIVISIONS`, `RANKINGLINES`, `GENREVERSE`, `ENVREVERSE`, `WINDOW`, `KEYWINDOW`.
Parameters: `DATA`, `GENOTYPES`, `ENVIRONMENTS`, `LEV1`, `LEV2`, `LABGENOTYPES`, `LABENVIRONMENTS`, `TITLE`, `MEGAGROUPS`.

Method

`GGEBIPLOT` calculates a principal components analysis on the data variates, which automatically column-centres the data thus removing the environmental effects. The eigenvectors for genotype *i* and/or the eigenvectors for environment *j* are multiplied by a constant to get environment and genotype scores. The constant is chosen by setting the `SCALING` option as follows:

`genotype`	$\lambda_i \times i$th environmental eigenvector
`environment`	$\lambda_i \times i$th genotype eigenvector
`symmetric`	genotype scores scaled by $\sqrt{\lambda_i} \times i$th environmental eigenvector, environment scores scaled by $\sqrt{\lambda_i} \times i$th genotype eigenvector

where $\{\lambda_i\}$ are the singular values of the data, with the values of *i* set by `DIMENSIONS`.

The singular values are equivalent to multiplying the roots from a principal components analysis by $(n-1)$ and then raising to the power of $-½$. The eigenvectors for the genotypes are obtained by multiplying the scores from a principal components analysis by a diagonal matrix containing the singular values. The enviromental eigenvectors are calculated by multiplying the

data by the inverse of (the genotype eigenvectors multiplied by the singular values).

The genotype-focused scaling is used to display the interrelationships of the genotypes. The environment-focused scaling is probably used most frequently. It displays the interrelationship among environments, and has the following properties.

(1) The cosine of the angle between any two environments approximates their correlation.
(2) The lengths of the environment vectors are approximately proportional to their standard deviations.
(3) The inner product between two environments approximates their covariance.

The symmetric scaling method allows for comparisons of the relative variances between the genotypes and environments.

References

Yan, W. & Kang, M.S. (2003). *GGE Biplot Analysis: a Graphical Tool for Breeders, Geneticists and Agronomists*. CRC Press, Boca Raton.

Hunt, L.A. & Yan, W. (2002). Biplot analysis of diallel data. *Crop Science*, **42**, 21-30.

See also

Procedures: `AMMI`, `GESTABILITY`, `RFINLAYWILKINSON`, `DBIPLOT`, `CABIPLOT`, `CRBIPLOT`, `CRTRIPLOT`.

Genstat Reference Manual 1 Summary sections on: REML analysis of linear mixed models, Graphics.

GHAT

Calculates an estimate of the G nearest-neighbour distribution function (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

Y = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
X = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
S = *variates*	Vectors of distances to use with each pattern; no default – this parameter must be set
GVALUES = *variates*	Variates to receive the estimated G nearest-neighbour distribution functions
NNDISTANCES = *variates*	Variates to receive the nearest-neighbour distances
NNUNITS = *variates*	Variates to receive the unit numbers of the nearest neighbours

Description

The G nearest-neighbour distribution function relates to the distribution of distances from each event of a spatial point pattern to the nearest other event in the pattern (see Diggle 1983). An estimate of G can be obtained by calculating the empirical distribution function (EDF) GHAT(s) which is defined as the proportion of events for which the nearest other event is within distance *s*.

The term complete spatial randomness (CSR) is used to represent the hypothesis that the overall density of events in a spatial point pattern is constant throughout the study region, and that the events are distributed independently and uniformly. Under CSR, the G nearest-neighbour distribution function is given by

$$G(s) = 1 - \exp(-\pi \times density \times (s^2)),$$

where *density* is the overall density of events per unit area. (The procedure FZERO can be used to calculate values of this function for a pattern with a given density.) The G nearest-neighbour distribution function for a clustered (regular) pattern will tend to be larger (smaller) than the corresponding function for a completely random pattern, at least for small distances.

The procedure GHAT requires the coordinates of a spatial point pattern (specified by the parameters X and Y) and a vector of distances at which to calculate the EDF of G (specified by the parameter S). The primary output of the procedure is a vector of estimates of G corresponding to the distances in S. The estimated G function can be saved using the parameter GVALUES. The nearest-neighbour distances and the unit numbers of the nearest-neighbours can be saved using the parameters NNDISTANCES and NNUNITS.

Printed output is controlled using the PRINT option. The default setting of summary prints the distances at which the G function is estimated and the estimates themselves under the headings S and GVALUES.

Option: PRINT.
Parameters: Y, X, S, GVALUES, NNDISTANCES, NNUNITS.

Method

A procedure PTCHECKXY is called to check that X and Y have identical restrictions. GHAT then calls a procedure PTPASS to call a Fortran program to calculate the G nearest-neighbour

distances. No corrections are made for edge effects. The EDF of the nearest-neighbour distances relative to the distances specified by the parameter S is obtained using the CALCULATE directive.

Action with RESTRICT
If X and Y are restricted, only the subset of values specified by the restriction will be included in the calculations. The parameter S may also be restricted.

Reference
Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

See also
Procedures: FHAT, KHAT, KSTHAT, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

GINVERSE

Calculates the generalized inverse of a matrix (S.K. Haywood).

Options

PRINT = *string token*	Printed output from the procedure (inverse); default *, i.e. no printing
METHOD = *string token*	Method to be used to invert symmetric matrices (svd, lrv); default lrv
TOLERANCE = *scalar*	How close a number must be be to zero before it is recognised as zero; default 1.0^{-6}

Parameters

INMATRIX = *matrices*	The matrix whose inverse is to be calculated
INVERSE = *matrices*	Matrix to save the generalized inverse

Description

GINVERSE calculates generalized inverses. The method of inversion depends on the type of the matrix to be inverted. Ordinary (square or rectangular) matrices are inverted using the singular value decomposition. This method can also be used for symmetric matrices, by setting option METHOD=svd; however, by default, these are inverted using an eigenvalue (lrv) decomposition. For a diagonal matrix, the inverse is calculated by taking the reciprocal of each individual value on the diagonal. The tolerance for zero, to be used in the calculations, can be set using the TOLERANCE option.

The inverse can be saved using the INVERSE parameter; its type and dimensions will be defined automatically if it has not been declared in advance. The inverse can also be printed, by setting option PRINT=inverse.

Options: PRINT, METHOD, TOLERANCE. Parameters: INMATRIX, INVERSE.

Method

GINVERSE inverts a matrix structure by performing an singular value decomposition to represent the input matrix as

```
left-matrix *+ diagonal-matrix *+ right-matrix.
```

In the case of a square matrix both left- and right-matrices are orthogonal while, for a non-square matrix, one of the left- or the right-matrices is orthogonal (which one depending on the dimensions of the input matrix), and the other one is orthonormal. The three matrices are inverted: the diagonal matrix by taking each non-zero element and calculating the inverse of that value, the left- and right-matrices by transposition. The order of the three matrices is then reversed, and they are multiplied together to form the generalized inverse of the original matrix.

A symmetric matrix is inverted in a similar way except that, by default, the matrix is decomposed using an eigenvalue decomposition.

A diagonal matrix is inverted by taking the reciprocal of each non-zero element down the diagonal.

For further details, see Graybill (1969) pages 96-103.

Reference

Graybill, A.F. (1969). *Introduction to Matrices with Applications in Statistics*. Colorado State University, Fort Collins, Colorado.

See also
Directive: CALCULATE.
Function: GINVERSE .

†GLDISPLAY

Displays further output from a GLMM analysis (R.W. Payne).

Options

PRINT = *string token*	What output to display (model, components, effects, fittedvalues, means, backmeans, vcovariance, waldtests, missingvalues, covariancemodels, deviance); default *
PTERMS = *formula*	Formula specifying fixed terms for which means or back-transformed means are to be printed; default * prints all the fixed model terms
PSE = *string token*	Standard errors to print with tables of means (se, sesummary, sed, sedsummary, vcovariance, differences, estimates, alldifferences, allestimates); default seds
OFFSET = *scalar*	Offset value to use when calculating predicted means; default 0
RMETHOD = *string token*	Which random terms to use when calculating RESIDUALS (final, all); default fina
CFORMAT = *string token*	Whether printed output for covariance models gives the variance matrices or the parameters (variancematrices, parameters); default vari
FMETHOD = *string token*	Controls whether and how to calculate F-statistics for fixed terms (automatic, none, algebraic, numerical); default auto
GLSAVE = *pointer*	Save structure from the GLMM analysis

No parameters

Description

GLDISPLAY allows you to display further output from a GLMM analysis. By default the output is from the most recent GLMM analysis. Alternatively, you can set the GLSAVE parameter to a save structure (saved using the GLSAVE parameter of GLMM) to obtain output from an earlier analysis.

The PRINT option selects the output to be displayed:

model	description of model fitted,
components	estimates of variance components and estimated parameters of covariance models,
effects	estimates of parameters α and β, the fixed and random effects,
fittedvalues	table containing the y-variate, fitted values, residuals on the natural scale and standardized residuals on the scale of the linear predictor,
means	predicted means for factor combinations,
backmeans	back-transformed means,
vcovariance	variance-covariance matrix of the estimated components,
waldtests	Wald tests for fixed terms,
missingvalue	estimates of missing values,
covariancemodels	estimated covariance models, and
deviance	deviance from the generalized linear model.

The default is PRINT= mode, comp, effe, mean, back, moni, vcov, cova.

The deviance represents the variation remaining after fitting the fixed terms and all the random

terms. It thus assesses how well those terms explain the random variation in the data.

The RMETHOD option controls the way in which residuals and fitted values are formed. With the default setting RMETHOD=final, the fitted values are calculated from all the fixed and random effects. The setting RMETHOD=all can be used to obtain fitted values constructed from the fixed terms alone, omitting all random terms. (The residuals are then calculated as the differences between the values of the y-variate and the fitted values.) To avoid problems with 0 and 100% observations, the standardized residuals on the linear-predictor scale are calculated as differences between the adjusted dependent variate and the fitted values on that scale (and then standardized by their standard errors).

The PTERMS option can specify which tables of means are printed; by default, tables of means are produced for all the terms in the fixed model.

The PSE option controls the standard errors that are printed with tables of means and effects:

se	standard errors,
sesummary	summary of the standard errors (default),
sed	standard errors of differences between pairs of means,
sedsummary	summary of the standard errors of differences,
vcovariance	variance-covariance matrix for the means,
allestimates	synonym of se,
estimates	synonym of sesummary,
alldifferences	synonym of sed,
differences	synonym of sedsummary.

The OFFSET option specifies the offset value to use when calculating predicted means. The default is zero.

The CFORMAT option controls the type of output produced for the estimated covariance models. The default setting, variancematrices, produces the variance-covariance matrices for the components, whereas the setting parameters prints their parameters.

The FMETHOD option controls whether to accompany the Wald tests for fixed effects with approximate F statistics and corresponding numbers of residual degrees of freedom. The computations, using the method devised by Kenward & Roger (1997), can be time consuming with large or complicated models. So, with the default setting FMETHOD=automatic, Genstat assesses the model itself and decides automatically whether to do the computations and which method to use. The other settings allow you to control what to do yourself:

none	no F statistics are produced;
algebraic	F statistics are calculated using algebraic derivatives (which may involve large matrix calculations);
numerical	F statistics are calculated using numerical derivatives (which require an extra evaluation of the mixed model equations for every variance parameter).

Options: PRINT, PTERMS, PSE, OFFSET, RMETHOD, CFORMAT, FMETHOD, GLSAVE.
Parameters: none.

See also

Procedures: GLMM, GLKEEP, GLPERMTEST, GLPLOT, GLPREDICT, GLRTEST.
Genstat Reference Manual 1 Summary section on: Regression analysis.

†GLKEEP

Saves results from a GLMM analysis (R.W. Payne).

Options

FACTORIAL = *scalar*	Limit on number of factors in the model terms generated from the TERMS parameter; default 3
RESIDUALS = *variate*	Residuals from the analysis
FITTEDVALUES = *variate*	Fitted values from the analysis
DISPERSION = *scalar*	Dispersion component
VCOVARIANCE = *symmetric matrix*	Variance-covariance matrix for the estimates of the variance components
VESTIMATES = *variate*	Saves a vector of all parameters in the variance model
VARESTIMATES = *symmetric matrix*	Variance-covariance matrix for the parameters in the variance model (as saved by VESTIMATES)
VLABELS = *text*	Vector of text labels for the VESTIMATES and VARESTIMATES structures
MVESTIMATES = *variate*	Estimates of missing values
MVSE = *variate*	Standard errors of missing-value estimates
MVUNITS = *variate*	Unit numbers of missing values
DEVIANCE = *scalar*	Saves the deviance
MODEL = *pointer*	Information defining the mode;
RMETHOD = *string token*	Which random terms to use when calculating RESIDUALS (final, all); default fina
DFFIXED = *scalar*	Number of degrees of freedom in the fixed model
DFRANDOM = *scalar*	Number of degrees of freedom in the random model
FMETHOD = *string token*	Controls how to calculate F-statistics for fixed terms (automatic, none, algebraic, numerical); default auto
WMETHOD = *string token*	Controls which Wald statistics are saved (add, drop); default drop
OFFSET = *scalar*	Offset value to use when calculating predicted means; default 0
ITERATIVEWEIGHTS = *variate*	Saves the iterative weights from the generalized linear model fitting
LINEARPREDICTOR = *variate*	Linear predictor from a generalized linear model
YADJUSTED = *variate*	Adjusted response variate
ZADJUSTED = *variate*	Adjusted dependent variate on the linear predictor scale
LPRESIDUALS = *variate*	Residuals from the fit on the linear predictor scale
SELPRESIDUALS = *variate*	Standard errors for the residuals from the fit on the linear predictor scale
EXIT = *scalar*	Exit status of the fit (0 if successful)
GLSAVE = *pointer*	Save structure from the GLMM analysis

Parameters

TERMS = *formula*	Model terms for which information is required
COMPONENTS = *scalar* or *pointer to scalars*	Estimated variance components
MEANS = *table* or *pointer* to *tables*	Predicted means for each term
BACKMEANS = *table* or *pointer* to *tables*	Back-transformed means

SEDMEANS = *symmetric matrix* or *pointer* to *symmetric matrices*
Standard errors of differences between means
VARMEANS = *symmetric matrix* or *pointer* to *symmetric matrices*
Variance-covariance matrix for the means
EFFECTS = *table* or *pointer* to *tables*
Effects for each term
SEDEFFECTS = *symmetric matrix* or *pointer* to *symmetric matrices*
Standard errors of differences between effects
VAREFFECTS = *symmetric matrix* or *pointer* to *symmetric matrices*
Variance-covariance matrix for the effects
CADJUSTMENT = *scalar* or *pointer to scalars*
For a term involving covariates, saves the adjustment made to its values during the analysis
WALD = *scalar* or *pointer to scalars* Wald statistic (fixed terms only)
FSTATISTIC = *scalar* or *pointer to scalars*
F statistics (fixed terms only)
NDF = *scalar* or *pointer to scalars* Numerator d.f. (fixed terms only)
DDF = *scalar* or *pointer to scalars* Denominator d.f. (fixed terms only)

Description

GLKEEP saves results from a GLMM analysis. By default the results are from the most recent GLMM analysis. Alternatively, you can set the GLSAVE parameter to a save structure (saved using the GLSAVE parameter of GLMM) to save results from an earlier analysis.

The RESIDUALS and FITTEDVALUES options can specify variates to save the residuals and fitted values, respectively. The RMETHOD option controls the way in which residuals and fitted values are formed. With the default setting RMETHOD=final, the fitted values are calculated from all the fixed and random effects. The setting RMETHOD=all can be used to obtain fitted values constructed from the fixed terms alone, omitting all random terms. (The residuals are then calculated as the differences between the values of the y-variate and the fitted values.)

The DISPERSION option saves the dispersion coefficient, in a scalar.

The variance-covariance matrix for the estimates of the variance component can be saved using the VCOVARIANCE option. (The estimates themselves are saved using the COMPONENTS parameter, as described below.)

The VESTIMATES option saves a variate containing all the variance parameters estimated in the model. The VARESTIMATES option can supply a symmetric matrix to save the variance-covariance matrix for the estimates of the variance parameters, matching the ordering and contents of VESTIMATES. The vector of labels for these parameters can be saved by the VLABELS option.

The MVESTIMATES option saves a variate containing estimates of the missing values, the MVSE option saves their standard errors, and the MVUNITS option saves a list of the units that are missing.

The DEVIANCE option saves the deviance from the generalized linear model. This represents the variation remaining after fitting the fixed terms and all the random terms. It thus assesses how well those terms explain the random variation in the data.

The degrees of freedom fitted by the fixed model can be saved by the DFFIXED option, and the degrees of freedom in the random model can be saved by the DFRANDOM option.

The MODEL option can be used to save a pointer, with labels 'distribution', 'link', 'aggregation', 'klogratio', 'owndist', 'ownlink', 'random', 'fixed', 'constant', 'factorial', 'offset', 'cdefinitions', 'cvectors', 'y', and 'nbinomial', storing the settings of the corresponding options and parameters of GLMM. The labels can be specified in either lower or upper case, or any mixture.

The ITERATIVEWEIGHTS parameter saves the iterative weights used in the last cycle of the iteration, and the LINEARPREDICTOR parameter saves the linear predictor. The YADJUSTED parameter saves the adjusted response variate used in the last cycle of the iteration, and the ZADJUSTED parameter similarly saves the adjusted response variate on the scale of the linear predictor. The LPRESIDUALS option saves the residuals from the fit on the linear predictor scale. To avoid problems with 0 and 100% observations, they are calculated as differences between the adjusted dependent variate and the fitted values on that scale. The SELPRESIDUALS option saves their standard errors. The EXIT option saves a scalar indicating the exit status for the fit of the GLMM (0 if successful, 1 otherwise).

The parameters of GLKEEP save information about particular model terms in the analysis. With the TERMS parameter you specify a model formula, which Genstat expands to form the series of model terms about which you wish to save information. The FACTORIAL option sets a limit on the number of factors in each term. Any term containing more than that limit is deleted. The subsequent parameters allow you to specify identifiers of data structures to store various components of information for each of the terms that you have specified.

The MEANS parameter saves tables of predicted means, and the BACKMEANS parameter saves back-transformed means. The OFFSET option specifies the offset value to use when calculating predicted means; the default is zero. The SEDMEANS parameter saves symmetric matrices of standard errors of differences for the means, and the VARMEANS parameter saves symmetric matrices of their variances and covariances. The EFFECTS parameter saves tables of effects, and the SEDEFFECTS and VAREFFECTS parameter saves symmetric matrices with standard errors for their differences and their variances and covariances, respectively.

If a term involves a covariate, the CADJUSTMENT parameter can save the adjustment that will have been made to its values during the analysis. This will be zero if option CADJUST was set to none in GLMM. Alternatively, if CADJUST had its default setting of mean, each covariate will have been centred by subtracting its (weighted) mean.

The Wald statistic for fixed terms can be saved in scalars using the WALD parameter. The WMETHOD option controls whether these are from the table where terms are added sequentially to the model, or that where terms are dropped from the full fixed model. The associated F statistic, and its numerator and denominator numbers of degrees of freedom, can be saved in scalars by the FSTATISTIC, NDF and DDF parameters, respectively. The FMETHOD option specifies which algorithm to use to calculate the denominator numbers of degrees of freedom. The default, automatic, will use any stored values that have been calculated for this analysis by earlier GLMM, GLDISPLAY or GLKEEP statements; otherwise it will choose automatically between the two available methods. (See REML for more details.)

If you have a single term, you can supply a table, symmetric matrix or scalar for each of these parameters, as appropriate. However, if you have several terms, you must supply a pointer which will then be set up to contain as many tables, symmetric matrices or scalars as there are terms.

Options: FACTORIAL, RESIDUALS, FITTEDVALUES, DISPERSION, VCOVARIANCE, VESTIMATES, VARESTIMATES, VLABELS, MVESTIMATES, MVSE, MVUNITS, DEVIANCE, MODEL, RMETHOD, DFFIXED, DFRANDOM, FMETHOD, WMETHOD, OFFSET, ITERATIVEWEIGHTS, LINEARPREDICTOR, YADJUSTED, ZADJUSTED, LPRESIDUALS, SELPRESIDUALS, EXIT, GLSAVE.

Parameters: TERMS, COMPONENTS, MEANS, BACKMEANS, SEDMEANS, VARMEANS, EFFECTS, SEDEFFECTS, VAREFFECTS, CADJUSTMENT, WALD, FSTATISTIC, NDF, DDF.

See also

Procedures: GLMM, GLDISPLAY, GLPERMTEST, GLPLOT, GLPREDICT, GLRTEST.

Genstat Reference Manual 1 Summary section on: Regression analysis.

GLM

Analyses non-standard generalized linear models (P.W. Lane).

Options

PRINT = *string tokens*	What to display (deviance, estimates, correlations, monitoring); default devi, esti
DISTRIBUTION = *string token*	Distribution of response (Normal, Poisson, binomial, gamma, inversenormal); default * indicates calculations supplied for non-standard distribution via procedure GLMDISTRIBUTION (see the details of the procedures called by GLM)
LINK = *string token*	Link function (identity, logarithm, logit, reciprocal, power, squareroot, probit, complementaryloglog); default * indicates calculations supplied for non-standard link via procedure GLMLINK (see Method)
EXPONENT = *scalar*	Exponent for power link; default −2
TERMS = *list* or *formula*	Explanatory variates, factors, and interactions specified as for the standard regression directives; default null model
CONSTANT = *string token*	Whether to include constant term (estimate, omit); default esti
INITIALLINEAR = *variate*	Initial guess at linear predictor, if specifying own link function and not defining procedure GLMINITIAL

Parameters

Y = *variates*	Response variate; this parameter must be set
NBINOMIAL = *variates*	Totals for use when DISTRIBUTION=binomial; must then be set
FITTEDVALUES = *variates*	To store correct fitted values

Description

A range of standard generalized linear models can be fitted using the regression directives MODEL, FIT and so on. Procedure GLM allows non-standard models to be fitted: you can choose to define your own link function, or the distribution of the response variable, or both. The standard links and distributions can be chosen by setting the options DISTRIBUTION, LINK and EXPONENT as in the MODEL directive; non-standard ones require the definition of auxiliary procedures to carry out the necessary calculations: see the details of the procedures called by GLM. The terms in the fitted model are specified by the TERMS option, which may be set to a list of terms or to a formula, as in the TERMS directive, or may be left unset to fit a null model. The CONSTANT option may be set to estimate or omit a constant term. The Y parameter must be set to specify the response variate, and for a binomial distribution the NBINOMIAL parameter must be set, as in the MODEL directive.

The output from the procedure is controlled by the PRINT option: by default, the residual deviance with d.f. and the parameter estimates with s.e.s are given. Standard errors are based on the residual mean square for all distributions: there is no SCALE option like in the MODEL directive. After using the procedure, the regression directives RDISPLAY and RKEEP may be used to display or save results, as for standard models fitted with the FIT directive. However, some of the output will not be appropriate: the total deviance from the summary setting will be incorrect, but the residual deviance should be correct; also, the response variate, fitted values and residuals will be incorrect in the output for the fittedvalues setting and if RKEEP is used to

save results. The correct fitted values may be saved with parameter FITTEDVALUES of GLM.

Options: PRINT, DISTRIBUTION, LINK, EXPONENT, TERMS, CONSTANT, INITIALLINEAR.
Parameters: Y, NBINOMIAL, FITTEDVALUES.

Method

The model is fitted by iteratively-reweighted least-squares, as outlined by Nelder & Wedderburn (1972).

If a non-standard distribution is required, the option DISTRIBUTION should be left unset and the GLMDISTRIBUTION procedure defined, before using the GLM procedure. The NBINOMIAL parameter must be included in the definition, even if the NBINOMIAL parameter of GLM is not used.

```
GLMDISTRIBUTION Y=variate; FITTEDVALUES=variate; \
  VARIANCE=variate; DEVIANCE=scalar; NBINOMIAL=variate
```

Forms the variance function and the deviance using the fitted values and the response variate.

If a non-standard link function is required, the option LINK should be left unset and the procedure GLMLINK defined, before using the GLM procedure, to specify the necessary calculations for the link function. The NBINOMIAL parameter must be included in the definition, even if the NBINOMIAL parameter of GLM is not used. In addition, either the GLMINITIAL procedure must be defined, to specify calculations to form an initial guess at the linear predictor, or the INITIALLINEAR option must be set to a variate that holds this initial guess.

```
GLMLINK LINEARPREDICTOR=variate; FITTEDVALUES=variate; \
  DERIVATIVE=variate; NBINOMIAL=variate
```

Forms the fitted values and the derivative of the link function – the derivative of the linear predictor with respect to the fitted value – using the linear predictor.

```
GLMINITIAL Y=variate; LINEARPREDICTOR=variate; NBINOMIAL=variate
```

Forms initial values for the linear predictor using the response variate, adjusting if necessary to avoid values unsuitable for the link function.

Action with RESTRICT

Any restriction on a variate in the Y parameter list is applied to all calculations. No vector in the TERMS list or formula should be restricted, unless with the same restriction as for the Y variate.

Reference

Nelder, J.A. & Wedderburn, R.W.M. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A*, **135**, 370-384.

See also

Directive: MODEL.
Genstat Reference Manual 1 Summary section on: Regression analysis.

GLMM

Fits a generalized linear mixed model (S.J. Welham).

Options

†PRINT = *string token*	What output to display (model, components, effects, fittedvalues, means, backmeans, monitoring, vcovariance, waldtests, missingvalues, covariancemodels, deviance); default mode, comp, effe, mean, back, moni, vcov, cova
DISTRIBUTION = *string token*	Error distribution (binomial, poisson, normal, gamma, negativebinomial); default bino
LINK = *string token*	Link function (identity, logarithm, logit, reciprocal, probit, complementaryloglog, logratio); default * gives the canonical link
DISPERSION = *scalar*	Value at which to fix the residual variance, if missing the variance is estimated; default 1 for binomial, Poisson and negative binomial distributions, a missing value otherwise
RANDOM = *formula*	Random model *excluding* bottom stratum; this must be set
FIXED = *formula*	Fixed model; default *
ABSORB = *factor*	Absorbing factor to be used at the REML step of the iterations
CONSTANT = *string token*	Whether to estimate or omit constant term in fixed model (omit, estimate); default esti
FACTORIAL = *scalar*	Limit on number of factors/covariates in a model term; default 3
PTERMS = *formula*	Formula specifying fixed terms for which means or back-transformed means are to be printed; default * prints all the fixed model terms
†PSE = *string token*	Standard errors to print with tables of means (se, sesummary, sed, sedsummary, vcovariance, differences, estimates, alldifferences, allestimates); default seds
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or variates or y-variates
MAXCYCLE = *scalar*	Maximum number of iterations of the GLMM algorithm; default 20
TOLERANCE = *scalar*	Convergence criterion for iterative procedure; default 0.0001
†FMETHODGLMM = *string token*	Specifies fitting method (all, fixed): all indicates the method of Schall (1991); fixed indicates the marginal method of Breslow & Clayton (1993) ; default all
OFFSET = *variate*	Variate holding values to be used as an offset on the linear predictor scale; default *
CADJUST = *string token*	What adjustment to make to covariates for the REML analysis (mean, none); default mean

AGGREGATION = *scalar*	Fixed parameter for negative binomial distribution (parameter *k* as in variance function var = mean + mean^2/*k*); default 1
KLOGRATIO = *scalar*	Parameter *k* for logratio link, in form log(mean / (mean + *k*)); default as set in AGGREGATION option
OWNDIST = *text*	For non-standard distributions only: text specifying the variance function to be used with dummy variable DUM, e.g. OWNDIST='DUM'
OWNLINK = *text*	For non-standard link functions only: text specifying 3 functions using dummy variable DUM – the link function, its inverse and its derivative, e.g. OWNLINK = !T('log(DUM)','exp(DUM)','1/DUM')
CDEFINITIONS = *text*	Statements to execute to define correlation models; default * i.e. none
CVECTORS = *pointer*	Data structures involved in the correlation models
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 1
VCONSTRAINTS = *string token*	Whether to constrain variance components to be positive (none, positive); default posi
†VMETHOD = *string token*	Indicates whether to use the standard Fisher-scoring algorithm or the new AI algorithm with sparse matrix methods (Fisher, AI); default AI
†VMAXCYCLE = *scalar*	Limit on the number of iterations; default 30

Parameters

Y = *variates*	Dependent variates
NBINOMIAL = *scalars* or *variates*	Number of binomial trials for each unit (must be set if DISTRIBUTION=binomial)
FITTEDVALUES = *variates*	Variates to save fitted values
COMPONENTS = *variates*	Variate to save estimated variance components
VCOVARIANCE = *symmetric matrices*	Variance-covariance matrix for the variance components
MEANS = *pointers*	Pointer to save tables of means for each Y variate
VARMEANS = *pointers*	Pointer to save covariance matrices of tables of means for each Y variate
BACKMEANS = *pointers*	Pointer to save tables of back-transformed means for each Y variate
ITERATIVEWEIGHTS = *variates*	Saves the iterative weights from the generalized linear model fitting
INITIALFITTEDVALUES = *variates*	Defines initial values for the fitted values; if unset, these are formed automatically
†EXIT = *scalar*	Exit status for the fit of the GLMM (0 if successful)
SAVE = *REML save structures*	Saves details of the REML analysis used to fit the model
†GLSAVE = *pointer*	Saves details of the GLMM analysis

Description

Procedure GLMM estimates the parameters of a generalized linear mixed model using either the method of Schall (1991) or the marginal method of Breslow & Clayton (1993), as described in the *Methods* Section.

The procedure assumes a generalized linear mixed model, that is a generalized linear model

with both fixed and Normally-distributed random effects on the scale of the linear predictor. The procedure estimates the fixed effects together with the variance components associated with the random effects.

The DISTRIBUTION option sets the error distribution; the default is to assume a binomial distribution but the Poisson, gamma and negative-binomial distributions are also available. Other distributions can be used via the OWNDIST option; this should be set to a text containing the formula for calculating the variance function for the required distribution, in terms of dummy variable DUM. The link can be set using the LINK option; the default takes the canonical link. Identity, logarithm, logit, reciprocal, probit, complementaryloglog or logratio link functions are also provided, and alternative link functions can be used via the OWNLINK option. In this case, OWNLINK must be set to a text with three values containing formulae (in terms of dummy variable DUM) for calculating the link function, its inverse and its first derivative. For example, instead of specifying a Poisson distribution with log link, the OWNDIST and OWNLINK options could be set as

```
OWNDIST='DUM'; OWNLINK=!T(LOG(DUM),EXP(DUM),'1/DUM')
```

Where necessary, these expressions should be constructed so that invalid results (eg. divide by zero or log(zero)) are avoided.

The AGGREGATION option supplies the aggregation parameter for the negative-binomial distribution; default 1. The KLOGRATIO option supplies the parameter *k* to be used in the logratio link, and takes its default from AGGREGATION.

The DISPERSION option specifies the dispersion parameter. The default is 1 for binomial, Poisson and negative binomial distributions, a missing value otherwise (indicating that the dispersion parameter is to be estimated).

The fixed and random models are specified by the FIXED and RANDOM options. The number of factors in the terms of the fixed model can be limited using the FACTORIAL option. By default the variance components are constrained to be positive, but you can set option VCONSTRAINTS to none to allow them to become negative.

The VMETHOD option specifies the algorithm to use in the REML steps of the GLMM algorithm: either Fisher or AI(default). The ABSORB option can specify an absorbing factor for use with the Fisher algorithm. However, if the absorbing factor appears in any of the terms of the FIXED model, no estimates of error will be available for these terms (see the *Guide to the Genstat Command Language*, Part 2, Sections 5.3.3 and 5.3.7). The VMAXCYCLE option controls the number of iterations used by the REML algorithm.

By default, a constant term is included in the model; this can be suppressed by setting option CONSTANT=omit. An offset can be included in the linear predictor by setting option OFFSET. By default any covariates are centred for the REML fitting by subtracting their means, weighted according to the iterative weights of the generalized linear model. You can save the iterative weights using the ITERATIVEWEIGHTS parameter, or you can set option CADJUST=none to request that the uncentred covariates are used instead.

It is also possible to define correlation models on the random terms, although the results should be used with caution as their properties are not yet well understood. To do this, you should set the CDEFINITIONS option to a text containing the Genstat statements required to define the models (e.g. using VSTRUCTURE). You also need to set the CVECTORS option to a pointer containing the data structures involved in the statements. Then, in the statements themselves, you should refer to each of these as CVECTORS[n], where n is the position of the relevant data structure in the pointer. For example:

```
TEXT cdef; VALUE=\
'VSTRUCTURE [CVECTORS[1].CVECTORS[2]] ar,ar; FACTOR=CVECTORS[1,2];
ORDER=1'
GLMM  [DISTRIBUTION=gamma; LINK=log; FIXED=variety;\
      RANDOM=fieldrow*fieldcolumn; CDEFINITION=cdef;\
      CVECTORS=!p(fieldrow,fieldcolumn)] yield
```

The MVINCLUDE option allows the inclusion of units with missing values, as in the REML directive. By default, units where there is a missing value in the y-variate or in any of the factors or variates in the model terms are excluded. The setting explanatory allows units with missing values in factors or variates in the model to be included. For missing covariate values, this is equivalent to substituting the mean value. The setting yvariate includes units with missing values in the y-variate. This can be useful to retain the balanced structure of the data for use with direct product covariance matrices (see VSTRUCTURE), or to produce predictions of data values for given values of explanatory factors and/or variates.

The FMETHODGLMM option specifies the method used to form the fitted values and therefore determines the fitting method to be used. The default setting all specifies that both fixed and random terms should be used to form fitted values which gives the method of Schall (1991); setting fixed indicates that only fixed terms are used to form fitted values which gives the marginal method of Breslow & Clayton (1993).

The PRINT option selects the output to be displayed:

model	description of model fitted,
components	estimates of variance components and estimated parameters of covariance models,
effects	estimates of parameters α and β, the fixed and random effects,
fittedvalues	table containing the y-variate, fitted values, residuals on the natural scale and standardized residuals on the scale of the linear predictor,
means	predicted means for factor combinations,
backmeans	back-transformed means,
monitoring	monitoring information at each iteration,
vcovariance	variance-covariance matrix of the estimated components,
waldtests	Wald tests for fixed terms,
missingvalue	estimates of missing values,
covariancemodels	estimated covariance models, and
deviance	deviance from the generalized linear model.

The default is PRINT=mode, comp, effe, mean, back, moni, vcov, cova.

The deviance represents the variation remaining after fitting the fixed terms and all the random terms. It thus assesses how well those terms explain the random variation in the data.

To avoid problems with 0 and 100% observations, the standardized residuals on the linear-predictor scale are calculated as differences between the adjusted dependent variate and the fitted values on that scale (and then standardized by their standard errors). The fitted values include the random as well as the fixed terms. The GLDISPLAY procedure can print residuals and fitted values where the fitted values are calculated only from the fixed terms.

The PTERMS option can specify which tables of means are printed; by default, tables of means are produced for all the terms in the fixed model.

The PSE option controls the standard errors that are printed with tables of means and effects:

se	standard errors,
sesummary	summary of the standard errors (default),
sed	standard errors of differences between pairs of means,
sedsummary	summary of the standard errors of differences,
vcovariance	variance-covariance matrix for the means,
allestimates	synonym of se,
estimates	synonym of sesummary,
alldifferences	synonym of sed,
differences	synonym of sedsummary.

Some control over the iterative GLMM algorithm is provided by option MAXCYCLE which sets

the maximum number of iterations (default 20), and by option TOLERANCE which specifies the criterion for determining convergence of the algorithm (default 0.0001). Convergence is judged to have been attained once the maximum change in the ratio (variance component)/(residual variance) and the change in the residual variance are less than the specified TOLERANCE.

The dependent variate is specified using the Y parameter. The NBINOMIAL parameter must be set when DISTRIBUTION=binomial to specify the total number of trials on each unit, as a variate if the number varies from unit to unit or as a scalar if it is constant over all the units.

The other parameters are used to save results. The variance components and residual variance can be saved in a variate using parameter VCOMPONENTS, with their variance-covariance matrix stored in a symmetric matrix specified by parameter VCOVARIANCE. The tables of means to be saved are determined by the setting of PTERMS. The tables are stored in a pointer specified by parameter MEANS, in the order in which they appear in the FIXED model. Their variance matrices and tables of back-transformed means are stored similarly in pointers specified by parameters VARMEANS and BACKMEANS. The EXIT parameter saves a scalar indicating the exit status for the fit of the GLMM (0 if successful, 1 otherwise).

You can display further output from the analysis using the GLDISPLAY procedure, and use the GLKEEP procedure to save information in Genstat data structures. The GLPREDICT procedure can form predictions. You can use the GLRTEST procedure to assess the random model, and the GLPERMTEST procedure to do permutation tests to assess the fixed model. By default these procedures take the most recent GLMM analysis, but you can use the GLSAVE to save the results of the analysis, to use instead in future calls of these procedures.

Alternatively, VDISPLAY and VKEEP can be used to redisplay or store other results from the internal REML estimation, provided REML has not been used in the interim. You can use the SAVE parameter to save the REML save structure, and use that as input to these directives, if REML may be used for another analysis.

Options: PRINT, DISTRIBUTION, LINK, DISPERSION, RANDOM, FIXED, ABSORB, CONSTANT, FACTORIAL, PTERMS, PSE, MVINCLUDE, MAXCYCLE, TOLERANCE, FMETHODGLMM, OFFSET, CADJUST, AGGREGATION, KLOGRATIO, OWNDIST, OWNLINK, CDEFINITIONS, CVECTORS, WORKSPACE, VCONSTRAINTS, VMETHOD, VMAXCYCLE.

Parameters: Y, NBINOMIAL, FITTEDVALUES, COMPONENTS, VCOVARIANCE, MEANS, VARMEANS, BACKMEANS, ITERATIVEWEIGHTS, INITIALFITTEDVALUES, EXIT, SAVE, GMSAVE.

Method

GLMM estimates the parameters of the Generalized Linear Mixed Model using either the method of Schall (1991) or the marginal method of Breslow & Clayton (1993). The method used is determined by the setting of option FMETHODGLMM.

The data y arises from some specified distribution with variance function sV and expected value μ. The link function g (with inverse h) is such that

$$g(\mu) = \eta = X\,a + Z\,b$$

where X is the design matrix for the vector a of fixed effects and Z is the design matrix for the vector b of random effects. The random effects b can be attributed to c random factors which are assumed to have zero mean and to be uncorrelated with each other and with e:

$$\mathrm{Cov}(b) = D = \mathrm{Diag}\{\ \sigma_1^2\, I_1 \ldots \sigma_c^2\, I_c\ \}$$

The method used by Schall (1991) develops the algorithm by analogy with the algorithm for estimating conventional generalized linear models. The link function applied to the data is linearized about μ to give the adjusted dependent variate z,

$$z = X\,a + Z\,b + e\ g'(\mu)$$

where $e=y-\mu$ and $g' = dg/d\mu$.

Then

$$\mathrm{E}(z) = X\,a;\ \ \mathrm{Cov}(b) = D;$$

$$\mathrm{Cov}(e\, g'(\mu)) = sV(\mu) \times (d\eta/d\mu) \times (d\eta/d\mu) = s \times W(\mu)^{-1}$$

where s is the dispersion parameter. Hence

$$\mathrm{Cov}(z) = s \times W(\mu)^{-1} + Z\,D\,Z'.$$

This has the same form as the general linear mixed model, and the fixed effects and variance components can be estimated by REML with (iterative) weights W.

This leads to the following algorithm:

Step 1) Using initial estimates of the variance components and of μ, calculate the adjusted variate z and weights W.

Step 2) Get new estimates of the variance components and of μ by REML on adjusted variate z with weights W.

Step 3) Convergence in estimates ⇒ exit algorithm.

Step 4) Use new estimates to update adjusted variate z and weights vector W.

Step 5) Go to Step 2.

The marginal model used by Breslow and Clayton is derived from a first order approximation (linearisation about Xa) to give

$$y \sim h(Xa) + h'(Xa)Zb + e$$

where ~ indicates approximation, h is the inverse of the link fuction g and e is $y-\mu$. They then work in terms of the marginal mean, M=h(Xa). Quasi-likelihood estimation leads to an algorithm similar to the one above, but the working variate becomes

$$z = Xa + (y-M)g'(M) = Xa + Eg'(M)$$

where $E=y-M$. The working variate z then has variance

$$\mathrm{Cov}(z) = s \times W(M)^{-1} + Z\,D\,Z'.$$

The same algorithm is used to fit the model, replacing μ by M and e by E.

The only difference between the two algorithms is then in the method used to form the mean μ or M and the "error" variate e or E. The option RMETHOD of REML controls the method of forming fitted values after REML estimation (i.e. including just fixed terms, or all terms except the residual) and this option is used inside the procedure to determine which of the models is fitted.

Initial values for the variance components are calculated by REML estimation using the fixed and random models on the data transformed by the link function. Initial values for the fixed effects are calculated by fitting the fixed model only to a generalized linear model with the specified link and error distribution. The WORKSPACE option specifies the number of blocks of internal memory to be set up for use by the REML algorithm; see the REML directive for more details.

Action with RESTRICT

If the Y-variate is restricted, only the units not excluded by the restriction will be analysed.

References

Breslow, N.E. & Clayton, D.G. (1993). Approximate inference in generalized linear mixed models. *Journal of the American Statistical Association*, **88**, 421, 9-25.

McCullagh, P. & Nelder, J.A. (1989). *Generalized Linear Models (second edition)*. Chapman & Hall, London.

Schall, R. (1991) Estimation in generalized linear models with random effects. *Biometrika*, **78**, 719-727.

See also

Procedures: GEE, HGANALYSE, GLDISPLAY, GLKEEP, GLPERMTEST, GLPLOT, GLPREDICT, GLRTEST.

Genstat Reference Manual 1 Summary section on: Regression analysis.

†GLPERMTEST

Does random permutation tests for generalized linear mixed models (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (prwald, criticalwald, ownstatistics, monitoring); default prwa, crit
NTIMES = *scalar*	Number of permutations to make; default 99
NRETRIES = *scalar*	Maximum number of extra samples to take when some analyses fail to converge; default NTIMES
BLOCKSTRUCTURE = *formula*	Model formula defining any blocking to consider during the randomization; default none
EXCLUDE = *factors*	Factors in the block formula whose levels are not to be randomized
SEED = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically
BINMETHOD = *string token*	How to permute binomial data (individuals, units; default indi
WMETHOD = *string token*	Controls which Wald statistics are used (add, drop); default add
OWNMETHOD = *string token*	Type of test required for own statistics (twosided, greaterthan, lessthan); default twos
CIPROBABILITY = *scalar*	Probability level for the confidence interval for own statistics; default 0.95

Parameters

GLSAVE = *pointers*	Save structure of the original analysis from GLMM; default * uses the save structure from the most recent GLMM analysis
WALD = *pointers*	Saves a pointer with a variate for each of the fixed terms containing the Wald statistics from the permuted data sets
PRWALD = *pointers*	Saves a pointer with a scalar for each of the fixed terms, containing the test probability obtained from the position of its Wald statistic within those from the permuted data sets
CRITICALWALD = *pointers*	Saves a pointer with variates for the 5%, 1% and 0.1% significance levels containing the corresponding critical values for the fixed terms, obtained from the quantiles of the Wald statistics from the permuted data sets
NNOTCONVERGED = *scalars*	Saves the number of permuted data sets whose analyses failed to converge
OWNDATA = *pointers*	Data required to calculate own statistics
OWNOBSERVEDVALUES = *variates*	Saves observed values of the own statistics
OWNPROBABILITIES = *variates*	Saves bootstrap probabilities for the own statistics
OWNESTIMATES = *variates*	Saves bootstrap estimates for the own statistics
OWNSES = *variates*	Saves bootstrap standard errors for the own statistics
OWNLOWERCIS = *variates*	Saves bootstrap lower values of the confidence intervals for the own statistics
OWNUPPERCIS = *variates*	Saves bootstrap upper values of the confidence intervals for the own statistics

OWNSTATISTICS = *pointers*	Saves the own statistics obtained from the permuted data sets, in a pointer with a variate for each statistic

Description

GLPERMTEST performs random permutation tests for fixed terms in a generalized linear mixed model, analysed by GLMM. A problem with these analyses is that their estimates of the variance components are generally biased i.e. the estimates are smaller than the true values. The Wald tests also suffer from bias, in that their test probabilities may be too small. You therefore need to be cautious when the probabilities from the tests are close to their critical values, especially when analysing small data sets or data from a binary distribution.

GLPERMTEST uses random permutation tests to provide an alternative way of assessing the fixed terms. It forms random permutations of the response, analyses those data sets, and records their Wald statistics. The distributions of the Wald statistics, under the null hypothesis of no fixed effects, can be estimated by the sets of statistics obtained from the analyses of the permuted data sets. Test probabilities for the original Wald statistics can therefore be estimated by their locations within those sets.

Before using GLPERMTEST, you need to analyse the original data set by GLMM. The GLSAVE parameter supplies the save structure from that analysis. If this is not specified, GLPERMTEST uses the save structure from the most recent GLMM analysis. The save structure provides the settings of all the options and parameters that GLMM used in that analysis. The analyses of the permuted data sets can therefore be done in exactly the same way as the original analysis.

The NTIMES option defines how many random permutations to perform; by default there are 99. The NRETRIES option specifies the maximum number of extra samples to take when some analyses fail to converge; the default is to use the same number as specified by NTIMES. The NNOTCONVERGED parameter can save a scalar containing the number of permuted data sets whose analyses failed to converge. The results may be unreliable if more than a few analyses fail.

The SEED option allows you to specify the seed to use for the random-number generator that is used for the randomizations to form the permutations. The default, SEED=0, continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, it initializes the seed automatically. If NTIMES exceeds the maximum possible number of permutations for the data, an "exact" test is performed in which every permutation is used once. This is feasible only for small data sets. There are $n!$ (n factorial) permutations of n units: 3!=6, 4!=24, 5!=120, 6!=720, 7!=5040, 8!=40320, and so on.

If the data are from a designed experiment, you may need to use the BLOCKSTRUCTURE option to specify a block model to define how to do the randomization. The EXCLUDE option can then restrict the randomization so that one or more of the factors in the block model is not randomized. See the RANDOMIZE directive for further details.

The BINMETHOD option controls how the permutations are done for binomial data. The original data set will have contained a set of units, each recording a number of "successes" obtained from an observed number of individuals. The default, and recommended, method is to expand the data set to contain individuals themselves, and permute these. Alternatively, you can set BINMETHOD=units if you prefer to permute the units as a whole instead.

The WALD parameter can save a pointer with a variate for each of the fixed terms containing the Wald statistics from the analyses of the permuted data sets. Similarly the PRWALD parameter can save a pointer with a scalar for each of the fixed terms, containing the test probability obtained from the position of its Wald statistic within those from the permuted data sets.

You can define your own statistics to be assessed by the test. They are calculated by a procedure _GLPERMownstatistics, which is called by GLPERMTEST following the GLMM analysis of each permuted data set. Its use is shown in the GLPERMTEST example, which can be modified to calculate your own statistics instead. The information required by

_GLPERMownstatistics to do the calculations is supplied, in a pointer, by the OWNDATA parameter. The OWNMETHOD option specifies the type of test to be made. The default, twosided tests whether the statistics differ from zero. The greaterthan setting tests whether they are greater than zero, and the lessthan setting tests whether they are less than zero. Permutation estimates, standard errors and confidence intervals are also calculated, The CIPROBABILITY option specifies the probability for the confidence intervals (default 0.95). The OWNOBSERVEDVALUES parameter can save a variate containing the values of the own statistics from the original data set. The OWNPROBABILITIES can save a variate containing the probabilities from the tests. The OWNESTIMATES can save a variate containing the bootstrap estimates of the statistics (calculated as the mean of the values obtained from the bootstrap samples) The OWNSES can save a variate containing standard errors of bootstrap estimates. The OWNLOWERCIS and OWNUPPERCIS parameters can save variates containing the lower and upper values, respectively, of the confidence intervals. Finally, the OWNSTATISTICS can save the values of the own statistics obtained from the permuted data sets, in a pointer with a variate for each statistic.

Output is controlled by the PRINT option, with settings:

prwald	to print probabilities for the fixed terms, estimated from the locations of their Wald statistics within the sets obtained from the permuted data sets;
criticalwald	to print critical values for the Wald statistics, estimated by quantiles within the sets from the permuted data sets;
ownstatistics	to print estimates, standard errors and confidence intervals for the own statistics, and
monitoring	to monitor the progress of the analyses.

The default is to print probabilities and critical values.

Options: PRINT, NTIMES, NRETRIES, BLOCKSTRUCTURE, EXCLUDE, SEED, BINMETHOD, WMETHOD, OWNMETHOD, CIPROBABILITY.

Parameters: GLSAVE, WALD, PRWALD, CRITICALWALD, NNOTCONVERGED, OWNDATA, OWNOBSERVEDVALUES, OWNPROBABILITIES, OWNESTIMATES, OWNSES, OWNLOWERCIS, OWNUPPERCIS, OWNSTATISTICS.

Method

GLPERMTEST uses RANDOMIZE to perform the permutations, taking account of any block structure of the date. The model is fitted, for each data set using GLMM, and GLKEEP is used to save the Wald statistics. The QUANTILES function is used to calculate the critical values.

Action with RESTRICT

GLPERMTEST takes account of any restrictions on any of the y-variates or x-variates or factors in the model.

See also

Procedures: GLDISPLAY, GLKEEP, GLMM, GLPLOT, GLPREDICT, GLRTEST, APERMTEST, RPERMTEST.

Genstat Reference Manual 1 Summary section on: Regression analysis.

†GLPLOT

Plots residuals from a GLMM analysis (R.W. Payne).

Options

RMETHOD = *string token*	Which random terms to use when calculating the residuals (final, all); default all
BACKTRANSFORM = *string token*	Whether to plot residuals on the natural scale (calculated using back-transformed fitted values) or standardized residuals on the linear-predictor scale (link, none); default none
INDEX = *variate* or *factor*	X-variable for an index plot; default !(1,2...)
OFFSET = *scalar*	Value of offset to use when calculating the residuals; default 0
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	Overall title for the plots; the default is to form a title displaying the identifier of the y-variate and the type of residual
GLSAVE = *pointer*	Save structure from the GLMM analysis; default * uses the GLSAVE structure from the most recent GLMM analysis

Parameters

METHOD = *string tokens*	Type of residual plot (fittedvalues, normal, halfnormal, histogram, absresidual, index); default fitt, norm, half, hist
PEN = *scalars*, *variates* or *factors*	Pen(s) to use for each plot

Description

GLPLOT provides up to four types of residual plots from a GLMM analysis. These are selected using the METHOD parameter, with settings: fitted for residuals versus fitted values, normal for a Normal plot, halfnormal for a half-Normal plot, histogram for a histogram of residuals, absresidual for a plot of the absolute values of the residuals versus the fitted values, and index for a plot against an "index" variable (specified by the INDEX option). The PEN parameter can specify the graphics pen or pens to use for each plot.

The residuals and fitted values are accessed automatically from the analysis specified by the GLSAVE option. If the GLSAVE option has not been set, they are taken from the most recent GLMM analysis.

The RMETHOD option controls which random terms are used to calculate the residuals:

all	all the random effects (default), and
final	only the final random term.

Note that residuals based on the final random term will not be calculated when any of the variance components are negative, as the associated negative correlations can generate very misleading patterns. GLPLOT will then generate a warning that all the residuals are missing. You should then use RMETHOD=all instead.

The BACKTRANSFORM option specifies the scale of the residuals. The default is to plot standardized residuals on the linear-predictor scale. To avoid problems with 0 and 100% observations, these are formed as the difference between the adjusted dependent variate and the fitted values on the linear predictor scale (and then standardized). Alternatively, you can set BACKTRANSFORM=link to plot (unstandardized) residuals on the natural scale.

The OFFSET option specifies the offset value to use when calculating the residuals. The default is zero.

By default, high-resolution graphics are used. Line-printer graphics can be used instead, by setting option GRAPHICS=lineprinter.

The TITLE option can supply an overall title. If this is not set, a default title is formed displaying the identifier of the y-variate and the type of residual.

Options: RMETHOD, BACKTRANSFORM, INDEX, OFFSET, GRAPHICS, TITLE, GLSAVE.
Parameters: METHOD, PEN.

Method
Residuals and fitted values effects are accessed, using GLKEEP. The plots are produced using the DRESIDUALS procedure.

Action with RESTRICT
If the y-variate in the GLMM analysis was restricted, only units included by the restriction will be used in the graphs.

See also
Procedures: GLMM, GLDISPLAY, GLKEEP, GLPERMTEST, GLPREDICT, GLRTEST.
Genstat Reference Manual 1 Summary section on: Regression analysis.

†GLPREDICT

Forms predictions from a GLMM analysis (R.W. Payne).

Options

PRINT = *string tokens*	What to print (description, predictions, backpredictions, se, sesummary, sed, sedsummary, vcovariance); default desc, pred, back, seds
MODEL = *formula*	Indicates which model terms (fixed and/or random) are to be used in forming the predictions; default * includes all the fixed terms and relevant random terms
OMITTERMS = *formula*	Specifies terms to be excluded from the MODEL; default * i.e. none
FACTORIAL = *scalar*	Limit on the number of factors or variates in each term in the models specified by MODEL or OMITTERMS; default 3
PRESENTCOMBINATIONS = *factors*	Lists factors for which averages should be taken across combinations that are present
WEIGHTS = *tables*	One-way tables of weights classified by factors in the model; default *
OFFSET = *scalar*	Value of offset on which to base predictions; default 0
NBINOMIAL = *scalar*	Supplies the total number of trials to be used for prediction with a binomial distribution (providing a value *n* greater than one allows predictions to be made of the number of "successes" out of *n*, whereas the value one predicts the proportion of successes); default 1
PREDICTIONS = *table* or *scalar*	To save the predictions; default *
BACKPREDICTIONS = *table* or *scalar*	To save back-transformed predictions; default *
SE = *table* or *scalar*	To save standard errors of predictions; default *
SED = *symmetric matrix*	To save standard errors of differences between predictions; default *
VCOVARIANCE = *symmetric matrix*	To save variances and covariances of predictions; default *
GLSAVE = *pointer*	Save structure from the GLMM analysis; default * uses the SAVE structure from the most recent GLMM analysis

Parameters

CLASSIFY = *vectors*	Variates and/or factors to classify table of predictions
LEVELS = *variates*, *scalars* or *texts*	To specify values of variates and/or levels of factors for which predictions are calculated
PARALLEL = *identifiers*	For each vector in the CLASSIFY list, allows you to specify another vector in the CLASSIFY list with which the values of this vector should change in parallel (you then obtain just one dimension in the table of predictions for these vectors)
NEWFACTOR = *identifiers*	Identifiers for new factors that are defined when LEVELS are specified

Description

GLPREDICT can be used after the GLMM directive to produce predictions of the values of the response variate at particular values of the variables in the fixed or random models. By default the predictions are from the most recent GLMM analysis, but you can use another analysis by supplying its save structure using the GLSAVE option.

The parameters are the same as those of VPREDICT (which GLPREDICT uses to form the predictions). The CLASSIFY parameter specifies variates or factors that are to be included in the table of predictions, and the LEVELS parameter supplies the values at which the predictions are to be made. For a factor, you can select some or all of the levels, while for a variate you can specify any set of values. A single level or value is represented by a scalar; several levels or values must be combined into a variate (which may of course be unnamed). Alternatively, if the factor has labels, you can use these to select the levels for prediction by setting LEVELS to a text. A missing value in the LEVELS parameter is taken to stand for all the levels of a factor, or the mean value of a variate.

The PARALLEL parameter allows you to indicate that a factor or variate should change in parallel with another factor or variate. Both of these should have the same number of values specified for it by the LEVELS parameter of GLPREDICT. The predictions are then formed for each set of corresponding values rather than for every combination of these values.

When you specify LEVELS, a new factor must be defined to classify that dimension of the table. By default this will be an unnamed factor, but you can use the NEWFACTOR parameter to give it an identifier. The EXTRA attribute of the factor is set to the name of the corresponding factor or variate in the CLASSIFY list; this will then be used to label that dimension of the table of predictions.

The prediction calculations consist of two steps. The first step is to calculate a table of fitted values. The MODEL, OMITTERMS and FACTORIAL options specify the model to use for this. The formula specified by MODEL is expanded into a list of model terms, deleting any that contain more variates of factors than the limit specified by the FACTORIAL option. Then, any terms in the formula specified by OMITTERMS are removed.

The second step averages the fitted values over the classifications that are not in the list that was supplied by the CLASSIFY parameter. The WEIGHTS option can supply one-way tables classified by any of the factors in the model. These are used to calculate the weight to be used for each fitted value when calculating the averages. Equal weights are assumed for any factor for which no table of weights has been supplied. (Note, this differs from the default in PREDICT, which uses *marginal weights*; see the PREDICT option ADJUSTMENT for details.) In the averaging all the fitted values are generally used. However, if you define a list of factors using the PRESENTCOMBINATIONS option, any combination of levels of these factors that does not occur in the data will be omitted from the averaging. Where a prediction is found to be inestimable, i.e. not invariant to the model parameterization, a missing value is given.

The OFFSET option specifies the offset value to use when calculating predicted means. The default is zero.

The NBINOMIAL parameter can be used to supply the total number of trials to be used for back-transformed predictions with a binomial distribution. If you provide a value *n* greater than one, GLPREDICT predicts the number of "successes" out of *n*. The default, NBINOMIAL=1, predicts the proportion of successes.

Printed output is controlled by settings of the PRINT option with settings:

description	describes the terms and standardization policies used when forming the predictions,
predictions	prints the predictions,
backpredictions	prints back-transformed predictions,
se	prints standard errors of the predictions,
sesummary	prints the minimum, average and maximum standard error,

`sed`	prints standard errors of differences between the predictions,
`sedsummary`	prints the minimum, average and maximum standard error of difference,
`vcovariance`	prints the variance and covariances of the predictions.

The default is to print descriptions, predictions, back-transformed predictions, and a summary of the standard error of differences. Standard errors and standard errors of differences are printed only if the predictions themselves are printed.

You can also save the results, using the `PREDICTIONS`, `BACKPREDICTIONS`, `SE`, `SED` and `VCOVARIANCE` options.

Options: `PRINT`, `MODEL`, `OMITTERMS`, `FACTORIAL`, `PRESENTCOMBINATIONS`, `WEIGHTS`, `OFFSET`, `NBINOMIAL`, `PREDICTIONS`, `BACKPREDICTIONS`, `SE`, `SED`, `VCOVARIANCE`, `GLSAVE`.
Parameters: `CLASSIFY`, `LEVELS`, `PARALLEL`, `NEWFACTOR`.

See also
Procedures: `GLMM`, `GLDISPLAY`, `GLKEEP`, `GLPERMTEST`, `GLPLOT`, `GLRTEST`.
Genstat Reference Manual 1 Summary section on: Regression analysis.

†GLRTEST

Calculates likelihood tests to assess the random terms in a generalized linear mixed model (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (tests); default test
SELECTION = *string tokens*	Specifies information to print with the tests (aic, sic, bic, critical); default crit
CRITICAL = *variate*	Saves the critical values
GLSAVE = *pointer*	Save structure of the original analysis from GLMM; default * uses the save structure from the most recent GLMM analysis

Parameters

TERMS = *formula*	Random terms to be tested; default is to test them all
TESTSTATISTIC = *scalar* or *pointer* to *scalars*	Test statistics for each term
DF = *scalar* or *pointer* to *scalars*	Degrees of freedom of the test statistics
AIC = *scalar* or *pointer* to *scalars*	Akaike information coefficients for each term
SIC = *scalar* or *pointer* to *scalars*	Schwarz (Bayesian) information coefficients for each term

Description

GLRTEST can be used after a GLMM analysis to assess the effect of dropping random terms from the model. It uses the REML deviances to do this. In the GLMM algorithm, REML is used to analyse the adjusted dependent variate *z*, with the variate of iterative weights, defined by the generalized linear model. These depend on the current fitted values, and change at each iteration until convergence. (See the *Method* section of the GLMM procedure for more details.) The REML deviance is taken from the analysis of the final adjusted z-variate with the final iterative weights.

GLRTEST saves the deviance from the original analysis using VKEEP, and the final adjusted z-variate and variate of iterative weights using GLKEEP. It then does REML analyses with these variates, omitting each random term, saving their deviances, and calculating their differences from the original deviance. Akaike and Schwarz (Bayesian) information coefficients are obtained using the VAIC procedure.

Note that, for compatibility, it is important to use the same adjusted z-variate and the same iterative weights as in the original analysis. With the alternative, of doing GLMM analyses removing each random term, we would be taking deviances from REML analyses with their own adjusted z-variates and weights, which could be very different from those in the original analysis. So we would be comparing REML analyses with different models, different response variates and different weights, which would not provide a valid comparison. Of course this does mean that the results pertain to the REML analysis rather than to the GLMM analysis itself. So they should be used as guidance rather than as a definitive test. Often, however, the random terms will have been defined by the design of the investigation. The tests will then be used more as an indication of the effectiveness of the design than to decide whether to omit terms from the analysis.

By default, GLRTEST produces tests for every random term. However, you can use the TERMS parameter to request tests for a specific set of terms.

The default is to print the tests, but you can set option PRINT=* to suppress this. The additional information to be printed with the tests is controlled by the SELECTION option, with settings:

aic	Akaike information coefficients;
sic	Schwarz (Bayesian) information coefficients;

`bic`	synonym for `sic`, and
`critical`	critical values (default).

If the variance components are unconstrained, the critical values are from a chi-square distribution with one degree of freedom. Alternatively, if they are constrained to be positive, the asymptotic distribution of test is a 50:50 mixture of chi-square distributions with zero and one degree of freedom. Essentially this means that the critical values are from a chi-square distribution with one degree of freedom but at double the probability level. See, for example, Lee, Nelder & Pawitan 2006, Section 6.5. The `CRITICAL` option can save three critical values, in a variate with units for probabilities of 0.05, 0.001 and 0.001.

The `TESTSTATISTIC` parameter can save the statistics. the `DF` parameter can save their numbers of degrees of freedom. (These will always be equal to one, but the parameter is included for compatibility with the `HGFTEST` and `HGRTEST` procedures.) The `AIC` and `SIC` parameters can save the Akaike and Schwarz (Bayesian) information coefficients, respectively. If you are making a test for a single term, you can supply a scalar for each of these parameters. However, if you have several terms, you must supply a pointer which will then be set up to contain as many scalars as there are terms.

Options: `PRINT`, `SELECTION`, `CRITICAL`, `GLSAVE`.
Parameters: `TERMS`, `TESTSTATISTIC`, `DF`, `AIC`, `SIC`.

See also
Procedures: `GLDISPLAY`, `GLKEEP`, `GLMM`, `GLPLOT`, `GLPERMTEST`, `GLPREDICT`.
Genstat Reference Manual 1 Summary section on: Regression analysis.

GPREDICTION

Produces genomic predictions (breeding values) of tested and untested individuals using phenotypic information of the tested set and the whole population genetic relationships, as inferred from molecular marker information (M. Malosetti, M.P. Boer & S.J. Welham).

Options

PRINT = *string token*	What to print (summary); default summ
PLOT = *string token*	What to plot (scatterplot, pco); default scat, pco
MODELTYPE = *string token*	Model to use to obtain the predictions (gblup, gaussian, exponential); default gblu
THETA = *variate*	Values to use for the tuning parameter θ when the model is Gaussian or exponential
SIMILARITY = *symmetric matrix*	Similarity matrix between individuals of the whole population

Parameters

TRAIT = *variates*	Quantitative trait to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
MKSCORES = *pointers*	Marker scores
IDMGENOTYPES = *texts*	Labels of the tested and untested genotypes
PREDICTIONS = *variates*	Saves the predictions
NEWGENOTYPES = *factors*	Factor to index the predictions
TESTED = *factors*	Factor that classifies NEWGENOTYPES as part of the tested or the untested set
SAVE = *pointers*	Pointer to REML save structures to save details of the analyses

Description

In genomic prediction (or genomic selection as introduced by Meuwissen *et al*. 2001), molecular markers of individuals of a population are used in combination with phenotypic information of a subset of that population (tested set) to obtain predictions (breeding values) of all the individuals of the population (i.e. both tested and untested).

GPREDICTION can be used to obtain predictions by one of three different mixed models, according to the setting of the MODELTYPE option. These differ according to the way in which the genetic variance covariance matrix is defined. The default setting, gblup, uses a realised additive relationship matrix calculated from markers, which is equivalent to the inclusion of all the markers as random explanatory variables in the model (with a common variance component). Alternatively, with the gaussian setting, a Gaussian kernel is used to model the genetic variance-covariance, which effectively accounts for non additive relationships (Gianola & van Kamp 2008, Piepho 2009). Finally, with the exponential setting, an exponential kernel is used. For the Gaussian and exponential models, an extra (tuning) parameter θ is required, which determines how covariance between individuals decays in relation to distance in the genetic space. Values for θ can be supplied, in a variate, using the THETA option. If this is unset, the value suggested by Crossa *et al*. (2010) is used (see the Method section). The SIMILARITY option can be used either to provide a similarity matrix, or to store the one that is calculated using the markers.

The TRAIT parameter must supply the observations (phenotypes) of the tested genotypes, and the GENOTYPES parameter must supply a factor to identify individuals within the tested set. The MKSCORES parameter supplies the marker scores of all the individuals in the population (tested and untested), and the IDMGENOTYPES parameter provides labels for all the genotypes in the population (tested and untested). MKSCORES must be set unless a relationship matrix has been

supplied by the SIMILARITY option. The PREDICTIONS parameter can save the predictions, the NEWGENOTYPES parameter can save a factor identifying each individual in the population, and the TESTED parameter can save a factor classifying individuals as being part of the tested or untested set.

You can set PRINT=summary to print a summary of the analysis. The SAVE parameter can save a pointer containing save structures from REML analyses that have been done.

The PLOT option controls the graphs that are produced, with settings:

scatterplot	for a scatter plot of predictions versus observed values of the tested set, and
pco	for a plot showing the first three axes of a principal coordinates analysis of the genetic similarities estimated from markers, to enable you to assess the coverage of the genetic space of the population given by the training set

Options: PRINT, PLOT, MODELTYPE, THETA, SIMILARITY.
Parameters: TRAIT, GENOTYPES, MKSCORES, IDMGENOTYPES, PREDICTIONS, NEWGENOTYPES, TESTED, SAVE.

Method

The prediction model is:

$$y = X\beta + Z u + \varepsilon$$

with u a vector of random genetic effects,

$$u \sim N(0, A\, \sigma_u^2),$$

and residuals ε with

$$\varepsilon \sim N(0, I\, \sigma^2).$$

The relationship matrix A is obtained from molecular marker information and formed depending of the model as:

Model	Relationship matrix	
GBLUP	$A = Z Z'$	Z is the genotype by markers matrix
Gaussian	$A = \exp(-D^2 / \theta)$	D^2 is the Euclidean squared distance between individuals based on markers, and θ is a tuning parameter
Exponential	$A = \exp(-D / \theta)$	D is the Euclidean distance between individuals based on markers, and θ is a tuning parameter

Before fitting the mixed model, the matrix A is checked to ensure that it is positive-semi definite. If not procedure POSSEMIDEFINITE is called to produce a positive semi-definite approximation to be used instead. If one value is set for θ, a mixed model is fitted for each value, and the Akaike Information Coefficient is used to select the best one. If no value is given for θ, then

$$\theta = \text{median}(D^2) / 2$$

is used, as suggested by Crossa *et al*. (2010).

After fitting the mixed model, predictions are formed using the VPREDICT directive.

Action with `RESTRICT`
Restrictions are not allowed.

References
Crossa, J., De Los Campos, G., Pérez, P., Gianola, D., Burgueño, J., Araus, J.L., Makumbi, D., Singh, R.P., Dreisigacker, S., Yan, J., Arief, V., Banziger, M. & Braun, H.J. (2010), Prediction of genetic values of quantitative traits in plant breeding using pedigree and molecular markers. *Genetics*, **186**,713-724.

Gianola, D. & van Kaam, J.B.C.H.M. (2008). Reproducing kernel Hilbert spaces regression methods for genomic assisted prediction of quantitative traits. *Genetics,* **178**, 2289-2303.

Meuwissen, T.H.E., Hayes, B.J. & Goddard, M.E. (2001). Prediction of total genetic value using genome-wide dense marker maps. *Genetics,* **157**, 1819-1829.

Piepho, H.P. (2009). Ridge regression and extensions for genome wide selection in maize. *Crop Science*, **49**,1165-1176.

See also
Directives: `REML`, `VPREDICT`, `PCO`.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

GRANDOM

Generates pseudo-random numbers from probability distributions (D.M. Roberts & P.W. Lane).

Options

DISTRIBUTION = *string token*	Type of distribution required (beta, chisquare, exponential, F, gamma, logNormal, Normal, t, uniform, Weibull, binomial, hypergeometric, Poisson); default Norm
NVALUES = *scalar*	Number of values to generate; default 1
SEED = *scalar*	Seed to start random number generation; default set by CALCULATE or continued from previous generation
MEAN = *scalar*	Mean for distribution, except for Weibull or hypergeometric); default 0 for Normal distribution and 1 for Poisson and exponential, otherwise *
VARIANCE = *scalar*	Variance for distribution, except for the Weibull or hypergeometric; must be positive; default *, except for Normal when default is 1
LOWER = *scalar*	Lower bound for the uniform or beta distribution; default 0
UPPER = *scalar*	Upper bound for the uniform or beta distribution; default 1
LOCATION = *scalar*	Location parameter for the log-Normal, gamma or Weibull distribution; default 0
SCALE = *scalar*	Scale parameter for the Weibull distribution; must be positive; default 1
SHAPE = *scalar*	Shape parameter for the Weibull distribution; must be positive; default 1
ABETA = *scalar*	First shape parameter for the beta distribution; must be positive; default 1
BBETA = *scalar*	Second shape parameter for the beta distribution; must be positive; default 1
AGAMMA = *scalar*	Location-scale parameter for the gamma distribution, must be positive, usually denoted by alpha or theta; default 1
BGAMMA = *scalar*	Shape parameter for the gamma distribution, must be positive, usually denoted by beta or kappa; default 1
DF = *scalar*	Number of degrees of freedom for the t or chi distribution, must be 1 or greater; default 1
DFNUMERATOR = *scalar*	Number of degrees of freedom of the numerator for the F distribution, must be 1.0 or greater; default 1
DFDENOMINATOR = *scalar*	Number of degrees of freedom of the denominator for the F distribution, must be 1.0 or greater; default 1
NBINOMIAL = *scalar*	Number of binomial trials for the binomial distribution, must be positive; default 1
PROBABILITY = *scalar*	probability of success for the binomial or hypergeometric distribution, must be positive and not greater than 1; default 0.5
NHYPERGEOMETRIC = *scalar*	Number of elements for the hypergeometric distribution, must be positive; default 1
SSHYPERGEOMETRIC = *scalar*	Sample size for the hypergeometric distribution, must be

positive and less than NHYPERGEOMETRIC; default 1

Parameter

NUMBERS = *scalar* or *variate* — The generated numbers are returned here; if the length of the supplied structure is defined, it must equal the setting of the NVALUES option

Description

GRANDOM generates pseudo-random numbers from the beta, chi-square, exponential, F, gamma, log-Normal, Normal, Student's t, uniform, Weibull, binomial, hypergeometric and poisson distributions.

The NUMBERS parameter of GRANDOM must be set to a scalar or variate to store the generated numbers. The NVALUES option can be set to specify how many values are required; if this is unset, a single value is generated. The SEED option can be set to initialize the random-number generator, hence giving identical results if the procedure is called again with the same options. If SEED is unset, generation will continue from the previous sequence in the program, or, if this is the first generation, the generator will be initialized by CALCULATE.

Most distributions can be specified by their mean and variance. In GRANDOM these are defined by the MEAN and VARIANCE options. For some distributions there are other defining parameters, which are often more convenient. These can be set by other options relevant to the distribution concerned.

Normal distributions can be defined only by mean and variance; by default these are zero and one respectively. For the exponential and Poisson distributions, either one of these is sufficient to define the distribution and if neither is given the mean is set to one. For the Poisson if both are specified they must be equal, while for the exponential the variance is the square of the mean. The chi-square distribution can be defined by any one of the DF, MEAN or VARIANCE options (the mean is equal to the degrees of freedom, and the variance to twice the degrees of freedom). Similarly, the Student's t distribution can be defined by either the DF or the VARIANCE option; if MEAN is set, it must be zero. The F distribution can be generated by setting either the MEAN and VARIANCE options or the DFDENOMINATOR and DFNUMERATOR options.

The binomial distribution can be specified either by the MEAN and VARIANCE options (with MEAN greater than VARIANCE), or by the NBINOMIAL and PROBABILITY options. However, the hypergeometric distribution cannot be specified by MEAN and VARIANCE: instead the three options PROBABILITY, NHYPERGEOMETRIC and SSHYPERGEOMETRIC must be used.

The uniform distribution in the range (0,1) can be generated by setting the single option DIST=uniform. However, you can set the MEAN and VARIANCE options, or the LOWER and UPPER options, to get a uniform distribution in any other range. Similarly, the beta distribution is generated by default in the range (0,1), by setting the MEAN and VARIANCE options, or the ABETA and BBETA options: the mean is $A/(A+B)$ and the variance is $AB/((A+B+1)\times(A+B)^2)$. By setting the LOWER and UPPER options, the four-parameter beta distribution is generated, within the specified range.

The two-parameter gamma distribution can be generated by setting either the MEAN and VARIANCE options, or the AGAMMA and BGAMMA options. (The mean is AB and the variance is AB^2: A is sometimes denoted by α or θ, and B by β or κ.) The three-parameter gamma can be generated by setting the LOCATION option, which simply has the effect of shifting a two-parameter gamma distribution. Similarly, the two- and three-parameter log-Normal distributions can be generated, though using the SCALE and SHAPE options rather than AGAMMA and BGAMMA. (If LOCATION is zero, the mean is $sc \times \exp(sh^2/2)$ and the variance is $sc^2 \times \exp(sh^2) \times (\exp(sh^2)-1$; the square of the shape parameter is the variance of the associated Normal distribution, and the log(SCALE) is the mean.) The three-parameter Weibull is defined also by the LOCATION, SCALE and SHAPE options: it cannot be specified in terms of MEAN and

VARIANCE. (The mean of the distribution is LOCATION+SCALE×G(1+1/SHAPE) and the variance is SCALE^2×G(1+2/SHAPE)−(G(1+1/SHAPE))^2), where G() is the gamma function.)

Options: DISTRIBUTION, NVALUES, SEED, MEAN, VARIANCE, LOWER, UPPER, LOCATION, SCALE, SHAPE, ABETA, BBETA, AGAMMA, BGAMMA, DF, DFNUMERATOR, DFDENOMINATOR, NBINOMIAL, PROBABILITY, NHYPERGEOMETRIC, SSHYPERGEOMETRIC.
Parameter: NUMBERS.

Method
GRANDOM uses the "table look-up" method for the majority of the distributions, using the ED** functions in the CALCULATE directive. It uses the transformation method for the Weibull distribution, and the rejection method for the binomial, hypergeometric and Poisson distributions.

Action with RESTRICT
A variate that has been restricted will receive output from GRANDOM only in those units that are not excluded by the restriction. Values in the excluded units remain unchanged. Note that the NVALUES option must equal the full size of the variate.

See also
Directive: CALCULATE.
Procedures: GRCSR, GREJECTIONSAMPLE, GRLABEL, GRMNOMIAL, GRMULTINORMAL, GRTHIN, GRTORSHIFT, SAMPLE, SVSAMPLE.
Functions: GRBETA, GRBINOMIAL, GRCHISQUARE, GRF, GRGAMMA, GRHYPERGEOMETRIC, GRLOGNORMAL, GRNORMAL, GRPOISSON, GRSAMPLE, GRSELECT, GRT, GRUNIFORM.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

GRCSR

Generates completely spatially random points in a polygon (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NPOINTS = *scalars*	How many points to generate in each polygon; no default – this parameter must be set
YCSR = *variates*	Variates to receive the vertical coordinates of the points that have been generated
XCSR = *variates*	Variates to receive the horizontal coordinates of the points that have been generated
SEED = *scalars*	Seeds for the random numbers used to generate the points; default 0

Description

The term complete spatial randomness (CSR) is used to represent the hypothesis that the overall density of events in a spatial point pattern is constant throughout the study region, and that the events are distributed independently and uniformly (see Diggle 1983). This procedure generates a simulated realization of CSR in a given polygon. The coordinates of the polygon are specified using the parameters XPOLYGON and YPOLYGON. The number of points to be generated is specified using the parameter NPOINTS. The coordinates of the points which are generated may be saved using the parameters XCSR and YCSR. The SEED parameter allows a seed to be supplied for generating the random numbers used to generate the points (thereby producing reproducible results). If this is not supplied, the default of 0 initializes the random number generator (if necessary) from the system clock.

Print output is controlled using the PRINT option. The default setting of summary prints the horizontal and vertical coordinates of the points which are generated under the headings XCSR and YCSR.

Option: PRINT.
Parameters: YPOLYGON, XPOLYGON, NPOINTS, YCSR, XCSR, SEED.

Method

A procedure PTCHECKXY is called to check that XPOLYGON and YPOLYGON have identical restrictions. The parameters XPOLYGON, YPOLYGON and NPOINTS are then passed to a sub-procedure called GRCSR_GENPTS. The sub-procedure generates points randomly in the bounding box of the polygon specified by XPOLYGON and YPOLYGON using the URAND function. It then calls PTSINPOLYGON to exclude any points which lie outside the polygon. If the number of points retained is less than NPOINTS then GRCSR_GENPTS is called again recursively until at least NPOINTS points have been generated. Finally, the EQUATE directive is used to transfer the coordinates of the first NPOINTS points generated by GRCSR_GENPTS to the parameters XCSR and YCSR.

Action with RESTRICT
If XPOLYGON and YPOLYGON are restricted, only the subset of values specified by the restriction will be included in the calculations.

Reference
Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

See also
Procedures: GRLABEL, GRTHIN, GRTORSHIFT.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Spatial statistics.

GREJECTIONSAMPLE

Generates random samples using rejection sampling (W. van den Berg).

Options

PLOT = *string tokens*	What to plot (density, sample); default dens, samp
NVALUES = *scalar*	Size of each random sample; no default, must be set
PRDENSITY = *expression structure*	Calculation defining the probability density function f(x) to sample; no default, must be set
X = *identifier*	Data structure used inside PRDENSITY for the x-coefficient of the density function f(x) no default, must be set
XLOWER = *scalar*	Lower bound of the region in which f(x) is non-negligible; default −10
XUPPER = *scalar*	Upper bound of the region in which f(x) is non-negligible; default 10
PRENVELOPE = *expression structure*	Calculation defining the probability density function g(x) used to generate the sample; default !e(PRT(X; 60))
GRENVELOPE = *expression structure*	Calculation to sample from the probability density g(x) used to generate the sample (note, PRENVELOPE and GRENVELOPE must either be both set, or both unset); default !e(GRT(NTRIES; 60))
MULTIPLIER = *scalar*	Multiplier M used in the definition of the envelope $M \times$ g(x) that must always be greater than f(x); default 10
NTRIES = *scalar*	Number of random samples to take in each sampling step; default * i.e. determined automatically

Parameters

NUMBERS = *variates*	Saves each random sample
SEED = *scalars*	Seed to use for the random numbers used to generate each random sample; default 0

Description

GREJECTIONSAMPLE generates random samples from a probability density function, using rejection sampling. The density function f(x), which need not integrate to one (e.g. a posterior in a Bayesian analysis), is specified as a Genstat expression using the PRDENSITY option. The X option specifies the identifier of the data structure used to represent the x-coordinate of the density in the calculation.

The method operates by generating a random sample from a probability density g(x), selected so that

$$f(x) \le M \times g(x)$$

where M is a suitably chosen multiplier. The density g(x) can be defined using the PRENVELOPE option, as a Genstat expression which uses the same identifier X as the PRDENSITY expression. If PRENVELOPE is not set, GREJECTIONSAMPLE uses the probability density function of a t-distribution with 60 degrees of freedom i.e.

```
PRENVELOPE = !e( PRT(X; 60) )
```

The multiplier M is specified by the MULTIPLIER option; default 10.

If you set PRENVELOPE, you must also set the GRENVELOPE option to define an expression to take a random value z from the distribution g(x). GREJECTIONSAMPLE also takes a random value

from the uniform distribution U[0,1], and accepts z if the uniform random number u is less than or equal to

$$f(z) / (M \times g(z))$$

The process works more efficiently if several values are sampled at once from $g(x)$ and U[0,1]. GREJECTIONSAMPLE can decide the number automatically; or you can define it yourself using the NTRIES option. If you are defining GRENVELOPE you should set NTRIES to the identifier of a Genstat scalar, and use this in the expression defined by GRENVELOPE. You set the scalar to a missing value if you still want GREJECTIONSAMPLE to decide the number. For example

```
SCALAR ntry
GREJECTIONSAMPLE [...\
                 X=xcoord; PRENVELOPE=!e( PRT(xcoord; 5) );\
                 GRENVELOPE=!e( GRT(ntry; 5) ); NTRIES=ntry]
```

By default,

```
GRENVELOPE = !e( GRT(NTRIES; 60) )
```

The random samples can be saved, in variates, using the NUMBERS parameter. You can use the SEED parameter to supply a seed to use in the CALCULATE directive for the sequences of the random numbers used to generate the random values from $g(x)$ and U[0,1]. The default, SEED=0, continues an existing sequence of random numbers, if any of the random-number functions has already been used in the current Genstat run. If, however, this is the first time that these functions have been used, Genstat picks a random seed.

The PLOT option controls the graphs produced by GREJECTIONSAMPLE, with settings:

density	to plot the density $f(x)$ and the envelope $M \times g(x)$, and
sample	to plot a histogram of the selected sample from $f(x)$.

By default these are both plotted.

Options: PLOT, NVALUES, PRDENSITY, X, XLOWER, XUPPER, PRENVELOPE, GRENVELOPE, MULTIPLIER, NTRIES.
Parameters: NUMBERS, SEED.

Method

For further details of the method see e.g. Carlin & Louis (2000) or Robert & Casella (2004).

References

Carlin, B.P. & T.A. Louis (2000). *Bayes and Empirical Bayes methods for Data Analysis*. Chapman & Hall, London.

Robert, C.R. & Casella, G. (2004). *Monte Carlo Statistical Methods, Second Edition*. Springer, New York.

See also

Directive: CALCULATE.
Procedures: GRCSR, GRLABEL, GRMULTINORMAL, GRTHIN, GRTORSHIFT, SAMPLE, SVSAMPLE.
Functions: GRBETA, GRBINOMIAL, GRCHISQUARE, GRF, GRGAMMA, GRHYPERGEOMETRIC, GRLOGNORMAL, GRNORMAL, GRPOISSON, GRSAMPLE, GRSELECT, GRT, GRUNIFORM.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

GRIBIMPORT

Reads data from a GRIB2 meteorological data file, and loads it or converts it to a spreadsheet file (D.B. Baird).

PRINT = *string token*	What information to print (catalogue); default cata
OUTTYPE = *string token*	Output file type (GEN, GSH, GWB, XLS, XLSX, TXT, CSV, RECORDS); default GWB
METHOD = *string token*	Whether to load data into the Genstat server after creating the file, or merely to create the file (create, load); default load
SERIAL = *string token*	Whether to store the records in series, in a single column, instead of in parallel columns (no, yes); default no
LONGITUDERANGE = *string token*	What range to use for longitude (negative, positive); default posi
MISSING = *scalar*	What value represents a missing value; default -999
GRID = *variate*	Specifies limits on the longitude and latitude for the data to be read; default * i.e. read all grid points
ENDTIME = *string token*	Whether to keep the end time for each period when SERIAL = yes (yes, no); default no
SCOPE = *string token*	Whether to create the data locally in a procedure that is using GRIBIMPORT, or globally in the whole program (local, global); default loca

Parameters

FILE = *texts*	Input file or URL to be read
OUTFILE = *texts*	Name of the output file to be created; if this is not provided a temporary file will be created, and then deleted if the data are loaded
RECORDS = *scalars* or *variates*	The numbers of the records to read; default is to read all the records in the file
MATCH = *texts*	Text strings to match in the record descriptions; default * requests all the records selected by RECORDS
COLUMNS = *texts*	Names and/or type codes for the columns that are read (the type of column can be forced by ending the column name, if supplied, with the code **!** for a factor, **#** for a variate, and **$** for a text), using a name of '*' will cause a column to be dropped
ISAVE = *pointers*	Saves the identifiers of the columns

Description

The name of the file, containing the data values to be imported, is specified by the FILE parameter. This can also be an internet URL prefixed with http://, https:// ftp:// or file://. The data source is then downloaded and imported. This procedure requires wgrib2.exe available from http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2 to be installed. Open a GRIB file in the Genstat client to get help installing this. Note, this procedure does not support the older GRIB 1 data format.

Data in the file are extracted and saved in the specified file format, depending on the extension of OUTFILE. If this is not provided, the type is indicated by the OUTTYPE option, as either GEN (Genstat Command file), GSH (Genstat Spreadsheet), GWB (Genstat Spreadsheet Book), XLS (Excel 5 Spreadsheet), XLSX (Excel 2007 Spreadsheet), TXT (ASCII Text file) or CSV (comma-delimited file); the default is GWB. Setting OUTTYPE=RECORDS reads the record names

descriptions and grid definitions into columns `Id`, `Message`, `LonMin`, `LonMax`, `LatMin`, `LatMax`, `NPoints`; these give the record number, description, minimum and maximum longitude and latitude, and number of grid points, respectively. If `PRINT=catalogue`, the record details are printed in the output window.

If `METHOD=load`, the resulting file is read into Genstat data structures. When `GRIBIMPORT` is used within a procedure, the `SCOPE` option controls whether the structures are created locally in the procedure (default), or globally in the main program.

The `RECORDS` and `MATCH` parameters can be used to read just a selection of the records in the file. `RECORDS` identifies records by non-negative integers (e.g. 0,1,2...), which may contain a sub-record number after the decimal point (e.g. 8.1, 8.2). `MATCH` specifies strings of characters to indicate records to read: a record is read if its description contains any of the strings specified by `MATCH`. This can be used to select all variables with a specific time, type, level or other option (e.g. `'d=2018100812'`, `'TMP:'` or `'MM-ENS=2'`). If both `RECORDS` and `MATCH` are specified, their actions are combined to include all records selected by either parameter. If neither is specified, all the records are read. The `COLUMNS` parameter can be used to set the names and types of the structures in the columns (see below).

The `LONGITUDERANGE` controls the range of values used for longitudes: `negative` uses values between -180 and 180, and `positive` uses values between 0 and 360.

The `GRID` option can be used to read a sub-grid of values by providing a variate of length 4 containing the minimum and maximum longitude, and the minimum and maximum latitude, of the sub-grid. The `GRID` values should use the longitude range within the GRIB file, rather than that specified by the `LONGITUDERANGE` option.

The `MISSING` option specifies a value that represents missing values within the records. If `MISSING = *`, rows for values that are undefined will not be output, giving incomplete grids. These may cause problems with multiple records in parallel format. So, if there are undefined values in the records when reading in parallel format, it is best to set `MISSING` to a value that does not occur any of the records.

The `SERIAL` option indicates how data are to be imported, with the following settings:

`no`	each record is loaded into its own column with two factors `Longitude` and `Latitude` at the start of the data to index the grid; and
`yes`	all the record data are loaded into a single column, `Measurement`, with factors `Time0`, `Time1` (if `ENDTIME = yes`), `Variable`, `Level`, `Longitude` and `Latitude` at the start of the data to index the start and end times, record variable, level and grid.

If `SERIAL= yes`, the `ENDTIME` option controls whether an end time for a measurement (`Time1`) is included in the data. The end time is used for variables that are taken over a period such as rainfall, a maximum wind gust or an average temperature. If no variables of this type are in your data, the end time will be the same as the start time, and can safely be dropped (the default setting).

The `COLUMNS` parameter can be used to specify the names for the columns, in a text. The type of each column can be forced by providing a **`!`**, **`#`** or **`$`** character on the end of its string. A string **`'*'`** can be given as a name in `COLUMNS`, to remove a column from the data that are read. If only a single type character is given, only the types of the column (and not its names) is changed.

Options: `PRINT`, `OUTTYPE`, `METHOD`, `SERIAL`, `LONGITUDERANGE`, `MISSING`, `GRID`, `ENDTIME`, `SCOPE`.

Parameters: `FILE`, `OUTFILE`, `RECORDS`, `MATCH`, `COLUMNS`, `ISAVE`.

Method
The request is passed to the DATALOAD.DLL library which uses wgrib2.exe to read the file, and save the results to OUTFILE, or to a temporary file if OUTFILE is not specified.

Action with RESTRICT
Restrictions are not applicable to any of the parameters.

See also
Directive: SPLOAD.
Procedures: IMPORT, WINDROSE.
Genstat Reference Manual 1 Summary section on: Input and output.

GRLABEL

Randomly labels two or more spatial point patterns (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string token*	What to print (summary); default summ
SEED = *scalar*	Seed for the random numbers used to create the random labellings; default 0

Parameters

OLDY = *variates*	Vertical coordinates of two or more spatial point patterns; no default – this parameter must be set
OLDX = *variates*	Horizontal coordinates of two or more spatial point patterns; no default – this parameter must be set
NEWY = *variates*	Variates to receive the vertical coordinates of the spatial point patterns created by random labelling
NEWX = *variates*	Variates to receive the horizontal coordinates of the spatial point patterns created by random labelling

Description

This procedure pools the coordinates of two or more spatial point patterns (specified using the OLDX and OLDY parameters) and randomly groups (or labels) the points to form *n* new patterns, where *n* is the number of patterns supplied using OLDX and OLDY. Each new pattern contains the same number of points as its old counterpart. The coordinates of the new patterns can be saved using the parameters NEWX and NEWY. The SEED option allows a seed to be supplied for generating the random numbers used to create the random labelling (thereby producing reproducible results). If this is not supplied, the default of 0 initializes the random number generator (if necessary) from the system clock.

Printed output is controlled using the PRINT option. The default setting of summary prints the coordinates of each randomly labelled pattern under the headings NEWX[i] and NEWY[i], (i = 1 ... *n*).

Options: PRINT, SEED.
Parameters: OLDY, OLDX, NEWY, NEWX.

Method

A procedure PTCHECKXY is called to check that each pair of structures in OLDX and OLDY have identical restrictions. The procedure APPEND is then used to create a single variate containing the horizontal coordinates of all the point patterns and a factor whose levels indicate the source (original label) of the points. The vertical coordinates are combined in a similar way. The URAND and SORT functions are then used to randomly permute the labels. Finally, the RESTRICT directive is used to extract the horizontal and vertical coordinates corresponding to each level of the permuted factor.

Action with RESTRICT

If any of the variates in OLDX and OLDY are restricted, only the subset of values specified by the restriction will be included in the calculations.

See also

Procedures: GRCSR, GRTHIN, GRTORSHIFT.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Spatial statistics.

GRMNOMIAL

Generates multinomial pseudo-random numbers (D.B. Baird).

Options

NVALUES = *scalar*	Number of values to generate
SEED = *scalar*	Seed to generate the random numbers; default 0 continues an existing sequence or initializes the sequence automatically if no random numbers have been generated in this job

Parameters

PROBABILITIES = *variates* or *tables*	Probabilities for the categories
NUMBERS = *factors*	Saves the random numbers
COUNTS = *tables*	Saves counts of the numbers generated in each category

Description

GRMNOMIAL generates pseudo-random numbers from a multinomial distribution. The probabilities for the categories are specified by the PROBABILITIES option, in either a variate or a table.

The random numbers can be saved, in a factor, using the NUMBERS parameter. The NVALUES option specifies how many to generate. If this is not set, the length of the NUMBERS factor is used or, if that has not been defined, a single random number is generated.

The COUNTS parameter can save a table with counts of the numbers generated in each category. If COUNTS has not already been defined as a table with a suitable classifying factor, it is defined as follows. Firstly, if NUMBERS has been set, COUNTS is defined as a table with NUMBERS as the classifying factor. Otherwise, if PROBABILITIES has supplied a table rather than a variate, COUNTS is defined as a table classified by the same classifying factor as PROBABILITIES. Finally, the fall-back is to define COUNTS as a table with an unnamed classifying factor.

The SEED option can be set to initialize the random-number generator. The default of zero continues an existing sequence, or initializes the sequence automatically if no random numbers have been generated in this job.

Options: NVALUES, SEED.
Parameters: PROBABILIIES, NUMBERS, COUNTS.

Method

The pseudo-random numbers are generated using the GRUNIFORM function.

Action with RESTRICT

Any restrictions are ignored.

Directive: CALCULATE.
Procedures: GRANDOM, GRCSR, GREJECTIONSAMPLE, GRLABEL, GRMULTINORMAL, GRTHIN, GRTORSHIFT, SAMPLE, SVSAMPLE.
Functions: GRBETA, GRBINOMIAL, GRCHISQUARE, GRF, GRGAMMA, GRHYPERGEOMETRIC, GRLOGNORMAL, GRNORMAL, GRPOISSON, GRSAMPLE, GRSELECT, GRT, GRUNIFORM.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

GRMULTINORMAL

Generates multivariate Normal pseudo-random numbers (P.W. Goedhart & K.L. Moore).

Options

NVALUES = *scalar*	Number of values to generate; default 1
MEANS = *variate*	The mean for the multivariate Normal distribution; default is a variate with values all equal to 0
VCOVARIANCE = *symmetric matrix*	The variance/covariance matrix for the multivariate Normal distribution; default is to use an identity matrix
SEED = *scalar*	Seed to generate the random numbers; default 0 continues an existing sequence or initializes the sequence automatically if no random numbers have been generated in this job

Parameters

NUMBERS = *pointers* or *matrices*	Saves the random numbers as either a pointer to a set of variates or a matrix

Description

GRMULTINORMAL generates pseudo-random numbers from a multivariate Normal distribution $N_p(\mu, \Sigma)$. The mean μ is specified by the option MEANS as a variate of length p; the variance-covariance matrix Σ is specified by the option VCOVARIANCE as a symmetric matrix with p rows and columns; and the option NVALUES specifies the number of values n to be generated. Note that VCOVARIANCE must be positive semi-definite.

The numbers can be saved using the NUMBERS parameter, in either a pointer to a set of variates, or a matrix. If the NUMBERS structure or structures are already declared, their dimensions must be compatible with the settings of the NVALUES, MEANS and VCOVARIANCE options. The dimensions are also used, if necessary, to set defaults for the options. By default, MEANS is taken to be a variate of zero values, and VCOVARIANCE is taken to be the identity matrix. If the setting of NUMBERS is not already declared, if will be defined as a pointer to a set of variates with dimensions deduced from the option settings.

Options: NVALUES, MEANS, VCOVARIANCE.
Parameter: NUMBERS.

Method

Pseudo-random numbers from a multivariate Normal distribution are generated by forming a matrix Y of columns of univariate Normal random numbers, using the Box-Muller method (Box & Muller 1958), followed by a linear transformation

$$X = A Y + \mu,$$

where A is calculated by a CHOLESKI decomposition, $AA' = \Sigma$. (See, for example, Johnson 1987 pages 52-55, Tong 1990 pages 181-186).

Action with RESTRICT

Variates that have been restricted will receive output from GRMULTINORMAL only in those units that are not excluded by the restriction. Values in the excluded units remain unchanged. Note that the NVALUES option must equal the full size of the variates. Restrictions on the MEANS variate are ignored.

References

Box, G.E.P. & Muller, M.E. (1958). A note on generation of normal deviates. *Annals of Mathematical Statistics*, **28**, 610-611.

Johnson, M.E. (1987). *Multivariate Statistical Simulation*. John Wiley & Sons, New York.

Tong, Y.L. (1990). *The Multivariate Normal Distribution*. Springer-Verlag, New York.

See also

Directive: CALCULATE.

Procedures: GRANDOM, GRCSR, GREJECTIONSAMPLE, GRLABEL, GRMNOMIAL, GRTHIN, GRTORSHIFT, SAMPLE, SVSAMPLE.

Functions: GRBETA, GRBINOMIAL, GRCHISQUARE, GRF, GRGAMMA, GRHYPERGEOMETRIC, GRLOGNORMAL, GRNORMAL, GRPOISSON, GRSAMPLE, GRSELECT, GRT, GRUNIFORM.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

GRTHIN

Randomly thins a spatial point pattern (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

OLDY = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
OLDX = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
NPOINTS = *scalars*	How many points to return from each pattern; no default – this parameter must be set
NEWY = *variates*	Variates to receive the vertical coordinates of the randomly thinned patterns
NEWX = *variates*	Variates to receive the horizontal coordinates of the randomly thinned patterns
SEED = *scalars*	Seeds for the random numbers used to select the thinned points; default 0
THINNED = *variates*	Variate whose values indicate whether the coordinates of each spatial point pattern are included (1) or excluded (0) from the thinned pattern

Description

This procedure randomly thins a spatial point pattern with coordinates specified by the parameters OLDX and OLDY. The number of points required in the thinned pattern is specified using the NPOINTS parameter which must be a positive integer. If NPOINTS is equal to or greater than the number of points in the original pattern then no points will be deleted. The coordinates of the points which remain after thinning can be saved using the parameters NEWX and NEWY. The SEED parameter allows a seed to be supplied for generating the random numbers used to select the thinned points (thereby producing reproducible results). If this is not supplied, the default of 0 initializes the random number generator (if necessary) from the system clock. The THINNED parameter can be used to save a logical variable containing the value 1 when a coordinate is used within the thinned pattern and 0 if the coordinate has not been selected in the thinned pattern.

Printed output is controlled by the PRINT option. The default setting of summary prints the horizontal and vertical coordinates of the points which remain after thinning under the headings NEWX and NEWY.

Option: PRINT.
Parameters: OLDY, OLDX, NPOINTS, NEWY, NEWX, SEED, THINNED.

Method

A procedure PTCHECKXY is called to check that OLDX and OLDY have identical restrictions. A dummy variate with the same number of elements as OLDX and OLDY and containing uniform random numbers in the interval (0,1) is created using the URAND function. The SORT function is then used to sort the elements of OLDX and OLDY into the order which would put the elements of the dummy variate in ascending order. Finally, the first NPOINTS elements of the sorted coordinates are transferred to the variates NEWX and NEWY using the EQUATE directive.

Action with `RESTRICT`
If `OLDX` and `OLDY` are restricted, only the subset of values specified by the restriction will be included in the calculations.

See also
Procedures: `GRCSR`, `GRLABEL`, `GRTORSHIFT`.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Spatial statistics.

GRTORSHIFT

Performs a random toroidal shift on a spatial point pattern (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

OLDY = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
OLDX = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
YBOX = *variates*	Vertical coordinates of the toroidal regions
XBOX = *variates*	Horizontal coordinates of the toroidal regions
NEWY = *variates*	Variates to receive the vertical coordinates of the randomly shifted patterns
NEWX = *variates*	Variates to receive the horizontal coordinates of the randomly shifted patterns
SEED = *scalars*	Seeds for the random numbers used to perform the shifts; default 0

Description

A simple shift of a spatial point pattern is obtained by applying constant horizontal and vertical displacements to every event in the pattern. Although the pattern itself is translated, its internal structure is preserved. In a random shift, the vertical and horizontal displacements are obtained using a pseudo-random number generator.

Applying a random shift to a spatial point pattern contained in a polygonal study region may move some of the events outside the polygon. If the polygon is a rectangle, then events which are moved outside can be mapped onto the rectangle by assuming that the pattern repeats periodically at the scale of the rectangle. This may be achieved by wrapping the rectangle on a torus, so that the top edge is connected to the bottom edge, and the left-hand edge to the right-hand edge. An event which moves beyond the right-hand boundary will re-enter from the left, and so on. This process is termed toroidal edge-correction.

The procedure GRTORSHIFT performs a random toroidal shift on a spatial point pattern given the coordinates of the pattern (specified using the OLDX and OLDY parameters) and the coordinates of a rectangle on which the shift is to be performed (specified using the XBOX and YBOX parameters). The default values of XBOX and YBOX are the coordinates of the bounding box of the spatial point pattern (see the procedure PTBOX). The coordinates of the shifted pattern can be saved using the parameters NEWX and NEWY. The SEED parameter allows a seed to be supplied for generating the random numbers used to perform the random shift (thereby producing reproducible results). If this is not supplied, the default of 0 initializes the random number generator (if necessary) from the system clock.

Printed output is controlled using the PRINT option. The default setting of summary prints the coordinates of the shifted pattern under the headings NEWX and NEWY.

Option: PRINT.
Parameters: OLDY, OLDX, YBOX, XBOX, NEWY, NEWX, SEED.

Method
A procedure PTCHECKXY is used to check that OLDX and OLDY have identical restrictions. If either XBOX or YBOX is unset, the procedure PTBOX is then used to assign the corresponding coordinates of the bounding box for the point pattern specified by OLDX and OLDY (any values supplied for XBOX and YBOX will remain unchanged). PTCHECKXY is then used to check that XBOX and YBOX have identical restrictions. The URAND function is then used to generate the horizontal and vertical displacements. The coordinates of the pattern and the horizontal and vertical displacements are passed to a sub-procedure (TORSHIFT) which performs the toroidal shift.

Action with RESTRICT
If OLDX and OLDY are restricted, only the subset of values specified by the restriction will be included in the calculations. XBOX and YBOX may also be restricted, as long as the same restrictions apply to both parameters.

See also
Procedures: GRCSR, GRLABEL, GRTHIN.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Spatial statistics.

GSTATISTIC

Calculates the gamma statistic of agreement for ordinal data (A.W. Gordon).

Options

PRINT = *string token*	Whether to print the statistic with its associated information and the resulting test (test); default test
METHOD = *string token*	Type of test required (twosided, positive, negative); default twos

Parameters

DATA = *tables*	Tables of data each classified by the two variables (factors) of interest
STATISTIC = *scalars*	Save the value of gamma for each data table
VARIANCE = *scalars*	Save the corresponding variances

Description

The gamma statistic (Siegel & Castellan 1988, pages 291-298) provides a way of assessing the agreement between two variables measured using ordinal scales. In Genstat these would each be represented as factors whose levels represent a ranking of the individuals according to some measurement.

For example, suppose we have a factor A with r levels and a factor B with k levels. The data for GSTATISTIC, specified by the DATA parameter, consists of an r by k table classified by A and B, whose entries indicate the number of times that the ith level of variable A occurs with the jth level of variable B. The table must not contain any missing values. The statistic has the value 1 when there is no disagreement in the ordering of the variables, −1 if the ordering defined by A has no disagreement with the reverse of the ordering defined by B, and zero if the variables are independent.

The printing of the test statistic and its associated information is controlled by the PRINT option. With the default, test, the procedure prints the number of times that the variables agree and disagree, the resulting value of gamma and its variance. When the number of observations N is large, the sampling distribution of gamma is approximately Normal. The procedure thus also prints the value of gamma divided by the variance, and its probability assuming a Normal distribution. A warning is printed if N is less than 20.

The test is assumed to be two-sided (i.e. no prior knowledge is assumed about the type of association) unless otherwise requested by the METHOD option. Setting METHOD=positive will give a one-sided test of the null hypothesis that there is a positive association. Similarly, METHOD=negative will produce a one-sided test that there is a negative association.

The STATISTIC and VARIANCE parameters allow gamma and its variance to be saved, in scalars.

Option: PRINT, METHOD.
Parameters: DATA, STATISTIC, VARIANCE.

Method

The method used is as described in Siegel & Castellan (1988, pages 291-298).

Reference

Siegel, S. & Castellan, N.J. (1988). *Nonparametric Statistics for the behavioural sciences (second edition)*. McGraw-Hill, New York.

See also
Procedure: KAPPA.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

G2AEXPORT

Forms a dbase file to transfer ANOVA output to Agronomix Generation II (R.W. Payne).

Options

PRINT = *strings*	Controls printed output (columns); default * i.e. none
REPLICATETERMS = *formula*	Specifies the term or terms that define the replication in the design
METHOD = *string*	How to form the means (loweststratum, combined); default lowe
ALPHALEVEL = *scalar*	Alpha value to use when calculating least significant differences; default 0.05
TAIL = *scalar*	Number of tails in the calculation of least significant differences (1, 2); default 1
SAVE = *ANOVA save structure*	Save structure for the analysis from which the means &c are to be saved; default * takes the information from the most recent ANOVA analysis

Parameters

MEANTERM = *formula*	Defines the treatment term whose means are to be saved; no default (must be specified)
OUTFILE = *text*	Name of the output file (dbf) to form; default * i.e. file not formed

Description

G2AEXPORT can be used after a Genstat ANOVA analysis, to write a dbase file with a table of means and associated information to be loaded into the Agronomix Generation II system (see agronomix.com).

Printed output is controlled by the PRINT option, with settings:

columns	to print the columns of information to be saved.

By default the means and other information are taken from the analysis of the last y-variate to have been analysed by ANOVA. Alternatively, you can take the information from an analysis of another y-variate, by saving a save structure using the SAVE parameter of ANOVA when it is analysed, and then supplying this to G2AEXPORT using its SAVE option.

The MEANTERM parameter specifies a formula defining the term whose means are to be saved; note that only one table of means can be saved in each call of the G2AEXPORT. The OUTFILE parameter specifies the file (assumed to be a dbase file) where the information is to be stored. The means are usually constructed in the standard way of the ANOVA directive, namely by taking the treatment effects from the lowest stratum where they are estimated. However, you can set option METHOD=combined to obtain means that combine information from every stratum where the relevant treatment effects are estimated.

The ALPHALEVEL option specifies the alpha value to use in the calculation of least significant differences that accompany the table of means (default 0.05), and the TAIL option specifies whether this is to be for a 1 or 2-sided test (default 1).

The REPLICATETERMS option can supply a model formula to specify one or more model terms defining complete replications of the treatments: for example, blocks in a complete randomized block design, or rows and columns in a Latin square.

Options: PRINT, REPLICATETERMS, METHOD, ALPHALEVEL, TAIL, SAVE.
Parameters: MEANTERM, OUTFILE.

Method

The information is mainly obtained using AKEEP. The first column (called NAME) describes the contents of each row. Then there is a column for every factor in the table of means, indexing the column of means (called AVG) which comes next. The ranks of the means are in the subsequent column (called RANK), and the next column (called CV) saves the standard deviation of the observations on each combination of the levels of the mean factors, expressed as a percentage of their mean. Finally, if the means are unequally replicated there is a column saving the replication of each mean.

At the top of the columns, there is a row for each mean in the table. Then there are some extra rows with the following names (in the NAME column) and information (in the AVG column):

GRAND MEAN	the grand (i.e. overall) mean;
CV	the coefficient of variation for the lowest stratum in which the maximal model term in the table of means (e.g. A.B for an A-by-B table of means) is estimated;
LSD	saves the least significant difference for the table of means if this is the same for all comparisons of means within the table, otherwise this is replaced by three rows with the minimum, average and maximum LSD (Min LSD, LSD and Max LSD);
Residual	the residual mean square for the lowest stratum in which the maximal model term in the table of means is estimated;
SED	saves the standard error of differences for the table of means if this is the same for all comparisons of means within the table, otherwise this is replaced by three rows with the minimum, average and maximum SED (Min SED, SED and Max SED);
Alpha level	alpha level used in the calculation of the LSDs (ALPHALEVEL option);
R Square	the value of R-square for analysis down to the lowest stratum in which the maximal model term in the table of means is estimated (this ensures that any lower strata that represent within-cell replication are ignored);
No. of Reps	saves replication of the table of means if this is the same for every mean in the table, otherwise this is replaced by three rows with the minimum, average and maximum replication (Min no. of Reps, No. of Reps and Max no. of Reps);
RE-RCBD	the efficiency factor of the maximal model term in the table of means, expressed as a percentage;
Rep-Msqr	the mean square of the REPLICATIONTERMS;
Heritability	this row is include for compatibility with the output that G2VEXPORT constructs following REML, but cannot be calculated for ANOVA analyses;
Prob. Entry	the F probability of the variance ratio of the maximal model term in the table of means;
Error d.f.	the residual degrees of freedom for the lowest stratum in which the maximal model term in the table of means is estimated;
Tail	Number of tails in the calculation of the least significant differences (TAIL option).

Action with `RESTRICT`
If the `Y` variate in the `ANOVA` was restricted, only the units not excluded by the restriction will have been analysed.

See also
Directive: `ANOVA`.
Procedures: `G2AFACTORS`, `G2VEXPORT`.
Genstat Reference Manual 1 Summary section on: Analysis of variance.

G2AFACTORS

Redefines block and treatment variables as factors (R.W. Payne).

No options

Parameter

FACTOR = *variates* or *texts*	Other variates or texts to convert into factors (if required)

Description

G2AFACTORS is one of a suite of procedures provided to simplify the use of Genstat to analyse data from the Agronomix Generation II system (see agronomix.com).

Data can be transferred to Genstat by writing a dbase file within Generation II, and reading this into Genstat using the IMPORT procedure. By default, IMPORT loads columns containing textual strings into Genstat text structures, and columms of numbers into variates. So, before data are analysed, some of the columns will need to be defined as factors. IMPORT allows this to be done by setting its COLUMNS parameter. An alternative (and simpler) method may be to use G2AFACTORS.

If analysis of variance is to be used, the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives will need to be used to define the block and treatment models for the analysis. If you then call G2AFACTORS, it will look through the models and redefines any texts or variates that they contain as factors, automatically, ready for the analysis (by the ANOVA directive).

If the analysis is to be done in some other way (e.g. by REML) you can still use G2AFACTORS, and use the FACTOR parameter specify the variates and texts that need to be converted,

Options: none.
Parameter: FACTOR.

Method

The GET directive is used to access the current settings define by the BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives. The FCLASSIFICATION directive obtains the list of variables involved in the model, and the GROUPS directive (with option REDEFINE=yes) does the redefinition.

See also

Directives: ANOVA, BLOCKSTRUCTURE, TREATMENTSTRUCTURE, REML.
Procedures: G2AEXPORT, G2VEXPORT.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, REML analysis of linear mixed models.

G2VEXPORT

Forms a dbase file to transfer REML output to Agronomix Generation II (R.W. Payne).

Options

PRINT = *strings*	Controls printed output (columns); default * i.e. none
REPLICATETERMS = *formula*	Specifies the term or terms that define the replication in the design
MODEL = *formula*	Indicates which model terms (fixed and/or random) are to be used in forming the predictions; default * includes all the fixed terms and relevant random terms
OMITTERMS = *formula*	Specifies terms to be excluded from the MODEL; default * i.e. none
FACTORIAL = *scalar*	Limit on the number of factors or variates in each term in the models specified by MODEL or OMITTERMS; default 3
PRESENT = *identifiers*	Lists factors for which averages should be taken across combinations that are present
WEIGHTS = *tables*	One-way tables of weights classified by factors in the model; default *
ALPHALEVEL = *scalar*	Alpha value to use when calculating least significant differences; default 0.05
TAIL = *scalar*	Number of tails in the calculation of least significant differences (1, 2); default 1
SAVE = *REML save structure*	Save structure for the analysis from which the means &c are to be saved; default * takes the information from the most recent REML analysis

Parameters

MEANTERM = *formula*	Defines the treatment term whose means are to be saved; no default (must be specified)
OUTFILE = *text*	Name of the output file (dbf) to form; default * i.e. file not formed

Description

G2VEXPORT can be used after a Genstat REML analysis, to write a dbase file with a table of means and associated information to be loaded into the Agronomix Generation II system (see agronomix.com).

Printed output is controlled by the PRINT option, with settings:

columns	to print the columns of information to be saved.

By default the means and other information are taken from the analysis of the last y-variate to have been analysed by REML. Alternatively, you can take the information from an analysis of another y-variate, by saving a save structure using the SAVE parameter of REML when it is analysed, and then supplying this to G2VEXPORT using its SAVE option.

The MEANTERM parameter specifies a formula defining the term whose means are to be saved; note that only one table of means can be saved in each call of the G2VEXPORT. The OUTFILE parameter specifies the file (assumed to be a dbase file) where the information is to be stored. The means are calculated using the VPREDICT directive. Options MODEL, OMITTERMS, FACTORIAL, PRESENT and WEIGHTS (which all operate exactly as in VPREDICT) are provided to control how this is done.

The ALPHALEVEL option specifies the alpha value to use in the calculation of least significant differences that accompany the table of means (default 0.05), and the TAIL option specifies

whether this is to be for a 1 or 2-sided test (default 1).

The REPLICATETERMS option can supply a model formula to specify one or more model terms defining complete replications of the treatments: for example, blocks in a complete randomized block design, or rows and columns in a Latin square.

Options: PRINT, REPLICATETERMS, MODEL, OMITTERMS, FACTORIAL, PRESENT, WEIGHTS, ALPHALEVEL, TAIL, SAVE.
Parameters: MEANTERM, OUTFILE.

Method

The information is mainly obtained using VKEEP and VPREDICT. The first column (called NAME) describes the contents of each row. Then there is a column for every factor in the table of means, indexing the column of means (called AVG) which comes next. The ranks of the means are in the subsequent column (called RANK), and the next column (called CV) saves the standard deviation of the observations on each combination of the levels of the mean factors, expressed as a percentage of their mean. Finally, if the means are unequally replicated there is a column saving the replication of each mean.

At the top of the columns, there is a row for each mean in the table. Then there are some extra rows with the following names (in the NAME column) and information (in the AVG column):

GRAND MEAN	the grand (i.e. overall) mean;
CV	the coefficient of variation for the lowest stratum in which the maximal model term in the table of means (e.g. A.B for an A-by-B table of means) is estimated;
LSD	saves the least significant difference for the table of means if this is the same for all comparisons of means within the table, otherwise this is replaced by three rows with the minimum, average and maximum LSD (Min LSD, LSD and Max LSD);
Residual	the residual mean square for the lowest stratum in which the maximal model term in the table of means is estimated;
SED	saves the standard error of differences for the table of means if this is the same for all comparisons of means within the table, otherwise this is replaced by three rows with the minimum, average and maximum SED (Min SED, SED and Max SED);
Alpha level	alpha level used in the calculation of the LSDs (ALPHALEVEL option);
R Square	the value of R-square for analysis down to the lowest stratum in which the maximal model term in the table of means is estimated (this ensures that any lower strata that represent within-cell replication are ignored);
No. of Reps	saves replication of the table of means if this is the same for every mean in the table, otherwise this is replaced by three rows with the minimum, average and maximum replication (Min no. of Reps, No. of Reps and Max no. of Reps);
RE-RCBD	this rows is include for compatibility with the output that G2AEXPORT constructs following ANOVA, but it is not useful for the unbalanced designs that REML usually analyses (with G2AEXPORT, it is the efficiency factor of the maximal model term in the table of means, e.g. A.B for an

	A-by-B table of means, expressed as a percentage);
Rep-Msqr	the mean square of the REPLICATIONTERMS;
Heritability	the efficiency factor of the maximal model term in the table of means;
Prob. Entry	the probability of the maximal model term in the table of means, calculated from the Wald statistic if MEANTERM is a fixed term, or from differences of deviances if it is a random term (note: the probability will then test for the inclusion not only of MEANTERM but also for any of its interactions);
Error d.f.	the residual degrees of freedom;
Tail	Number of tails in the calculation of the least significant differences (TAIL option).

Action with RESTRICT
If the Y variate in the REML was restricted, only the units not excluded by the restriction will have been analysed.

See also
Directive: REML.
Procedures: G2AEXPORT, G2AFACTORS.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

HANOVA

Does hierarchical analysis of variance/covariance for unbalanced data (P.W. Lane).

Options

PRINT = *string token*	Which analyses to print (all, some, none); default all
INCHANNEL = *scalar*	Channel from which to read data; default * specifies that the data values are already stored in the factors and variates specified by the parameters of HANOVA
FORMAT = *variate*	Format for reading data; default * requests free format
ANALYSIS = *symmetric matrix*	For PRINT=some, this indicates which analyses to print
SSPM = *SSPM*	Stores the corrected sums of squares and products; default *
COEFFICIENT = *matrix*	Stores the estimated variance and co-variance components; default *

Parameters

VARIATES = *pointers*	Variates to be analysed
FACTORS = *pointers*	Factors defining the hierarchy, the first factor of the pointer defining the first stratum, and so on

Description

Procedure HANOVA performs hierarchical analysis of variance and covariance, estimating the components of variance corresponding to each level of a nested classification. It uses the method of Gower (1962), which is based on the method of moments. This method is less efficient than REML, and may produce different results. However, it does not require the assumption of Normal distributions for the random terms.

Data are said to be classified hierarchically if the units have several groupings successively nested within each other. One way of representing such a classification would be to identify the groupings in each stratum of the hierarchy by a single factor; two units with the same value for one of the factors would then be required to have the identical values for the factors representing the previous strata. An alternative method is to use not only the factor for the current stratum, but also the factors for previous strata, to indicate the groupings that occur there. For example, the following classifications are effectively equivalent:

	(1)		(2)	
Unit	Factor 1	Factor 2	Factor 1	Factor 2
	(stratum 1)	(stratum 2)	(stratum 1)	(stratum 2)
1	1	1	1	1
2	1	1	1	1
3	1	2	1	2
4	2	3	2	1
5	2	4	2	2

Thus, in the second form of representation, the second factor indicates the sub-divisions within each group in the first stratum, using the same levels each time. This more efficient method is the one required by HANOVA.

The simplest way to use HANOVA is to set the VARIATES parameter to a single variate (or to a pointer if several variates are to be analysed), and set the FACTORS parameter to a pointer of factors. The factors must be in the order of the hierarchy with the first factor defining the coarsest grouping of the units and succeeding factors being nested within the first. The units of data stored in the variates and factors can be in any order.

Since hierarchical data can often be extensive, HANOVA can be requested to read the data sequentially, tabulating it with respect to the factors, so that the data need not all be held in core at the same time. The INCHANNEL defines the channel number of the file from which the data are to be read; if INCHANNEL is not set, the data are assumed to be present already, in the factors and variates contained in the VARIATES and FACTORS parameters. The FORMAT option allows a variate to be specified for use in the FORMAT option of the READ command within the procedure; if this is not set, the default format of READ is assumed.

If a unit has a missing value for any of the variates or factors, it is omitted from all the analyses. The procedure carries out analyses of variance for specified variates, and of covariance for specified pairs of variates. Variance components are calculated for each stratum: that is, the proportion of the total variance per individual ascribable to the various strata of the classification.

Output is controlled by the PRINT option: by default, the matrix of coefficients of variance components is printed, followed by an analysis of variance of each variate and of covariance of each pair of variates. To obtain only some of the analyses, option PRINT should be set to some, and the ANALYSIS option to a symmetric matrix with numbers of rows and columns equal to the number of variates. A non-zero value in the matrix indicates that the corresponding analysis of variance or covariance is to be displayed. Printed output can be suppressed by setting PRINT=none.

The matrix of coefficients can be saved using the COEFFICIENTS option, and the sum of squares and products of the variates using the SSPM option.

Options: PRINT, INCHANNEL, FORMAT, ANALYSIS, COEFFICIENT, SSPM.
Parameters: VARIATES, FACTORS.

Method

HANOVA uses the method described by Gower (1962).

Action with RESTRICT

Account is taken of restriction on any factor, or on the first variate in the VARIATES parameter: subsequent variates must either have the same restriction, or be unrestricted.

Reference

Gower, J.C. (1962). Variance component estimation for unbalanced hierarchical classifications. *Biometrics*, **18**, 537-542.

See also

Directive: REML.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

HBOOTSTRAP

Performs bootstrap analyses to assess the reliability of clusters from hierarchical cluster analysis (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (clusters, dendrograms); default * i.e. none
METHOD = *string token*	Criterion for forming clusters (singlelink, nearestneighbour, completelink, furthestneighbour, averagelink, mediansort, groupaverage); default sing
CLIMIT = *scalar*	Similarity value below which clusters are not recorded; default 0
UNITS = *text* or *variate*	Names to label the units of the clusters when they are printed; default *
MINKOWSKI = *scalar*	Index *t* for use with TEST=minkowski
CLUSTERS = *pointer*	Specifies or saves the clusters
REPLICATION = *variate*	Saves the replication of the clusters in the bootstrap samples
NDATASAMPLE = *scalar*	Number of DATA vectors to take in each sample; default takes the same number as supplied by the DATA parameter
NTIMES = *scalar*	Number of times to resample; default 100
SEED = *scalar*	Seed for random number generator; default continue from previous generation or use system clock

Parameters

DATA = *variates* or *factors*	The characteristics of the units to be clustered
TEST = *string tokens*	Test type, defining how each DATA variate or factor is treated in the calculation of the similarity between each unit (simplematching, jaccard, russellrao, dice, antidice, sneathsokal, rogerstanimoto, cityblock, manhattan, ecological, euclidean, pythagorean, minkowski, divergence, canberra, braycurtis, soergel); default * ignores that variate or factor
RANGE = *scalars*	Range of possible values of each DATA variate or factor; if omitted, the observed range is taken

Description

HBOOTSTRAP uses bootstrapping to assess the reliability of clusters formed in a hierarchical cluster analysis. The characteristics of the units to be clustered are described in a list of variates and factors, specified by the DATA parameter. The TEST parameter defines how each one is to be used when calculating similarities, and the RANGE parameter can specify ranges of their values. These operate as in the FSIMILARITY directive, which is used to form the similarity matrix for each cluster analysis. The MINKOWSKI option specifies the index *t* for the Minkowski tests.

For each bootstrap sample, a set of vectors is formed by sampling with replacement from the DATA vectors. The NDATASAMPLE option specifies the number of vectors to take; by default this is the same as the number of vectors supplied by DATA. The NTIMES option specifies the number of bootstrap samples; default 100. The SEED option specifies the seed to use for the random

numbers used to select the sample; the default of zero continues an existing sequence of random numbers or, if none, it initializes the sequence using the system clock. HBOOTSTRAP does a cluster analysis with those vectors using the HCLUSTER directive, and obtains the clusters that it forms using the HFCLUSTERS procedure. The CLIMIT option can be used to specify a limit, below which any clusters will be excluded.

The CLUSTERS option can supply a pointer containing a list of clusters whose reliability is to be assessed. This would usually have been obtained previously, from a cluster analysis performed with all the DATA vectors. Alternatively, if CLUSTERS is set to a pointer whose number of values has not been defined, or to an undeclared data structure, this will be defined as a pointer containing one of every cluster that has occurred during the bootstrapping. Each cluster is represented as a variate, containing the number of each unit in that cluster. (This number corresponds to the location of that unit in the DATA vectors.)

The REPLICATION option can save a variate containing the number of times each cluster has occurred during the bootstrapping. These replications can be used by the DCLUSTERLABELS procedure to label the clusters on a dendrogram.

The clusters and their replications can be printed by setting option PRINT=clusters. The UNITS option can be set to a text or a variate, to provide textual labels or other numbers to use for the units of the clusters, instead of the numbers in the CLUSTERS variates. The other PRINT setting, dendrogram, prints the dendrogram of the cluster analysis from each bootstrap sample.

Options: PRINT, METHOD, CLIMIT, UNITS, MINKOWSKI, CLUSTERS, REPLICATION, NDATASAMPLE, NTIMES, SEED.
Parameters: DATA, TEST, RANGE.

Action with RESTRICT
The DATA variates and factors must not be restricted.

See also
Directive: HCLUSTER.
Procedures: BOOTSTRAP, DCLUSTERLABELS, HFCLUSTERS, HPCLUSTERS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

HCOMPAREGROUPINGS

Compares groupings generated, for example, from cluster analyses (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (indexes, tests); default inde
PLOT = *string*	What to plot (histogram); default *
METHOD = *string tokens*	Which indexes to calculate (arand, jaccard, rand); default arand
NTIMES = *scalar*	Number of permutations to make for the tests; default 999

Parameters

FIRSTGROUPING = *factors*	First set of groupings
SECONDGROUPING = *factors*	Second set of groupings
ESTIMATES = *pointers*	Saves the values of the indexes calculated from the original data set
SEED = *scalars*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically
PERMUTATIONESTIMATES = *pointer*s	Saves the values of the indexes calculated from the permuted data sets

Description

HCOMPAREGROUPINGS calculates indexes to assess the similarity between two sets of groupings, which are specified in factors using the FIRSTGROUPING and SECONDGROUPING parameters. These may, for example, have been obtained from two different cluster analyses.

The METHOD option selects the indexes, with settings:

arand	adjusted Rand index,
jaccard	Jaccard index, and
rand	Rand index.

Details are given in the *Method* section. The default is to calculate only the adjusted Rand index.

The ESTIMATES parameter can save a pointer, containing a scalar for each index, to save the calculated values. The elements of the pointer are labelled by the index names, but defined so that you can refer to them in either lower- or upper-case or a mixture.

The PRINT option controls the printed output, with settings:

indexes	prints the indexes, and
tests	prints probabilities obtained from random permutation tests.

The random permutation tests allow you to assess whether the similarity may have arisen only by chance. The NTIMES option specifies the number of permutations to take (default 999). HCOMPAREGROUPINGS checks whether NTIMES is greater than the number of possible permutations available for the data set. If so, it does an exact test instead, which uses each possible permutation once. The SEED option specifies the seed that is used to obtain the random numbers used to form the permutations.

The PERMUTATIONESTIMATES parameter can save a pointer, containing a variate for each index, to save the values calculated in the random permutations. The elements of the pointer are labelled by the index names, but defined so that you can refer to them in either lower- or upper-case or a mixture.

You can set option PLOT=histogram to plot histograms showing where the calculated value

of each index lies within those obtained from the permutation tests.

Options: PRINT, PLOT, METHOD, NTIMES.
Parameters: FIRSTGROUPING, SECONDGROUPING, ESTIMATES, SEED, PERMUTATIONESTIMATES.

Method
The Rand index (Rand 1971) is defined as

$$(np_1 + np_2) / {}^{N}C_2$$

where

np_1 is the number of pairs of units that are in the same group in both factors,
np_2 is the number of pairs of units that are in different groups in both factors,
N is the total number of units, and
${}^{N}C_2$ is the total number of ways of selecting of 2 units from a sample of N units, which can be calculated as $N\times(N-1)/2$.

This ranges from zero (for no similarity) to one (for complete similarity).

The adjusted Rand index of Hubert & Arabie (1985) is defined as

$$\{ \sum_i \sum_j ({}^{mij}C_2) \} - \{ \sum_i ({}^{ai}C_2) \times \sum_j ({}^{bj}C_2) / ({}^{N}C_2) \} /$$
$$- \{ \sum_i ({}^{ai}C_2) + \sum_j ({}^{bj}C_2) \} - \{ \sum_i ({}^{ai}C_2) \times \sum_j ({}^{bj}C_2) / ({}^{N}C_2) \}$$

where

m_{ij} is the number of units that are in group i for the first factor, and group j for the second factor,
a_i is the number of units in group i of the first factor, and
b_j is the number of units in group j of the second factor.

The first term in the numerator measures the agreement between the groupings. The second term is the expected value of the first term, assuming a generalized hypergeometric distribution, and the first term of the denominator is its maximum value. The index has a value of zero if the groupings are independent, and one if they are in complete agreement.

The Jaccard index is defined as

$$np_1 / ({}^{N}C_2 - np_2)$$

This is similar to the Rand index, except that it excludes the pairs of units that are in different groups in both factors.

Action with RESTRICT
There must be no restrictions.

See also
Directives: CLUSTER, FACTOR, HCLUSTER.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis.

HEATUNITS

Calculates accumulated heat units of a temperature dependent process (R.J. Reader, R.A. Sutherland & K. Phelps).

Options

METHOD = *string token*	Temperature/time relationship to be used (sawtooth, cosine, linsine, expsine); default sawt
LATITUDE = *scalar*	Latitude at which temperatures were measured; default 52.205 N {Wellesbourne, U.K.}
RATE = *variate*	Value of rate relationship at cardinal temperatures
TEMPERATURE = *variate*	Cardinal temperatures
PARAMETERS = *variate*	Parameters *a*, *b*, *c* (*a*, *c* in hours) for the expsine method

Parameters

MINTEMPERATURE = *variates*	Minimum temperature on each day
MAXTEMPERATURE = *variates*	Maximum temperature on each day
FIRSTDAY = *scalars*	Day of year of first temperature recorded
HEATUNITS = *variates*	Development on each day

Description

HEATUNITS calculates heat units accumulated each day by a process whose rate depends on temperature. The temperature is assumed to vary diurnally. The rate function is defined as a linear spline so that any relationship can be approximated by specifying a set of cardinal temperatures and corresponding rates.

The METHOD option specifies the form of the diurnal temperature variation; this is derived from consecutive daily maximum and minimum temperatures according to methods compared by Reicosky *et al.* (1989). The LATITUDE option should be set to the latitude (degrees) at which the maxima and minima were recorded (positive for the northern hemisphere and negative for the southern hemisphere). The RATE and TEMPERATURE options define the rate/temperature relationship. They specify variates of equal length, RATE containing the rate of the process at the temperature of the corresponding unit of TEMPERATURE. The PARAMETERS option is a variate containing the values of the parameters *a*, *b* and *c* of the METHOD expsine.

The parameters MAXTEMP and MINTEMP contain the maximum and minimum temperatures on each day respectively. The FIRSTDAY parameter specifies the day of the year of the first unit of the MAXTEMP and MINTEMP variates. The HEATUNITS parameter returns the heat units accumulated on each day.

Options: METHOD, LATITUDE, RATE, TEMPERATURE, PARAMETERS.
Parameters: MINTEMPERATURE, MAXTEMPERATURE, FIRSTDAY, HEATUNITS.

Method

The integral of each segment of the rate/temperature relationship on each day is evaluated. These integrals are then added together. Further details are given by Reader & Phelps (1991).

Action with RESTRICT

None of the options or parameters of this procedure should be restricted as the maximum and minimum temperatures must be from consecutive days. Also they should not contain missing values, except for the first minimum and final maximum which are not used.

References

Reicosky, D.C., Winkelman, L.J., Baker, J.M. & Baker, D.G. (1989). Accuracy of hourly air temperatures calculated from daily minima and maxima. *Agricultural and Forest Meteorology*, **46**, 193-209.

Reader, R.J. & Phelps, K. (1991). Modelling the development of temperature-dependent processes. *Genstat Newsletter*, **28**, 27-32.

See also

Procedure: `DAYLENGTH`.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

HFAMALGAMATIONS

Forms an amalgamations matrix from a minimum spanning tree (R.W. Payne).

No options

Parameters

TREE = *matrices*	Minimum spanning tree
AMALGAMATIONS = *matrices*	Saves the amalgamation matrices formed from the minimum spanning trees

Description

Amalgamation matrices can be formed by HCLUSTER for all methods except single linkage. These can then be used, for example, by the HFCLUSTERS procedure to provide the set of clusters formed during the cluster analyses.

Information about a single-linkage cluster analysis can be saved, as a minimum spanning tree, by the HDISPLAY directive. HFAMALGAMATIONS can then use this to form an amalgamations matrix. The minimum spanning tree is supplied by the TREE parameter, and the amalgamations matrix is saved by the AMALGAMATIONS parameter.

Options: none.
Parameters: TREE, AMALGAMATIONS.

See also

Directives: HCLUSTER. HDISPLAY.
Procedures: HFCLUSTERS, HPCLUSTERS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

HFCLUSTERS

Forms a set of clusters from an amalgamations matrix (R.W. Payne).

Options

CLIMIT = *scalar*	Similarity value below which clusters are not formed; default 0
ORDERING = *string token*	How to order the clusters (join, lexicographic); default lexi
NCLUSTERS = *scalar*	Saves the number of clusters that have been formed

Parameters

AMALGAMATIONS = *matrices*	Amalgamation matrices
CLUSTERS = *pointers*	Saves the clusters
SIMILARITIES = *variates*	Saves the similarity values at which the clusters are formed

Description

HFCLUSTERS can form a set of clusters for use in bootstrapping e.g. by HBOOTSTRAP, or for labelling in a dendrogram by DCLUSTERLABELS.

The information required to form the clusters is supplied, in an amalgamation matrix, by the AMALGAMATIONS parameter. This can be formed by the HCLUSTER directive for all methods except single linkage. With single-linkage cluster analysis, it can be can be formed by the HFAMALGAMATIONS procedure, using a minimum spanning tree formed by the HDISPLAY directive.

The clusters are saved, in a pointer, by the CLUSTERS parameter. Each one is saved in a variate containing the numbers of the units in that cluster. (These numbers are the row or column positions of those units in the similarity matrix used by HCLUSTER.) By default, the clusters in the pointer are sorted into lexicographic order. This puts the clusters first into increasing order of size. Then, within each size, they are arranged in an order that would correspond to alphabetic order, if the units in the clusters were represented by the letters a-z. Alternatively, you can save the clusters in the order in which they are joined (i.e. in decreasing order of the similarity at which they join) by setting option ORDERING=similarity. Their similarities can be saved, in a variate, by the SIMILARITIES variate. The clusters can be printed by the HPCLUSTERS procedure.

The CLIMIT option can specify a limit on the similarity of the clusters that are saved. Clusters that join at a similarity value less than this are excluded. The NCLUSTERS option saves the number of clusters that are saved.

Options: CLIMIT, ORDERING, NCLUSTERS.
Parameters: AMALGAMATIONS, CLUSTERS, NCLUSTERS.

See also

Directives: HCLUSTER. HDISPLAY.
Procedures: DCLUSTERLABELS, HBOOTSTRAP, HFAMALGAMATIONS, HPCLUSTERS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

HGANALYSE

Analyses data using a hierarchical or double hierarchical generalized linear model (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

PRINT = *string tokens*	Controls printed output (model, fixedestimates, randomestimates, dispersionestimates, likelihoodstatistics, deviance, waldtests, fittedvalues, monitoring, dhgmonitoring); default mode, fixe, disp, devi, like, moni
LMETHOD = *string token*	Whether to use exact likelihood or extended quasi likelihood to obtain the y-variate and weights for the dispersion model (exact, eql); default exac
SEMETHOD = *string token*	Method to use to calculate the se's for the dispersion estimates (approximate, profilelikelihood); default appr
DMETHOD = *string token*	Method to use for the adjusted profile likelihood when calculating the likelihood statistics (automatic, choleski, lrv); default auto
EMETHOD = *string token*	Extrapolation method to use (aitken, adjustedaitken); default aitk
MLAPLACEORDER = *scalar*	Order of Laplace approximation to use in the estimation of the mean model (0 or 1); default 0
DLAPLACEORDER = *scalar*	Order of Laplace approximation to use in the estimation of the dispersion components (0, 1 or 2); default 0
MAXCYCLE = *scalars*	Maximum number of iterations of the hierarchical generalized linear model fits, and maximum number of iterations in the fitting of the mean and dispersion models; default 99,50
EXIT = *scalar*	Exit status (0 for success, 1 for failure to converge)
TOLERANCE = *scalar*	Criterion for convergence; default 0.0005
ETOLERANCE = *scalar*	Maximum size of ratio of the original to the new estimates allowed in Aitken extrapolation; default 7.5
GROUPTERM = *formula*	Random term to use as groups when fitting the augmented mean model; default * i.e. none

Parameters

Y = *variate*	Response variate (must be one only)
NBINOMIAL = *variate* or *scalar*	Total numbers for binomial data
RESIDUALS = *variate*	Saves the residuals
FITTEDVALUES = *variate*	Saves the fitted values
SAVE = *pointer*	Saves details of the analysis for use in subsequent HGDISPLAY, HGKEEP, HGPLOT or HGPREDICT statements

Description

HGANALYSE is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). These models extend generalized linear models (GLMs) to include additional random terms in the linear predictor. They include generalized linear mixed models (GLMMs) as a special case, but

do not constrain the additional terms to follow a Normal distribution and to have an identity link (as in the GLMM). For example, if the basic generalized linear model is a log-linear model (Poisson distribution and log link), a more appropriate assumption for the additional random terms might be a gamma distribution and a log link.

The analysis involves fitting an augmented generalized linear model to describe the mean of the distribution. This has units corresponding to the original data units, together with additional units for the effects of the random terms; see Lee & Nelder (1996). Then there are further GLMs to describe the dispersion for each random term (including the residual dispersion, phi); see Lee & Nelder (2001). In a DHGLM, some of these dispersion GLMs are themselves extended to become HGLMs by the inclusion of random terms; see Lee & Nelder (2006).

Before calling HGANALYSE, the fixed and random terms in the HGLM must be defined by the HGFIXEDMODEL and HGRANDOMMODEL procedures, respectively. The HGDRANDOMMODEL procedure can then add random terms to a dispersion GLM, so that the model becomes a DHGLM.

The variate to be analysed must be supplied by the Y parameter and, if the y-values are binomial responses, the NBINOMIAL parameter should supply the corresponding total numbers. Residuals and fitted values can be saved using the RESIDUALS and FITTEDVALUES parameters, respectively. Note that only one y-variate can be analysed at once, so any additional variates are ignored (as occurs with the MODEL directive when generalized linear models are defined).

The SAVE parameter allows you to save a pointer containing full details of the analysis. This can then be used to generate further output from HGDISPLAY, HGKEEP, HGPLOT or HGPREDICT. The most recent save structure is kept automatically inside Genstat to use as a default for the SAVE options of HGDISPLAY, HGKEEP, HGPLOT and HGPREDICT. So, you need save the pointer explicitly only if you want to display output from more than one analysis at a time.

The PRINT, SEMETHOD and DMETHOD options control printed output, almost exactly as in the HGDISPLAY procedure (which is called by HGANALYSE to produce the output). The only difference is that PRINT has additional settings: monitoring provides information about the fitting process of an ordinary HGLM, and dhgmonitoring provides information about the fitting of the HGLM for the dispersion model in a DHGLM.

The other options control various aspects of the fitting process. The fitting process involves alternative fits of the augmented GLM for the mean given the current estimates of the dispersion parameters, and of the models that estimate the dispersion parameters. The convergence of the process is assessed by comparing the dispersion estimates from successive fits. The MAXCYCLE option can specify two scalars. The first sets a limit on the number of alternating fits (default 99), and the second controls the number of iterations in the estimation of the mean model and of the dispersion model (default 50). The TOLERANCE option defines the criterion for convergence in the alternating fits (default 0.005). The EMETHOD option determines whether Aitken (default) or adjusted Aitken extrapolation is used in the estimation of the dispersion estimates, or you can set EMETHOD=* to use neither. The ETOLERANCE option sets an upper limit on the ratio of the changed value to the original values in the extrapolations; the default value is 7.5. The GROUPTERM option allows you to specify a random term whose factor combinations should be used as a groups factor during the fitting of the augmented mean model (see the GROUPS option of the MODEL directive). This allows models with large numbers of random effects to be fitted much more efficiently. However, algorithmic complications mean that predictions can then be made by HGPREDICT only using a BLUP for a specific random effect of that term – you cannot form predictions at the expected value of the term. The EXIT option can be set to a scalar which will be set to zero or one according to whether or not the fitting has been successful.

By default HGANALYSE uses exact likelihood to obtain the y-variate and weights for the dispersion model. This produces estimates with less bias than the previous method, of extended quasi likelihood (EQL). However, option LMETHOD is provided to enable EQL estimates to be obtained if required. For some of the models the DLAPLACEORDER option allows the order of

Laplace approximation involved in the estimation of the dispersion components to be increased from the standard value (and default) of 0, to either 1 or 2. This is appropriate for generalized linear mixed models with the binomial or Poisson distributions, where use of Laplace order 0 can lead to serious downwards bias. The MLAPLACEORDER option similarly allows you to set the order of Laplace approximation to use in the estimation of the mean model to 1 instead of 0.

Options: PRINT, LMETHOD, SEMETHOD, DMETHOD, EMETHOD, MLAPLACEORDER, DLAPLACEORDER, MAXCYCLE, EXIT, TOLERANCE, ETOLERANCE, GROUPTERM.
Parameters: Y, NBINOMIAL, RESIDUALS, FITTEDVALUES, SAVE.

Method
The model is fitted using the method of Lee & Nelder (2006).

Action with RESTRICT
Restrictions are not allowed.

References
Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.
Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.
Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.
Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also
Procedures: GEE, GLMM, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.
Genstat Reference Manual 1 Summary section on: Regression analysis.

HGDISPLAY

Displays results from a hierarchical or double hierarchical generalized linear model analysis (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

PRINT = *string tokens*	Controls printed output (model, fixedestimates, randomestimates, dispersionestimates, likelihoodstatistics, deviance, waldtests, fittedvalues); default *
SEMETHOD = *string token*	Method to use to calculate the se's for the dispersion estimates (approximate, profilelikelihood); default appr
DMETHOD = *string token*	Method to use for the adjusted profile likelihood when calculating the likelihood statistics (automatic, choleski, lrv); default auto
DISPERSIONTERM = *formula*	Model term for output from a dispersion analysis
SAVE = *pointer*	Save structure (from HGANALYSE) to provide details of the analysis; if omitted, output is from the most recent analysis

No parameters

Description

HGDISPLAY is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGDISPLAY allows you to display further output from the analysis.

The PRINT option specifies what output is required, with settings:

model	details of the model that has been fitted;
fixedestimates	estimates of the fixed effects in the HGLM;
randomestimates	estimates of the random effects in the HGLM;
dispersionestimates	estimates of the parameters in the dispersion models;
likelihoodstatistics	likelihood statistics for assessing the models;
deviance	scaled deviances for assessing goodness of fit;
waldtests	Wald tests of the terms that can be dropped from the fixed model (see HGWALD);
fittedvalues	table with unit number, response variable, fitted values, residuals and leverages.

The SEMETHOD option specifies which method to use to calculate standard errors for the estimated parameters of the dispersion models. The default, approximate, method is efficient to compute, but it may show downwards bias. However, the alternative profilelikelihood method can be very time-consuming. The DMETHOD option controls the method used to calculate the adjusted profile likelihood during the calculation of the likelihood statistics. The choleski method is fastest, while the lrv method provides a more robust alternative to use if choleski fails. The default setting, automatic, tries choleski first and then, if that fails, uses lrv instead.

By default the output is from the analysis of the mean model, but you can set the DISPERSIONTERM option to a formula defining one of the random terms to obtain information from the analysis to model its dispersion parameter.

Options: PRINT, SEMETHOD, DMETHOD, DISPERSIONTERM, SAVE.
Parameters: none.

Method
The output is mainly produced using RDISPLAY, RWALD and HGWALD.

References
Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.
Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.
Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.
Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also
Procedures: HGANALYSE, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.
Genstat Reference Manual 1 Summary section on: Regression analysis.

HGDRANDOMMODEL

Defines the random model in a hierarchical generalized linear model for the dispersion in a double hierarchical generalized linear model (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

DISTRIBUTION = *string token*	Distribution for the random model (beta, normal, gamma, inversegamma); default norm
LINK = *string token*	Link for the random model (identity, logarithm, logit, reciprocal); default iden
RANDOMTERM = *formula*	Random term whose dispersion is being modelled; if unset, the model is assumed to be for the residual dispersion parameter (phi)
PHIMETHOD = *string token*	Whether to fix or estimate the residual dispersion parameter in the dispersion HGLM (fix, estimate); default fix

Parameters

TERMS = *formula*	Random model
DLINK = *string tokens*	Link for the dispersion model for each random term (logarithm, reciprocal); default loga
DFORMULA = *formula structures*	Dispersion model for each random term; default * i.e. none
DOFFSET = *variates*	Offset variate for dispersion model for each random term; default * i.e. none
LMATRIX = *matrices*	Linear transformation to apply to design matrix **Z** of each random term, in order to define correlations between its effects; default * i.e. none
DDISPERSION = *scalar*	Dispersion parameter to use in the dispersion model for each random term; default 1
FDISPERSION = *scalar*	Fixed value for the dispersion parameter of each random term; default !s(*) i.e. dispersion is estimated

Description

HGDRANDOMMODEL allows you to extend a hierarchical generalized linear model (HGLM) to become a double hierarchical generalized linear model (DHGLM); see Lee & Nelder (1996, 2001a, 2006) or Lee, Nelder & Pawitan (2006). This is done by adding some random terms to one of the generalized linear models that is to model the dispersion, so that this becomes an HGLM. By default the residual dispersion of this HGLM is fixed, but you can set option PHIMETHOD=estimate to estimate it. The random term whose dispersion is to be modelled by the HGLM is indicated by the RANDOMTERM option. If RANDOMTERM is omitted, the dispersion model is assumed to be for the residual dispersion parameter (phi) of the original HGLM.

The TERMS parameter defines the additional random terms, and the LINK and DISTRIBUTION options specify their distribution and link function respectively. You can specify a generalized linear model (GLM) to model the dispersion parameter for any of these additional random terms by specifying a Genstat formula structure, containing the (fixed) terms to be fitted in the GLM, using the DFORMULA parameter (which runs in parallel with the list of random terms supplied by the TERMS parameter). The DLINK parameter specifies the link to use with each dispersion model, the DOFFSET parameter allows you to specify an offset variate, and the DDISPERSION parameter defines the dispersion parameter for the dispersion GLM (default 1). Alternatively, if you do not define a dispersion model for a random term, you can use the FDISPERSION parameter to fix its dispersion at a specific value.

The LMATRIX parameter allows correlation structures to be defined for random terms, using the method described by Lee & Nelder (2001b). This is done by setting LMATRIX to a matrix **L** that is used as a post-multiplier for the **Z** matrix of the random term concerned. Lee & Nelder (2001b) give examples illustrating the types of model that can be defined.

Options: DISTRIBUTION, LINK, RANDOMTERM, PHIMETHOD.
Parameters: TERMS, DLINK, DFORMULA, DOFFSET, LMATRIX, DDISPERSION, FDISPERSION.

Method
The information is stored in a workspace G5PL_HG (accessed using the WORKSPACE directive) for later use by HGANALYSE.

References
Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001a). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2001b). Modelling and analysing correlated non-normal data. *Statistical Modelling*, **1**, 3-16.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also
Procedures: HGANALYSE, HGDISPLAY, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.
Genstat Reference Manual 1 Summary section on: Regression analysis.

HGFIXEDMODEL

Defines the fixed model for a hierarchical or double hierarchical generalized linear model (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

DISTRIBUTION = *string token*	Distribution of the data (binomial, poisson, normal, gamma); default norm
LINK = *string token*	Link for the fixed model (identity, logarithm, logit, reciprocal, probit, complementaryloglog); default iden
DISPERSION = *scalar*	Value of dispersion parameter in calculation of s.e.s etc; default * for DIST=norm or gamm, and 1 for DIST=pois or bino
DLINK = *string token*	Link for the dispersion model (logarithm, reciprocal); default loga
DTERMS = *formula*	Dispersion model; default * i.e. none
CONSTANT = *string token*	How to treat the constant (estimate, omit) default esti
FACTORIAL = *scalar*	Limit on number of variates and/or factors in a fixed model term; default 3
WEIGHTS = *variate*	Prior weights; default * i.e. 1
OFFSET = *variate*	Offset variate; default * i.e. none
DOFFSET = *variate*	Offset variate for dispersion model; default * i.e. none
DDISPERSION = *scalar*	Dispersion parameter to use in a dispersion model for the residual dispersion parameter phi; default 1
IDISPERSION = *scalar*	Initial value for the residual dispersion parameter phi; default * i.e. formed automatically

Parameter

TERMS = *formula*	Fixed model

Description

HGFIXEDMODEL is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical generalized linear models defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). These models extend generalized linear models (GLMs) to include additional random terms in the linear predictor. They include generalized linear mixed models (GLMMs) as a special case, but do not constrain the additional terms to follow a Normal distribution and to have an identity link (as in the GLMM). For example, if the basic generalized linear model is a log-linear model (Poisson distribution and log link), a more appropriate assumption or the additional random terms might be a gamma distribution and a log link.

The role of HGFIXEDMODEL is to specify the fixed model terms in the HGLM, and to define the distribution of the data (this corresponds to error distribution of a GLM). The fixed model is given by the TERMS parameter. Most of the options operate similarly to those occurring in the directives FIT and MODEL. The link function for the fixed model is defined by the LINK option, and the FACTORIAL option sets a limit on the number of variates and/or factors for a term to be included in the fixed model (default 3). The CONSTANT option indicates whether or not to include a constant term or intercept (by default this is included), and the OFFSET option allows an offset variate to be included. The DISTRIBUTION option defines the distribution of the data, the WEIGHTS option allows you to specify a variate of prior weights, and the DISPERSION option governs how the dispersion parameter is obtained.

The HGLM methodology also caters for structured dispersion models, in which fixed terms are included in the generalized linear models that are used to estimate the dispersion parameters. Currently these GLMs must have a gamma distribution. The DTERMS option allows you to specify fixed terms for the GLM that estimates the residual dispersion parameter phi. The DLINK parameter specifies the link to use with the dispersion model, the DOFFSET option allows you to specify an offset variate, and the DDISPERSION option defines the dispersion parameter for the dispersion GLM (default 1). You can also extend the GLM to become an HGLM (thus making the full model a *double hierarchical generalized linear model* or *DHGLM*), by using the HGDRANDOMMODEL procedure to add some random terms.

The IDISPERSION option allows you to define an initial value for the residual dispersion parameter phi. Initial values for the dispersion parameters of the additional random terms of the HGLM can be defined using the IDISPERSION parameter of the HGRANDOMMODEL procedure. If you set both of these, the HGANALYSE procedure will then use them to initialize the weights that are involved in the fitting of the augmented mean model; for details see Chapter 6 of Lee, Nelder & Pawitan (2006). The default weights that are formed automatically if either of these is unset are satisfactory in most circumstances, but you may want to try your own initial values if you encounter convergemce problems.

Options: DISTRIBUTION, LINK, DISPERSION, DLINK, DTERMS, CONSTANT, FACTORIAL, WEIGHTS, OFFSET, DOFFSET, DDISPERSION, IDISPERSION.
Parameter: TERMS.

Method

The information is stored in a workspace G5PL_HG (accessed using the WORKSPACE directive) for later use by HGANALYSE.

References

Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman & Hall, London.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.
Genstat Reference Manual 1 Summary section on: Regression analysis.

HGFTEST

Calculates likelihood tests for fixed terms in a hierarchical generalized linear model (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

PRINT = *string token*	Controls printed output (tests); default test
FACTORIAL = *scalar*	Limit on number of factors in the model terms generated from the TERMS parameter
LMETHOD = *string token*	Whether to use exact likelihood or extended quasi likelihood to obtain the y-variate and weights for the dispersion model (exact, eql); default is to use the same setting as in the original analysis
DMETHOD = *string token*	Method to use for the adjusted profile likelihood when calculating the likelihood statistics (automatic, choleski, lrv); default auto
EMETHOD = *string token*	Extrapolation method to use (aitken, adjustedaitken); default is to use the same setting as in the original analysis
MLAPLACEORDER = *scalar*	Order of Laplace approximation to use in the estimation of the mean model (0 or 1); default is to use the same setting as in the original analysis
DLAPLACEORDER = *scalar*	Order of Laplace approximation to use in the estimation of the dispersion components (0, 1 or 2); default is to use the same setting as in the original analysis
MAXCYCLE = *scalars*	Maximum number of iterations of the hierarchical generalized linear model fits, and maximum number of iterations in the fitting of the mean and dispersion models; default 99,50
EXIT = *scalar*	Exit status (0 for success, 1 for failure to converge with any of the fixed terms)
TOLERANCE = *scalar*	Criterion for convergence; default is to use the same setting as in the original analysis
ETOLERANCE = *scalar*	Maximum size of ratio of the original to the new estimates allowed in Aitken extrapolation; default is to use the same setting as in the original analysis
SAVE = *pointer*	Save structure from the original analysis

Parameters

TERMS = *formula*	Terms to test
TESTSTATISTIC = *pointer* or *scalar*	Saves the test statistics
DF = *pointer* or *scalar*	Saves the degrees of freedom

Description

HGFTEST is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGFTEST allows you to print or save likelihood tests for terms that can be dropped from the fixed model of a hierarchical generalized linear model.

By default, HGFTEST produces tests for all the fixed terms that can be dropped: that is, for every term that is not marginal to another term in the fixed model. For example, in the formula

```
A + B + C + D + A.B + A.D + B.D
```

the terms C, A.B, A.D and B.D can be dropped as there are no other terms in the model that contain all their factors (i.e. none to which thay are marginal). However, A cannot be dropped until A.B and A.D have been dropped. You can use the TERMS parameter to request tests for a specific set of terms, but a missing value is given for any term that cannot be dropped. The FACTORIAL option sets a limit on the number of factors in each term that is formed from the TERMS formula (default 3).

The TESTSTATISTIC parameter can save the statistics, and the DF parameter can save their numbers of degrees of freedom. If you are making a test for a single term, you can supply a scalar for each of these parameters. However, if you have several terms, you must supply a pointer which will then be set up to contain as many scalars as there are terms.

The tests are made by calculating the change in the profile likelihood $P_v(h)$ as the term concerned is dropped from the fixed model. The LMETHOD, DMETHOD, EMETHOD, MLAPLACEORDER, DLAPLACEORDER, MAXCYCLE, TOLERANCE and ETOLERANCE, options control how the fitting is done, and the likelihood is calculated. These all operate exactly as in the HGANALYSE procedure. The default for DMETHOD is automatic, and the default for MAXCYCLE= is 99,50. For the other options the defaults are to use the same settings as in the HGANALYSE command that performed the original analysis.

By default, the terms are dropped from the most recent HGLM analysis, but you can use the SAVE option to supply the save structure from some earlier analysis.

Options: PRINT, FACTORIAL, LMETHOD, DMETHOD, EMETHOD, MLAPLACEORDER, DLAPLACEORDER, MAXCYCLE, EXIT, TOLERANCE, ETOLERANCE, SAVE.
Parameters: TERMS, TESTSTATISTIC, DF.

References

Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HGGRAPH

Draws a graph to display the fit of an HGLM or DHGLM analysis (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

GRAPHICS = *string token*	Type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	Title for the graph; default * sets an appropriate title automatically
WINDOW = *number*	Which high-resolution graphics window to use; default 4 (redefined if necessary to fill the frame)
SCREEN = *string token*	Whether to clear the graphics screen before plotting (clear, keep); default clea
BACKTRANSFORM = *string token*	What back-transformation to make (link, none, axis); default none
OMITRESPONSE = *string token*	Whether to omit the adjusted response values (no, yes); default no
SAVE = *pointer*	Specifies the save structure (from HGANALYSE) of the analysis from which to predict; default uses the most recent analysis

Parameters

INDEX = *variates* or *factors*	Which variate or factor to display along the x-axis; default * if GROUPS is set, otherwise INDEX is set to the first variate in the fixed model
GROUPS = *factors*	Factor to define groups of points to display; default * if INDEX is set, otherwise GROUPS is set to the first factor in the fixed model

Description

HGGRAPH is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGGRAPH has a similar role to the RGRAPH procedure in ordinary regression and generalized linear models. It displays the fitted model in one or two dimensions. It usually also displays the observed response values, adjusted for any other explanatory terms in the model, but these can be omitted by setting option OMITRESPONSE=yes.

The dimensions to display are specified by the INDEX and GROUPS parameters. The INDEX vector, which can be either a variate or a factor from the fixed model of the HGLM, defines the x-axis of the plot. (The y-axis corresponds to the response scale.) The GROUPS parameter can be set to another factor from the fixed model. A set of points is then plotted for each level of GROUPS, so that you can study the interaction between GROUPS and INDEX. If INDEX and GROUPS are not set, HGGRAPH takes the first variate (if any) and the first factor in the fixed model.

The relationship is usually plotted on the scale of the linear predictor. However, with a conjugate HGLM, you can set option BACKTRANSFORM=link to use the original scale of the response. Alternatively, you can set BACKTRANSFORM=axis to include axis markings, back-transformed onto the natural scale, on the right-hand side of the y-axis. However, this is not available for the reciprocal link.

The TITLE option can be used to supply a title for the graph. By default the graph is plotted on the current high-resolution device, but the GRAPHICS option can be set to line for a line printer plot. The WINDOW option can be used to select a pre-defined window for high-resolution plots; otherwise window 4 is used, and is redefined if necessary to fill the frame. The SCREEN option allows the graph to be added to an existing high-resolution plot. The colours and symbols used in the displays can be controlled by setting the attributes of the following pens with the PEN directive before calling the procedure:

pen 1	labels for lines when drawn for each level of a factor,
pen 2	fitted lines and means,
pen 3	points, and
pen 4	back-transformed axis marks and labels.

Options: GRAPHICS, TITLE, WINDOW, SCREEN, BACKTRANSFORM, OMITRESPONSE, SAVE.
Parameters: INDEX, GROUPS.

Method

HGGRAPH calculates the points using the HGPREDICT procedure.

References

Lee, Y. & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *J. R. Statist. Soc. B*, **58**, 619-678.

Lee, Y. & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HGKEEP

Saves information from a hierarchical or double hierarchical generalized linear model analysis (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

MODELTYPE = *string token*	Type of model from which to save information (mean, dispersion); default mean
RMETHOD = *string token*	Type of residuals to save using the RESIDUALS parameter (deviance, Pearson, simple); default devi
DMETHOD = *string token*	Method to use for the adjusted profile likelihood when calculating the likelihood statistics (automatic, choleski, lrv); default auto
IGNOREFAILURE = *string token*	Whether to save information even if the fitting of the HGLM failed to converge (yes, no); default no
SAVE = *pointer*	Save structure (from HGANALYSE) to provide details of the analysis; if omitted, information is saved from the most recent analysis

Parameters

RANDOMTERM = *formula*	Random model terms from whose analysis the information is to be saved
DHGRANDOMTERM = *formula*	Random model terms in a DHGLM from whose (HGLM) analysis the information is to be saved
RESIDUALS = *variates*	Residuals
FITTEDVALUES = *variates*	Fitted values
LEVERAGES = *variates*	Leverages
ESTIMATES = *variates*	Estimates of parameters
SE = *variates*	Standard errors of the estimates
VCOVARIANCE = *symmetric matrices*	Variance-covariance matrix of each set of estimates
DEVIANCE = *scalars* or *tables*	Scaled deviances (in a table) for a mean model, or residual deviance (in a scalar) for a dispersion model
DF = *scalars* or *tables*	Residual degrees of freedom
ITERATIVEWEIGHTS = *variates*	Iterative weights
LINEARPREDICTOR = *variates*	Linear predictors
YADJUSTED = *variates*	Adjusted responses
LIKELIHOODSTATISTICS = *variates*	Likelihood statistics
LDF = *variates*	Numbers of fixed and random parameters in the mean and dispersion models

Description

HGKEEP is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGKEEP allows you to copy information from the output into standard Genstat data structures.

The MODELTYPE option indicates the model (mean or dispersion) from which the information is to be saved; by default this is the model for the mean (i.e. the main HGLM). The

RANDOMTERM parameter specifies the random term from whose analysis the information is to be saved; if this is omitted the information is for the residual term (phi). If a DHGLM has been fitted, you can save information from the HGLM that is being used as a dispersion model by setting the DHGRANDOMTERM parameter to the random term concerned. The LIKELIHOODSTATISTICS parameter saves the likelihood statistics (as given by the likelihoodstatistics setting of the PRINT option of HGANALYSE and HGDISPLAY). The DMETHOD option controls the method used to calculate the adjusted profile likelihood during the calculation of the likelihood statistics. The choleski method is fastest, while the lrv method provides a more robust alternative to use if choleski fails. The default setting, automatic, tries choleski first and then, if that fails, uses lrv instead. The LDF parameter saves the numbers of fixed and random parameters in the mean and dispersion models. (These accompany the likelihood statistics in the output, and indicate the numbers of parameters represented by the various statistics.) The other parameters operate as in the RKEEP directive except that, for a mean model, DEVIANCE saves tables of scaled deviances and DF saves a table with the corresponding degrees of freedom. Similarly, as in the RKEEP directive, the RMETHOD option indicates the type of residual to form.

By default, HGKEEP will give a warning (and nothing will be saved) if the fitting of the HGLM failed to converge. Alternatively, you can set option IGNOREFAILURE=yes to save information from the final iteration.

Options: MODELTYPE, RMETHOD, DMETHOD, IGNOREFAILURE, SAVE.

Parameters: RANDOMTERM, DHGRANDOMTERM, RESIDUALS, FITTEDVALUES, LEVERAGES, ESTIMATES, SE, VCOVARIANCE, DEVIANCE, DF, ITERATIVEWEIGHTS, LINEARPREDICTOR, YADJUSTED, LIKELIHOODSTATISTICS.

Method

HGKEEP mainly uses the RKEEP directive.

References

Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HGNONLINEAR

Defines nonlinear parameters for the fixed model of a hierarchical generalized linear model (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

CALCULATION = *expression structures*	Calculation of explanatory variates involving nonlinear parameters
METHOD = *string token*	Algorithm for fitting the nonlinear model (GaussNewton, NewtonRaphson, FletcherPowell); default Gaus
VECTORS = *variates*	Vectors involved in the calculations (data vectors or factors or derived vectors that appear in the fixed model)

Parameters

PARAMETER = *scalars*	Nonlinear parameters in the model
LOWER = *scalars*	Lower bound for each parameter
UPPER = *scalars*	Upper bound for each parameter
STEPLENGTH = *scalars*	Initial step length for each parameter
INITIAL = *scalars*	Initial value for each parameter
DELTA = *scalars*	Parameter increment to use when calculating numerical derivatives

Description

HGNONLINEAR is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical generalized linear models defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). These models extend generalized linear models (GLMs) to include additional random terms in the linear predictor. They include generalized linear mixed models (GLMMs) as a special case, but do not constrain the additional terms to follow a Normal distribution and to have an identity link (as in the GLMM). For example, if the basic generalized linear model is a log-linear model (Poisson distribution and log link), a more appropriate assumption for the additional random terms might be a gamma distribution and a log link.

HGNONLINEAR allows you to extend a conjugate HGLM to become a hierarchical generalized nonlinear model by including nonlinear parameters in the fixed model (Payne 2014). This is done exactly as in a generalized nonlinear model (see *Guide to the Genstat Command Language*, Part 2 Section 3.5.8), by defining some calculations to form variates to include as linear terms in the model. So the nonlinear terms have the form

$$B \times \mathrm{f}(p)$$

where B is a (linear) regression coefficient and f() is a function of some nonlinear parameters e.g.

$$B \times R^X$$

defines an exponential term with nonlinear parameter R. (This can be written as $\exp(k \times X)$ where the $R = \exp(k)$.)

The calculations are specified, as a list of Genstat expression structures, by the CALCULATION option. (This corresponds to the CALCULATION option of the FIT directive.) You must also use the VECTORS option to list the vectors that appear in the calculations (either as data vectors or as derived vectors that then appear as linear terms in the fixed model). The METHOD option indicates which algorithm to use to fit the nonlinear model. (This corresponds to the METHOD option of the RCYCLE directive.)

The parameters of HGNONLINEAR supply information about the nonlinear parameters. Most of these correspond to parameters in the RCYCLE directive. PARAMETER lists the identifiers of

the parameters as they appear in the calculations. LOWER and UPPER can define lower and upper bounds. STEPLENGTH can define the step lengths to use for each parameter at the start of the optimization, and INITIAL can define initial values. Genstat will take default initial values if you do not specify these yourself. However, these may not lead to convergence, so you are strongly advised to specify your own. It is often feasible to fit the models in an ordinary generalized nonlinear model, with the random terms included as fixed terms, and then use those estimates as the initial values for the hierarchical generalized nonlinear model.

The final parameter, DELTA, specifies a small increment to each parameter to be used inside the algorithm when calculating derivatives of the fixed model with respect to each nonlinear parameter (needed to calculate leverages).

Options: CALCULATION, METHOD, VECTORS.
Parameters: PARAMETER, LOWER, UPPER, STEPLENGTH, INITIAL, DELTA.

Method
The information is stored in a workspace 'G5PL_HG' (accessed using the WORKSPACE directive) for later use by HGANALYSE.

References
Lee, Y. & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society Series B*, **58**, 619-678.

Lee, Y. & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 1-29.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

Payne, R.W. (2014). Hierarchical generalized nonlinear models. In: *Statistical Modelling in Biostatistics and Bioinformatics* (ed. G. MacKenzie & D. Peng), 111-124. Springer, New York.

See also
Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.
Genstat Reference Manual 1 Summary section on: Regression analysis.

HGPLOT

Produces model-checking plots for a hierarchical or double hierarchical generalized linear model analysis (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

MODELTYPE = *string token*	Type of model for which plots are required (mean, dispersion); default mean
RANDOMTERM = *formula*	Random term whose residuals are to be plotted; default * i.e. the residuals from the full model
DHGRANDOMTERM = *formula*	Random model term in a DHGLM whose residuals are to be plotted; default *
RMETHOD = *string token*	Type of residual to use (deviance, Pearson, simple); default devi
INDEX = *variate* or *factor*	X-values to use for an index plot; default !(1,2...)
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	Overall title for the plots; if unset, the identifier of the y-variate is used
SAVE = *pointer*	Specifies the analysis (by HGANALYSE) from which the residuals and fitted values are to be taken; by default they are taken from the most recent analysis

Parameters

METHOD = *string tokens*	Types of graph (up to four out of the six possible) to be plotted (histogram, fittedvalues, absresidual, normal, halfnormal, index); default hist, fitt, norm, absr
PEN = *scalars*, *variates* or *factors*	Pen(s) to use for each plot

Description

HGPLOT is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and discussed by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGPLOT displays plots of residuals to help with model checking.

Six types of plot are available. They are selected using the METHOD parameter with settings:

histogram	histogram of residuals;
fittedvalues	residuals versus fitted values;
absresidual	absolute values of residuals versus fitted values;
normal	Normal plot;
halfnormal	half-Normal plot; and
index	plot against an "index" variable (specified by the INDEX option).

Up to four can be examined in any call of the procedure. The PEN parameter can be used to specify the graphics pen or pens to use for each plot. The TITLE option can supply an overall title. If this is not set, the identifier of the y-variate is used.

The MODELTYPE option indicates the type of model for which the plots are required. The default setting mean requests plots from the mean model, and the alternative setting dispersion obtains plots from the dispersion model. The RANDOMTERM option specifies the

random term whose residuals are to be plotted; if this is omitted the plot is for the residual term (phi). If a DHGLM has been fitted, you can plot residuals from the HGLM that is being used as a dispersion model by setting the DHGRANDOMTERM parameter to the random term concerned. The type of residual to plot is specified by the RMETHOD option; by default these are deviance residuals.

By default, high-resolution graphics are used. Line-printer graphics can be used by setting option GRAPHICS=lineprinter.

Options: MODELTYPE, RANDOMTERM, DHGRANDOMTERM, RMETHOD, INDEX, GRAPHICS, TITLE, SAVE.
Parameters: METHOD, PEN.

Method

HGPLOT calls procedure DRESIDUALS to do the plotting.

References

Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HGPREDICT

Forms predictions from a hierarchical or double hierarchical generalized linear model analysis (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

PRINT = *string token*	What to print (description, predictions, se, sed, vcovariance); default desc, pred, se
COMBINATIONS = *string token*	Which combinations of factors in the current model to include (full, present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment (marginal, equal); default marg
WEIGHTS = *table*	Weights classified by some or all of the factors in the model; default *
OFFSET = *scalar*	Value of offset on which to base predictions; default mean of offset variate
METHOD = *string token*	Method of forming margin (mean, total); default mean
ALIASING = *string token*	How to deal with aliased parameters (fault, ignore); default faul
BACKTRANSFORM = *string token*	What back-transformation to apply to the values on the linear scale, before calculating the predicted means (link, none); default none
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, nonlinear); default *
NBINOMIAL = *scalar*	Supplies the total number of trials to be used for prediction with a binomial distribution (providing a value *n* greater than one allows predictions to be made of the number of "successes" out of *n*, whereas the value 1 predicts the proportion of successes); default 1
PREDICTIONS = *table* or *scalar*	To save the predictions; default *
SE = *table* or *scalar*	To save standard errors of predictions; default *
SED = *symmetric matrix*	To save matrices of standard errors of differences between predictions; default *
VCOVARIANCE = *symmetric matrix*	To save variance-covariance matrices of predictions; default *
SAVE = *pointer*	Specifies the save structure (from HGANALYSE) of the analysis from which to predict; default uses the most recent analysis

Parameters

CLASSIFY = *vectors*	Variates and/or factors to classify table of predictions
LEVELS = *variates* or *scalars*	To specify values of variates, levels of factors
NEWFACTOR = *identifiers*	Identifiers for new factors that are defined when LEVELS are specified

Description

HGPREDICT is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGPREDICT allows you to form predictions.

HGPREDICT uses the PREDICT directive internally. Its options and parameters are a subset of those of PREDICT, and are used in the same way except that back-transformations are possible

only with conjugate models. Consequently, the default for option BACKTRANSFORM is none.

The CLASSIFY list can contain factors from either the fixed or random models but you may specify only one level for each random factor. If all the factors in a particular random term are in the CLASSIFY list, the prediction will use the BLUP (best linear unbiased predictor) for the random effect of the term corresponding to the levels that are specified for its factors. Otherwise, provided that random term was not used as a group term in the analysis (see the GROUPTERM option of HGANALYSE), the predictions will be at the mean value of the random distribution of the term. Alternatively, if that random term was used as a group term, HGPREDICT will make the predictions using the smallest BLUP of the term.

Options: PRINT, COMBINATIONS, ADJUSTMENT, WEIGHTS, OFFSET, METHOD, ALIASING, BACKTRANSFORM, NOMESSAGE, NBINOMIAL, PREDICTIONS, SE, SED, VCOVARIANCE, SAVE.
Parameters: CLASSIFY, LEVELS, NEWFACTOR.

Method

HGPREDICT forms the predictions using the PREDICT directive.

References

Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGRANDOMMODEL, HGRTEST, HGSTATUS, HGWALD.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HGRANDOMMODEL

Defines the random model for a hierarchical or double hierarchical generalized linear model (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

DISTRIBUTION = *string token*	Distribution for the random model (beta, normal, gamma, inversegamma); default norm
LINK = *string token*	Link for the random model (identity, logarithm, logit, reciprocal); default iden

Parameters

TERMS = *formula*	Random model
DLINK = *string tokens*	Link for the dispersion model for each random term (logarithm, reciprocal); default loga
DFORMULA = *formula structures*	Dispersion model for each random term; default * i.e. none
DOFFSET = *variates*	Offset variate for dispersion model for each random term; default * i.e. none
LMATRIX = *matrices*	Linear transformation to apply to design matrix **Z** of each random term, in order to define correlations between its effects; default * i.e. none
DDISPERSION = *scalar*	Dispersion parameter to use in the dispersion model for each random term; default 1
FDISPERSION = *scalar*	Fixed value for the dispersion parameter of each random term; default !s(*) i.e. dispersion is estimated
IDISPERSION = *scalar*	Initial value for the dispersion parameter for each random term; default * i.e. formed automatically

Description

HGRANDOMMODEL is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical generalized linear models defined by Lee & Nelder (1996, 2001a, 2006) and described by Lee, Nelder & Pawitan (2006). These models extend generalized linear models (GLMs) to include additional random terms in the linear predictor. They include generalized linear mixed models (GLMMs) as a special case, but do not constrain the additional terms to follow a Normal distribution and to have an identity link (as in the GLMM). For example, if the basic generalized linear model is a log-linear model (Poisson distribution and log link), a more appropriate assumption or the additional random terms might be a gamma distribution and a log link.

The TERMS parameter defines the additional random terms. These should not include the final (residual) term, unless you want to define a saturated random model as, for example, in the use of a negative binomial distribution in the Fabric example, discussed in Lee, Nelder & Pawitan 2006, Section 6.6.3. The LINK and DISTRIBUTION options specify their distribution and link function respectively.

The HGLM methodology also caters for structured dispersion models, in which fixed terms are included in the generalized linear models that are used to estimate the dispersion parameters for the random terms of the HGLM. Currently these GLMs must have a gamma distribution. These fixed terms are specified in a Genstat formula structure using the DFORMULA parameter (which runs in parallel with the list of random terms supplied by the TERMS parameter). The DLINK parameter specifies the link to use with each dispersion model, the DOFFSET parameter allows you to specify an offset variate, and the DDISPERSION parameter defines the dispersion parameter for the dispersion GLM (default 1). You can also extend a dispersion GLM to become

an HGLM (thus making the full model a *double hierarchical generalized linear model* or *DHGLM*), by using the HGDRANDOMMODEL procedure to add some random terms.

Alternatively, if you do not define a dispersion model for a random term, you can use the FDISPERSION parameter to fix its dispersion at a specific value.

The LMATRIX parameter allows correlation structures to be defined for random terms, using the method described by Lee & Nelder (2001b). This is done by setting LMATRIX to a matrix **L** that is used as a post-multiplier for the **Z** matrix of the random term concerned. Lee & Nelder (2001b) give examples illustrating the types of model that can be defined.

The IDISPERSION parameter allows you to define initial values for the dispersion parameters of the random terms. An initial value for the residual dispersion parameter phi can be defined using the IDISPERSION option of the HGFIXEDMODEL procedure. If you set both of these, the HGANALYSE procedure will then use them to initialize the weights that are involved in the fitting of the augmented mean model; for details see Chapter 6 of Lee, Nelder & Pawitan (2006). The default weights that are formed automatically if either of these is unset are satisfactory in most circumstances, but you may want to try your own initial values if you encounter convergemce problems.

Options: DISTRIBUTION, LINK.
Parameters: TERMS, DLINK, DFORMULA, DOFFSET, LMATRIX, DDISPERSION, FDISPERSION, IDISPERSION.

Method

The information is stored in a workspace G5PL_HG (accessed using the WORKSPACE directive) for later use by HGANALYSE.

References

Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001a). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2001b). Modelling and analysing correlated non-normal data. *Statistical Modelling*, **1**, 3-16.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman & Hall, London.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRTEST, HGSTATUS, HGWALD.
Genstat Reference Manual 1 Summary section on: Regression analysis.

HGRTEST

Calculates likelihood tests for random terms in a hierarchical generalized linear model (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

PRINT = *string token*	Controls printed output (tests); default test
LMETHOD = *string token*	Whether to use exact likelihood or extended quasi likelihood to obtain the y-variate and weights for the dispersion model (exact, eql); default is to use the same setting as in the original analysis
DMETHOD = *string token*	Method to use for the adjusted profile likelihood when calculating the likelihood statistics (automatic, choleski, lrv); default auto
EMETHOD = *string token*	Extrapolation method to use (aitken, adjustedaitken); default is to use the same setting as in the original analysis
MLAPLACEORDER = *scalar*	Order of Laplace approximation to use in the estimation of the mean model (0 or 1); default is to use the same setting as in the original analysis
DLAPLACEORDER = *scalar*	Order of Laplace approximation to use in the estimation of the dispersion components (0, 1 or 2); default is to use the same setting as in the original analysis
MAXCYCLE = *scalars*	Maximum number of iterations of the hierarchical generalized linear model fits, and maximum number of iterations in the fitting of the mean and dispersion models; default 99,50
EXIT = *scalar*	Exit status (0 for success, 1 for failure to converge for any of the random terms)
TOLERANCE = *scalar*	Criterion for convergence; default is to use the same setting as in the original analysis
ETOLERANCE = *scalar*	Maximum size of ratio of the original to the new estimates allowed in Aitken extrapolation; default is to use the same setting as in the original analysis
GROUPTERM = *formula*	Random term to use as groups when fitting the augmented mean model; default is to use the same setting as in the original analysis
SAVE = *pointer*	Save structure from the original analysis

Parameters

TERMS = *formula*	Terms to test
TESTSTATISTIC = *pointer* or *scalar*	Saves the test statistics
DF = *pointer* or *scalar*	Saves the degrees of freedom

Description

HGRTEST is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGRTEST allows you to print or save likelihood tests for terms in the random model of a hierarchical generalized linear model.

By default, HGRTEST produces tests for every random term. However, you can use the TERMS parameter to request tests for a specific set of terms.

The TESTSTATISTIC parameter can save the statistics, and the DF parameter can save their numbers of degrees of freedom. If you are making a test for a single term, you can supply a scalar for each of these parameters. However, if you have several terms, you must supply a pointer which will then be set up to contain as many scalars as there are terms.

The tests are made by calculating the change in the profile likelihood $P_{\beta,v}(h)$ as the term concerned is dropped from the random model. So, HGRTEST needs to refit the model with the revised random model. The LMETHOD, DMETHOD, EMETHOD, MLAPLACEORDER, DLAPLACEORDER, MAXCYCLE, TOLERANCE, ETOLERANCE and GROUPTERM options control how the fitting is done, and the likelihood is calculated. These all operate exactly as in the HGANALYSE procedure. The default for DMETHOD is automatic, and the default for MAXCYCLE= is 99,50. For the other options the defaults are to use the same settings as in the HGANALYSE command that performed the original analysis.

By default, the random terms are dropped from the most recent HGLM analysis, but you can use the SAVE option to supply the save structure from some earlier analysis.

One point to note is that we are testing the random terms against a null hypothesis (that they have zero variance components) which is on the boundary of the parameter space. To allow for this, Lee, Nelder & Pawitan (2006, p. 219) suggest using the critical value for twice the required significance probability or, equivalently, dividing the chi-square probabilities by two. This is not done in the procedure, but is something to bear in mind when assessing the results.

Options: PRINT, LMETHOD, DMETHOD, EMETHOD, MLAPLACEORDER, DLAPLACEORDER, MAXCYCLE, EXIT, TOLERANCE, ETOLERANCE, GROUPTERM, SAVE.
Parameters: TERMS, TESTSTATISTIC, DF.

Method

The HGLM is refitted omitting each of the random terms of interest, and its effect is assessed using the change in the profile likelihood $-2 \times P_{\beta,v}(h)$, as suggested by Lee & Nelder (2006).

References

Lee, Y., & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *Journal of the Royal Statistical Society, Series B*, **58**, 619-678.

Lee, Y., & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGSTATUS, HGWALD.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HGSTATUS

Displays the current HGLM model definitions (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Option

SAVE = *pointer*	Save structure (from HGANALYSE) to provide details of the HGLM; if omitted, information is printed for the most recently defined or fitted HGLM

No parameters

Description

HGSTATUS is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGSTATUS allows you to display the current definitions of the various models.

By default the definitions are for the most recently defined or fitted HGLM, but you can use the SAVE option to supply the save structure for some other HGLM.

Options: SAVE.
Parameters: none.

References

Lee, Y. & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *J. R. Statist. Soc. B*, **58**, 619-678.

Lee, Y. & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGWALD.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HGWALD

Prints or saves Wald tests for fixed terms in an HGLM (R.W. Payne, Y. Lee, J.A. Nelder & M. Noh).

Options

PRINT = *string token*	Controls printed output (waldtests); default wald
FACTORIAL = *scalar*	Limit on number of factors in the model terms generated from the TERMS parameter; default 3
SAVE = *pointer*	Specifies the save structure (from HGANALYSE) of the analysis from which to calculate the tests; default uses the most recent analysis

Parameters

TERMS = *formula*	Model terms for which tests are required
WALDSTATISTIC = *scalar* or *pointer* to *scalars*	Saves Wald statistics
DF = *scalar* or *pointer* to *scalars*	Saves d.f. of Wald statistics

Description

HGWALD is one of several procedures with the prefix HG, which provide tools for fitting the hierarchical and double hierarchical generalized linear models (HGLMs and DHGLMs) defined by Lee & Nelder (1996, 2001, 2006) and described by Lee, Nelder & Pawitan (2006). The models are defined by the HGFIXEDMODEL, HGRANDOMMODEL and HGDRANDOMMODEL procedures, and fitted by the HGANALYSE procedure. HGWALD allows you to print or save Wald tests for terms that can be dropped from the fixed model of an HGLM.

By default, HGWALD produces tests for all the fixed terms that can be dropped: that is, for every term that is not marginal to another term in the fixed model. For example, in the formula

```
A + B + C + D + A.B + A.D + B.D
```

the terms C, A.B, A.D and B.D can be dropped as there are no other terms in the model that contain all their factors (i.e. none to which thay are marginal). However, A cannot be dropped until A.B and A.D have been dropped. You can use the TERMS parameter to request Wald tests for a specific set of terms, but a missing value is given for any term that cannot be dropped. The FACTORIAL option sets a limit on the number of factors in each term that is formed from the TERMS formula (default 3).

If option PRINT=waldtests (the default), HGWALD prints a table with columns containing the Wald statistic, its number of degrees of freedom and a probability value. The probabilities are calculated assuming chi-square distributions. These should be used with caution as they are based on the asymptotic properties of the statistic, and are likely to show downwards bias (i.e. to give too many significant values) with ordinary data sets.

The WALDSTATISTIC parameter can save the statistics, and the DF parameter can save their numbers of degrees of freedom. If you are making a Wald test for a single term, you can supply a scalar for each of these parameters. However, if you have several terms, you must supply a pointer which will then be set up to contain as many scalars as there are terms.

Options: PRINT, FACTORIAL, SAVE.
Parameters: TERMS, WALDSTATISTIC, DF.

Method

HGWALD uses FCLASSIFICATION to form the list of terms that can be dropped. It then calculates the statistics using estimates and variances saved using RKESTIMATES.

References

Lee, Y. & Nelder, J.A. (1996). Hierarchical generalized linear models (with discussion). *J. R. Statist. Soc. B*, **58**, 619-678.

Lee, Y. & Nelder, J.A. (2001). Hierarchical generalized linear models: a synthesis of generalised linear models, random-effect models and structured dispersions. *Biometrika*, **88**, 987-1006.

Lee, Y. & Nelder, J.A. (2006). Double hierarchical generalized linear models (with discussion). *Appl. Statist.*, **55**, 139-185.

Lee, Y., Nelder, J.A. & Pawitan, Y. (2006). *Generalized Linear Models with Random Effects: Unified Analysis via H-likelihood*. Chapman and Hall, Boca Raton.

See also

Procedures: HGANALYSE, HGDISPLAY, HGDRANDOMMODEL, HGFIXEDMODEL, HGFTEST, HGGRAPH, HGKEEP, HGNONLINEAR, HGPLOT, HGPREDICT, HGRANDOMMODEL, HGRTEST, HGSTATUS.

Genstat Reference Manual 1 Summary section on: Regression analysis.

HPCLUSTERS

Prints a set of clusters (R.W. Payne).

Option

UNITS = *variate* or *text*	Names to use for the units in the clusters

Parameters

CLUSTERS = *pointers*	Clusters to print
EXTRA = *pointers*	Extra information to print

Description

HPCLUSTERS can print a set of clusters. The clusters are specified by the CLUSTERS parameter, in a pointer containing a variate for each cluster. The variates contain the numbers of the units in their respective clusters (and the numbers are the row or column positions of their units in the similarity matrix used by HCLUSTER). The cluster pointers can be formed by the HFCLUSTERS procedure.

The UNITS option can be set to a text or a variate, to provide textual labels or other numbers to use for the units of the clusters, instead of the numbers in the CLUSTER variates.

You can supply extra information to print, in a pointer, using the EXTRA parameter. It should contain variates, texts or factors, with the same number of values as the CLUSTERS pointer.

Option: UNIT.
Parameters: CLUSTERS, EXTRA.

See also

Directive: HCLUSTER.
Procedure: HFCLUSTERS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

IDENTIFY

Identifies an unknown specimen from a defined set of objects (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (identification, transcript); default iden, tran
METHOD = *string token*	Type of run (batch, interactive); if this is not set IDENTIFY checks whether the run of Genstat itself is batch or interactive
TAXA = *text* or *factor*	Names for the taxa (i.e. the objects); default uses the positive integers 1, 2...
NMISTAKE = *scalar*	Number of mistakes to allow for; default 0
IDENTIFICATION = *text*	Saves the names of the taxa that are identified; default * i.e. not saved
DIFFERENCES = *variate*	Saves the number of differences between the observed character states and those that can be displayed by each taxon; default * i.e. not saved

Parameters

CHARACTER = *factors* or *tables*	Define the characteristics of the taxa; must be set
OBSERVATION = *scalars* or *texts*	Can define an observation for each character; default * i.e. none
COST = *scalars*	Costs of observing each character; default 1

Description

IDENTIFY allows you to identify an unknown specimen from a set of possible *taxa*, for example species of plant, types of machine fault, strains of bacteria, and so on. The specimen is identified by comparing observations that you specify for the specimen against the characteristics that you have defined for the taxa. Each character is assumed to have a set of distinct possible *states*, which are represented by the levels of a factor.

So, IDENTIFY assumes that the values of the characters are discrete. Often the characters will be binary, representing the presence or absence of some attribute. Alternatively, they may involve counts, for example of numbers of leaves or petals. If you want to use continuous variables, you will need to classify the values into ranges (for example using the GROUPS directive).

Generally, the properties of the taxa with respect to each character can be defined by a factor, whose levels represent the range of values that can occur for the character. If a taxon only ever displays one state of the character (i.e. if it has a *fixed* response), the unit of the factor corresponding to that taxon should be set to the relevant level. Conversely, if different specimens of the taxon can display different states of the character (i.e. it has a *variable* response), the unit should contain a missing value.

Representing the properties for a character by a factor assumes that, if a taxon is variable, any of the states of the character may occur. Information will thus be lost for taxa that can show several, but not all, of the states of a character. An alternative representation, therefore, uses a table classified by one factor representing the states of the factor, and another representing the taxa. So, there is a row of the table for each taxon, and this contains a zero value for the states that the taxon cannot display, and a non-zero value (usually one) for those that it can display. The table below defines the texture of the bark for the trees in the example for IDENTIFY.

	smooth	rough	corky	scored horizontally	scaling
Ash	1	1	0	0	0
Beech	1	0	0	0	0
Birch	0	0	0	0	1
Elder	0	0	1	0	0
Elm	0	1	0	0	0
Lime	1	0	0	0	0
Oak	0	1	0	0	0
Plane	0	0	0	0	1
Rowan	0	0	0	1	0
Sweet chestnut	0	1	0	0	0
Sycamore	1	0	0	0	1

Most of the trees have fixed responses, for example all Beech trees have smooth bark, and all Elm trees have rough bark. However, Ash trees may have either smooth or rough bark but not, for example, corky bark.

The factors and/or tables defining the properties of the taxa must be listed using the CHARACTER parameter. If any of these is a table, the TAXA option must be set to the factor used to represent the taxa there. The levels of the factor (or its labels if present) then supply names for the taxa that are used in the output. If there are no CHARACTER tables, TAXA can be set to a text containing the taxon names instead. If TAXA is not set, IDENTIFY uses the integers 1, 2... The COST parameter can be used to supply a list of scalars indicating the cost of observing each character; if this is not set, the costs are all assumed to be equal to one.

The METHOD option defines whether IDENTIFY operates interactively, or in batch mode. If this is not set, IDENTIFY checks whether Genstat itself is running interactively or in batch. In an interactive run, IDENTIFY displays menus to guide you through to achieving an identification. The main menu allows you to select any one of the following actions.

1) list potential identifications – IDENTIFY compares the observations that you specify for the specimen against the characteristics that you have defined for the taxa. It then lists the taxa (if any) that can display all of the character states that you have observed, then those that can display all except one, all except two, and so on. The list is displayed in sections, and you can terminate it at any time.
2) select and observe a character – IDENTIFY assesses the characters, and lists them in order of their effectiveness. Alongside each one it prints an estimate of the number (of cost if the COST parameter has been set) of the characters that must be observed to complete the identification, assuming that this one is observed next. After you have chosen a character, it displays another menu for you to specify the state that you have observed.
3) specify an observed character (find in list) – IDENTIFY lists the characters so that you can indicate which one you wish to observe next. After you have chosen a character, it displays another menu for you to specify the state that you have observed.
4) specify an observed character (type name) – IDENTIFY asks you to type the name of the character that you wish to observe next. If you type just the initial part of the name, IDENTIFY will give you a list of all the characters whose names begin like that. After you

have chosen a character, it displays another menu for you to specify the state that you have observed.

5) modify an observation – IDENTIFY lists the characters that have already been observed to allow you to choose which you want to modify. After you have chosen a character, it displays another menu for you to specify the revised value.
6) display observations – IDENTIFY displays the characters that have already been observed.
7) list the characteristics of a taxon – IDENTIFY lists the taxa so that you can indicate the one whose characteristics you wish to display.
8) show differences between 2 taxa – IDENTIFY lists the taxa so that you can indicate the two that you want to compare. IDENTIFY then lists the characters that differ between them.
9) set configuration options – IDENTIFY generates a menu allowing you to set various configuration options. Firstly, you can ask IDENTIFY to take account of a specified number of mistakes in your observations. It will then up to this number of differences between your observations and the characteristics of each taxon when suggesting which character to observe next, or when making an identification. The initial setting for the number of mistakes is set by the NMISTAKE option, with a default of zero (i.e. none). You can also control whether or not to produce a transcription of your activities and whether or not to print the identification obtained at the end of your run. The initial settings for these two aspects are set by the PRINT option; by default both are printed.
10) start a new identification (clearing observed characters) – IDENTIFY clears the current observations so that you can start again.
11) save/print identification and then exit – IDENTIFY prints and saves the identification, as requested, and then stops.

The identification is saved by setting the IDENTIFICATION option to a text to contain the names of all the taxa that can display the observed character states, allowing for any requested number of mistakes. You can also set the DIFFERENCES option to a variate to contain the number of differences between the observed character states and those that can be displayed by each taxon.

For a batch run, you should use the OBSERVATION parameter to supply values for all the characters that you have observed. These can be either scalars (referring to levels of the factor) or one-line texts (referring to its labels), or a missing value to denote characters that have not been observed. This parameter can be also used in an interactive run, as an alternative to supplying the observations through the menus.

Options: PRINT, METHOD, TAXA, NMISTAKE, IDENTIFICATION, DIFFERENCES.
Parameters: CHARACTER, OBSERVATION, COST.

Method

At each stage, IDENTIFY uses the QUESTION procedure to allow you to choose what action to take. The efficiency of the characters is asssessed using the selection criterion function *CMV'* of Payne (1981).

Reference

Payne, R.W. (1981). Selection criteria for the construction of efficient diagnostic keys. *Journal of Statistical Planning and Inference*, **5**, 27-36.

See also

Directive: IRREDUNDANT.
Procedures: BKIDENTIFY, BCIDENTIFY, BCFIDENTIFY.

IFUNCTION

Estimates implicit and/or explicit functions of parameters (W.M. Patefield).

Options

PRINT = *string token*	What to print (estimates, correlations, monitoring); default esti
NOMESSAGE = *string token*	Which warning messages to suppress (parameter, convergence); default *
NPARAMETER = *scalar*	Number of parameters; default zero
MAXCYCLE = *scalar*	Maximum number of iterations; default 20
STRINGENCY = *scalar*	Stringency of tests for convergence, 0,1,2...etc; default 5
EXITCONTROL = *string token*	Control for exit on fault detection (job, procedure); default job for batch jobs, proc for interactive
ZCALCULATION = *expression structures*	Specify the calculation of ZERO and DZBIMPLICIT
DZPCALCULATION = *expression structures*	Specify the calculation of DZBPARAMETER
ECALCULATION = *expression structures*	Specify the calculation of EXPLICIT, DEBPARAMETER and DEBIMPLICIT

Parameters

IMPLICIT = *variate* or *pointer to scalars*	Implicit functions
INITIAL = *variate*	Initial values for IMPLICIT functions
LOWER = *variate*	Lower bounds to IMPLICIT functions; default -10^{10}
UPPER = *variate*	Upper bounds to IMPLICIT functions; default $+10^{10}$
VCOVARIANCE = *symmetric matrix*	Variance-covariance matrix of parameter estimates
ZERO = *variate*	Equations defining implicit functions (values calculated by ZCALCULATION)
DZBIMPLICIT = *matrix*	First derivatives of equations ZERO with respect to implicit functions IMPLICIT (values calculated by ZCALCULATION); rows correspond to ZERO, columns correspond to IMPLICIT
DZBPARAMETER = *matrix*	First derivatives of equations ZERO with respect to parameters (must not be set for NPARAMETER=0; values calculated by DZPCALCULATION); rows correspond to ZERO, columns to parameters
DIBPARAMETER = *matrix*	First derivatives of IMPLICIT functions with respect to parameters (must not be set for NPARAMETER=0); rows correspond to IMPLICIT, columns correspond to parameters
EXPLICIT = *variate* or *pointer to scalars*	Explicit functions of parameters and/or implicit functions (values calculated by ECALCULATION)
DEBPARAMETER = *matrix*	First partial derivatives of EXPLICIT functions with respect to parameters (values calculated by ECALCULATION); rows correspond to EXPLICIT, columns correspond to parameters
DEBIMPLICIT = *matrix*	First partial derivatives of EXPLICIT functions with respect to IMPLICIT functions (values calculated by

	ECALCULATION); rows correspond to EXPLICIT, columns correspond to IMPLICIT
DFBPARAMETER = *matrix*	First derivatives of ESTIMATES with respect to parameters; rows correspond to ESTIMATES, columns correspond to parameters
ESTIMATES = *variate*	Estimates of IMPLICIT and EXPLICIT functions
SE = *variate*	Standard errors of ESTIMATES
CORRELATIONS = *symmetric matrix*	Correlation matrix of ESTIMATES
FCOVARIANCE = *symmetric matrix*	Variance-covariance matrix of ESTIMATES

Description

IFUNCTION solves implicit equations of functions of parameters. The equations are specified by the variate ZERO, the *i*th element defining the *i*th equation in terms of the IMPLICIT functions. The parameters ZERO and IMPLICIT must be of the same length (*n*), IMPLICIT being either a variate or a pointer to *n* scalars. The option ZCALCULATION supplies expressions for the calculation of both ZERO and the *n* by *n* matrix DZBIMPLICIT of first derivatives of ZERO with respect to the IMPLICIT functions. The element in the *i*th row and *j*th column of DZBIMPLICIT is the (partial) derivative of the *i*th element of ZERO with respect to the *j*th element of IMPLICIT. DZBIMPLICIT is initialized to zero and hence only non-zero elements need be calculated by ZCALCULATION.

The values of the IMPLICIT functions satisfying ZERO = 0 are obtained iteratively. Initial values may be given as a variate in the parameter INITIAL. If INITIAL is not set any current values of IMPLICIT are used as initial values. Output is controlled by the PRINT option. The option NOMESSAGE allows warning messages to be suppressed. The option MAXCYCLE and the parameters LOWER and UPPER are similar in their effect to their use in the RCYCLE directive. The option STRINGENCY controls the stringency with which tests for convergence are applied, higher values being more stringent. The option EXITCONTROL controls the action on fault detection. IFUNCTION may be used to solve n simultaneous nonlinear equations in *n* unknowns (the IMPLICIT functions) by not setting the NPARAMETER option (or setting it to zero). More generally, the variate ZERO is a function of both the IMPLICIT functions and NPARAMETER parameter estimates from a model previously fitted using FIT, FITCURVE or FITNONLINEAR. The DZPCALCULATION option supplies expressions for calculation of the *n* by NPARAMETER matrix DZBPARAMETER of (partial) derivatives of ZERO with respect to the model parameters (only non-zero elements need be calculated).

In addition (or instead) *m* explicit functions of the model parameters and/or the IMPLICIT functions may be specified by the parameter EXPLICIT, a variate of length *m* or a pointer to *m* scalars. The (partial) derivatives of the EXPLICIT functions with respect to the model parameters are given by the *m* by NPARAMETER matrix DEBPARAMETER and the (partial) derivatives with respect to the IMPLICIT functions by the *m* by *n* matrix DEBIMPLICIT. If either of these matrices is not set, then it is taken to be zero (i.e. the EXPLICIT functions do not depend on the model parameters or the IMPLICIT functions respectively). Expressions for calculating EXPLICIT, DEBPARAMETER and DEBIMPLICIT are supplied by the option ECALCULATION, the two matrices being initialized to zero and hence only their non-zero elements need be calculated. For EXPLICIT functions dependent on model parameters only (i.e. not on any IMPLICIT functions), ECALCULATION need not be set, in which case their values must be supplied by EXPLICIT and their (partial) derivatives with respect to model parameters by DEBPARAMETER on entry to IFUNCTION.

The parameters ZERO, DZBIMPLICIT, DZBPARAMETER, DEBPARAMETER and DEBIMPLICIT entering into the calculations ZCALCULATION, DZPCALCULATION and ECALCULATION need not be declared before using IFUNCTION. If they are declared they must have the correct attributes.

The only exception to this is when derivatives of the EXPLICIT functions are supplied directly in the matrix DEBPARAMETER rather than obtained by calculations using ECALCULATION.

It is essential that the expressions for calculating DZBIMPLICIT are formulated correctly. If they are not, faults such as divergence of the optimization algorithm or estimates becoming out of bounds may be detected and reported. Fault CA16 may also be caused by incorrectly calculating DZBIMPLICIT as a singular matrix.

The variance-covariance matrix of the fitted parameters is supplied by the parameter VCOVARIANCE containing the variance-covariance matrix from a previous FIT, FITCURVE or FITNONLINEAR.

Estimates of all $n+m$ functions (n IMPLICIT and m EXPLICIT functions of parameters) are saved by the parameter ESTIMATES. Their derivatives with respect to the model parameters are saved by the parameter DFBPARAMETER. Their variance-covariance matrix is saved by the parameter FCOVARIANCE. The standard errors of, and correlations between, the ESTIMATES are saved by the parameters SE and CORRELATIONS.

Options: PRINT, NOMESSAGE, NPARAMETER, MAXCYCLE, STRINGENCY, EXITCONTROL, ZCALCULATION, DZPCALCULATION, ECALCULATION.

Parameters: IMPLICIT, INITIAL, LOWER, UPPER, VCOVARIANCE, ZERO, DZBIMPLICIT, DZBPARAMETER, DIBPARAMETER, EXPLICIT, DEBPARAMETER, DEBIMPLICIT, DFBPARAMETER, ESTIMATES, SE, CORRELATIONS, FCOVARIANCE.

Method

The implicit functions are calculated by solving the simultaneous equations ZERO = 0 iteratively using Newton-Raphson. It is assumed that a solution exists and that the initial values are sufficiently close to a solution for the optimization to converge. Poor initial values can lead to divergence. A warning message is given when divergence is detected. Reasonable initial values may be obtained by using FITNONLINEAR to minimize the function $k \times$ MAX(ABS(ZERO)), with k equal to a large number such as 10^6.

A maximum of three convergence criteria may be employed. They are:

(i) the Increment criterion defined as MAX(ABS(Inc) / MAX(ABS(IMPLICIT), 1)), where Inc is the variate of implicit function increments in the iterative process,

(ii) the Zero criterion defined as MAX(ABS(ZERO) / Scaling-variate) where the Scaling-variate is the greater of the maximum value of ZERO over all cycles of the iterative process and 0.0001, and

(iii) the Gradient criterion defined as ABS(T(Inc) *+ DZBIMPLICIT *+ Inc).

The values of criterion (ii) may be highly dependent on the initial parameter values and criterion (iii) is of use primarily when the equations ZERO = 0 are derivatives of a scalar function and DZBIMPLICIT is the matrix of second derivatives of the function.

Convergence is completed when criterion (i) cannot be further reduced. However the iterative process continues searching for lower values until other criteria cannot be further reduced. The criteria involved are determined by the STRINGENCY option. For STRINGENCY = 0 or 1 only criterion (i) is used. For STRINGENCY = 2 or 3 criterion (ii) is also used. STRINGENCY = 1 or 3 requires convergence at two successive iterations. For STRINGENCY = 4 or 5 all criteria are used, STRINGENCY = 5 requiring convergence of both criteria (i) and (ii) at two successive iterations. Higher values of STRINGENCY require convergence of all three criteria at increasing numbers of successive iterations.

The default STRINGENCY value of 5 is recommended at least until the expressions for calculations are validated. Low values may give convergence at incorrect values of the implicit functions, particularly with poor INITIAL values when the equations ZERO are not approximately linear. High values will often result in an unneccessarily large number of iterations. IFUNCTION calculates the matrix DIBPARAMETER of derivatives of the implicit

functions with respect to the model parameters (Marsden, 1984, page 211). The matrices DEBPARAMETER and DEBIMPLICIT of partial derivatives of any explicit functions with respect to the model parameters and the implicit functions respectively are evaluated using expressions supplied in ECALCULATION. By the chain rule, the derivatives of the explicit functions with respect to the parameters are given by

```
DEBPARAMETER + ( DEBIMPLICIT *+ DIBPARAMETER ).
```

This matrix is appended to DIBPARAMETER to form the $n+m$ by NPARAMETER matrix DFBPARAMETER of derivatives of the length $n+m$ variate

```
ESTIMATES = !( #IMPLICIT, #EXPLICIT )
```

with respect to the model parameters.

The variance-covariance matrix of model parameters resulting from a previous FIT, FITCURVE or FITNONLINEAR is supplied by the parameter VCOVARIANCE, and the variance-covariance matrix of the ESTIMATES of both the implicit and explicit functions is computed as

```
FCOVARIANCE = QPRODUCT(DFBPARAMETER; VCOVARIANCE).
```

Action with RESTRICT

None of the parameters of IFUNCTION may be restricted.

Reference

Marsden, J.E. (1984). *Elementary Classical Analysis*. W.H. Freeman and Company, San Francisco.

See also

Genstat Reference Manual 1 Summary section on: Regression analysis.

IMPORT

Reads data from a foreign file format, and loads it or converts it to a spreadsheet file (D.B. Baird).

Options

`PRINT` = *string token*	What information to print (`catalogue`, `summary`); default `cata`
`OUTTYPE` = *string token*	Output file type (`GEN`, `GSH`, `GWB`, `XLS`, `XLSX`, `TXT`, `CSV`, `SHEETS`); default `GWB`
`METHOD` = *string token*	Whether to load data into the Genstat server after creating the file, or merely to create the file (`create`, `load`); default `load`
`IMETHOD` = *string token*	How identifiers are to be specified for the columns (`read`, `supply`, `none`, `overlay`); default `supply` if `COLUMNS` is set (and specifies names rather than just types), otherwise `read`
`ENDSTATEMENT` = *string token*	Ending statement for a type `GEN` output file (`return`, `endbreak`); default `retu`
`SPSSMV` = *string token*	What to do with SPSS missing value codes (`ignore`, `convert`); default `conv`
`MISSING` = *text*	What labels represent missing values in Excel, Quattro or Lotus files; default `'*'`
`FORDER` = *string token*	The order in which to define the labels or levels of a factor (`sorted`, `unsorted`); default `sort`
`TEXTCONVERSION` = *string token*	How to convert text to numbers for the columns (`strict`, `single`, `common`, `standard`, `lax`); default `stan`
`KEEPEMPTY` = *string tokens*	Whether to retain any empty rows or columns found in the data (`rows`, `columns`, `none`); default `none`
`NAMEROW` = *scalar*	The row number within an Excel or Quattro spreadsheet which contains the column names (`IMETHOD` must be unset or set to `read`); default, the first row in `CELLRANGE`
`EMETHOD` = *string token*	Whether to read column descriptions/extra from Excel, SigmaPlot or Quattro spreadsheets (`read`, `none`); default `none`
`EXTRAROW` = *scalar*	The row number within an Excel or Quattro spreadsheet which contains the column descriptions (`EMETHOD` must be set to `read`); default, the second row in `CELLRANGE`
`PREFIX` = *text*	The string with which to prefix numerical column names; default `'%'`
`TEMPMISSING` = *string token*	Whether to read temporarily missing values as missing (`yes`, `no`); default `no`
`INOPTIONS` = *text*	Optional input file arguments to be passed to the `Dataload.dll`
`OUTOPTIONS` = *text*	Optional output file arguments to be passed to the `Dataload.dll`
`RGBMETHOD` = *string token*	How to read colour values (`combined`, `separate`, `matrix`); default `sepa`
`SEPARATORS` = *text*	Alternative separators to use in text or csv files
`SCOPE` = *string token*	Whether to create the data locally in a procedure that is

	using IMPORT, or globally in the whole program (local, global); default loca
IPREFIX = *text*	Prefix to use with unnamed columns, default 'C'
TRANSPOSE = *string token*	Whether to transpose the rows and columns of the input file (yes, no); default no
UNICODE = *string token*	What to do with Unicode characters found e.g. in Excel XLSX input files (utf8, typeset, ascii, remove); default utf8
COLUNICODENAMES = *string token*	How to convert Unicode column names (suffix, extra, ignore) default suff
UNINAME = *text*	Name of the pointer for Unicode column names used as suffixes; default 'C'
†XLSCONTENT = *string tokens*	What content to read from an Excel XLSX file (values, formulae, forecolour, backcolour, fontname, style, size); default valu

Parameters

FILE = *texts*	Input file or URL to be read
OUTFILE = *texts*	Name of the output file to be created; if this is not provided a temporary file will be created, and then deleted if the data are loaded
SHEETNAME = *texts* or *scalars*	Name of a spreadsheet worksheet or named range, or number of a worksheet within the file; default is the first sheet in the file
CELLRANGE = *texts*	Cell range within a worksheet, giving the top left and bottom right cell in the format XXNN:XXNN where XX = A - IV, NN = 1 - 64384; default * requests all data on the sheet
COLUMNS = *texts*	Names and/or type codes for the columns read (the type of column can be forced by ending the column name, if supplied, with the code **!** for a factor, **#** for a variate, and **$** for a text), using a name of '*' will cause a column to be dropped
ISAVE = *pointers*	Saves the identifiers of the columns
START = *texts*	Contents of a cell in a spreadsheet file or a line in a text file from which to start reading
END = *texts*	Contents of a cell in a spreadsheet file or a line in a text file at which to end reading
ANCILLARY = *texts*	Extra information returned by some file formats (currently only population type from QTL location files)
ROWSELECTION = *variates*	Numbers of the rows to import; if unset, all rows are imported
COLSELECTION = *variates* or *texts*	Numbers or names of the columns to import; if unset, all the columns are imported

Description

The name of the file, containing the data values to be imported, is specified by the FILE parameter. This can also be an internet URL prefixed with http://, https:// ftp:// or file://. The data source is then downloaded and imported. Note: more control over the reading of GRIB2 meteorological data files is provided by the GRIBIMPORT procedure, which allows subsets of records and the grid to be loaded.

Data in the supported file formats are extracted and saved in the specified file format, depending on the extension of OUTFILE. If this is not provided, the type is indicated by the OUTTYPE option, as either GEN (Genstat Command file), GSH (Genstat Spreadsheet), GWB (Genstat Spreadsheet Book), XLS (Excel 5 Spreadsheet), XLSX (Excel 2007 Spreadsheet), TXT (ASCII Text file) or CSV (comma-delimited file); the default is GSH. Setting OUTTYPE=SHEETS reads in the worksheet names in a spreadsheet file (Excel/Quattro/Sigmaplot or SAS Transport) into a text named Worksheets. The ENDSTATEMENT option specifies the ending statement type for a type GEN output file: either RETURN (the default) or ENDBREAK). You can set ENDSTATEMENT=* if you do not want an ending statement.

The PRINT option controls printed output, with the following settings:

catalogue	lists the contents of the file (default); and
summary	prints a summary of the values in each data structure in the file.

If METHOD=load, the resulting file is read in to Genstat data structures. When IMPORT is used within a procedure, the SCOPE option controls whether the structures are created locally in the procedure (default), or globally in the main program.

In spreadsheet files (Excel, Quattro, 123, SigmaPlot), the SHEETNAME and CELLRANGE parameters can be used to read just a specified section of the data in the file. If CELLRANGE specifies only the starting cell, IMPORT reads all columns the from the given column onwards, and all rows from the given row downwards. For example, CELLRANGE='C8' reads columns C, D... onwards, and rows 8, 9... downwards, until the end of the data in the sheet. The COLUMNS parameter can be used to set the names and types of the structures (see below).

In spreadsheet files, the data that are extracted are labels, numerical values and the results from formulae. A label of * in an otherwise numerical column is taken as a missing value, unless one or more different missing value markers are specified with the MISSING option. Empty cells are taken as missing values. Empty rows at the start, middle and end of a block are removed. Empty columns are ignored by default; you can set the option KEEPEMPTY=rows or KEEPEMPTY=columns to retain empty rows or columns respectively, or KEEPEMPTY=rows,columns to keep both.

The IMETHOD option indicates how identifiers are to be specified for the columns, with the following settings:

read	assumes that the names are in the first non-empty row of data;
supply	assumes that the names are supplied by the COLUMNS parameter, but uses default names if they are not;
none	uses default names; and
overlay	uses the names from the COLUMNS parameter, or from the first non-empty row of data for any of those names that is blank.

If IMETHOD=read, and a column name cell contains a numerical value rather than a label, the column name is set to the numerical value prefixed with a % character. The prefix can be changed using the PREFIX option: a column named '15' is given the name %15 by default but, if PREFIX='X', the name would be X15.

The default for IMETHOD is to take the names from the COLUMNS parameter, if this is set and it contains names. Otherwise IMPORT looks for names in the data file (as with the read setting).

The default column names have the prefix C and an integer number (i.e. C1, C2 etc.), but you can supply your own prefix using the IPREFIX option.

The COLUMNS parameter can be used to specify the names for the columns, in a text. The type of each column can be forced by providing a **!**, **#** or **$** character on the end of its string. A string **'*'** can be given as a name in COLUMNS, to remove a column from the data that are read. If only a single type character is given, only the types of the column (and not its names) is changed. The

extension **:D** on a column name specifies that the values are to be read as dates. Similarly, when the column names are being read from a spreadsheet, their types can be specified by using **!** for a factor, **#** for a variate, **$** for a text and **:D** for a date.

The option FORDER controls the order in which the labels or levels of a factors are stored. With the default, FORDER=sorted, the levels are stored in ascending numerical order, and the labels are stored in alphabetical order. Alternatively, if FORDER=unsorted the levels and labels are stored in the order in which they are first met in the column.

The TEXTCONVERSION option controls how labels are converted to numbers in a column marked as a variate:

strict	only labels that contain numeric data only are converted (e.g. '10' becomes 10; '1O' becomes *)
single	a single character substitution is read as a number (o or O become 0; i, I, l or L become 1; s or S become 2; z or Z become 5; comma becomes decimal point)
common	multiple substitutions as in single are made (e.g. 'Io' becomes 10; '23X' becomes *)
standard	as in common but extra text is ignored at the end of the number (e.g. '23X' becomes 23; 'A2X3' becomes *)
lax	any digits are read from the text (e.g. 'A2X3' becomes 23).

You can set option EMETHOD=read, to read a row of column descriptions/extra from a spreadsheet file. By default, this row is taken as the second row in CELLRANGE. The EXTRAROW option can be used to modify the row form which the description is read. The row number is relative to the start of the cell range, unless a negative row number is provided; the descriptions are then read from the row in the spreadsheet, corresponding to the absolute value of the specified row number. If EXTRAROW=1, the column names are read from the second row.

The START parameter can supply a text to indicate where to start reading within a spreadsheet or text file. In a spreadsheet file (Excel, Quattro, Lotus), the cells from A1 are searched row by row, until a label is found that matches the text. Only cells below and to the right of this cell are then imported. The text could thus be the name of the first variable to be read. Note that the text must not contain spaces or the division symbol (/). Similarly the END parameter can supply a text to indicate where to stop reading a spreadsheet or text file.

The TEMPMISSING option controls the input of temporarily missing values. These are values that have been set to missing temporarily in the spreadsheet, and for which the original (non-missing) values are still available. The default is to read the original values, but you can set TEMPMISSING=yes to read them as missing values instead.

The INOPTIONS and OUTOPTIONS options allow extra options to be passed to Dataload.dll. For example: setting INOPTIONS='/k' keeps leading and trailing and doubled blanks in strings, OUTOPTIONS='/u' creates undecorated names in a CSV file (i.e. 'Factor', rather than 'Factor!'), OUTOPTIONS='/c' combines the three columns Red, Green and Blue in a BMP file into a single column RGB, and INOPTIONS='/m' loads the data as a matrix rather than as separate columns.

The RGBMETHOD option controls how to represent colour values from image files (JPG, GIF, TIF or PNG). The default setting, combined, stores an RGB value in a single column in the same form as generated by the RGB function. The separate setting creates three columns containing the red, green and blue values, respectively. Finally, the matrix setting puts the RGB values into a matrix.

The ROWSELECTION and COLSELECTION parameters allow you to import only a subset of the rows or columns, respectively, in the file. They can be set to a variate containing the numbers of the rows or columns. With COLSELECTION, you can also supply a text containing column names. So, for example, to import only rows where the variate X is greater than zero, you could

put

```
ROWSELECTION = WHERE(X.GT.0)
```

(the WHERE function gives the unit numbers where a logical expression has the value one i.e. true). Note that the variate X must already have been imported into Genstat, but you could import this column on its own using COLSELECTION. If ROWSELECTION (or COLSELECTION) are unset, all the rows (or columns) are imported.

The UNICODE option controls what happens to Unicode characters that are not part of the extended ASCII character set. These may occur, for example, in Excel XLSX files. The default setting, utf8, converts them into the UTF-8 format. In this format, the ASCII characters are stored in the usual way, in a single *byte* (of eight binary *bits*). More complicated characters, such as Chinese and Thai characters, require up to four bytes. UTF-8 characters cause no problems with most of the Genstat commands. The commands that cannot handle them, for example EDIT, issue a VA-43 fault. The remove setting removes UTF-8 characters from the input. The ascii option converts them to the nearest matching ASCII character. Finally, the typeset option replaces those that can be represented by Genstat typesetting strings by these strings: for example, α would be replaced by ~{alpha}, and √ would be replaced by ~{sqrt}. The correspondence between the lower-case Greek, ASCII and type-setting commands is shown in the table below; the capital letters have a similar correspondence. Extended Greek and Latin letters have their accents removed, as there are no type-setting commands for these. Some symbols like ♂ and ♀ are converted to their text equivalent (male and female).

α	a	~{alpha}	ι	I	~{iota}	ρ	r	~{rho}
β	b	~{beta}	κ	k	~{kappa}	σ	s	~{sigma}
γ	g	~{gamma}	λ	l	~{lambda}	τ	t	~{tau}
δ	d	~{delta}	μ	m	~{mu}	υ	u	~{upsilon}
ε	e	~{epsilon}	ν	n	~{nu}	φ	f	~{phi}
ζ	z	~{zeta}	ξ	c	~{xi}	χ	x	~{chi}
η	h	~{eta}	o	o	~{omicron}	ψ	y	~{psi}
θ	q	~{theta}	π	p	~{pi}	ω	w	~{omega}

The COLUNICODENAMES option controls how column names that contain Unicode characters are used. With the default setting, suffix, a pointer is defined to hold any columns with Unicode in their names, and the column names provide its suffix labels. The name of the pointer is specified in a text by the UNINAME option (default 'C'). The extra setting uses the default names for the columns, and the column names from the file are used as extra texts. It also sets the IPRINT attribute of the columns to extra, so that these are printed instead of the default identifiers. (You can modify this to print the default idendifiers instead, by using the Identifying information used in output list box for those columns in the spreadsheet Column Attributes/Format menu.) The ignore setting removes the Unicode characters from the name.

The XLSCONTENT option specifies the content to import from an Excel XLSX file: values, formulae, foreground colour, background colour, font name, style or size. The default is to read only the values.

(Note: IMPORT replaces the procedure DATALOAD from earlier editions of Genstat.)

Options: PRINT, OUTTYPE, METHOD, IMETHOD, ENDSTATEMENT, SPSSMV, MISSING, FORDER, TEXTCONVERSION, KEEPEMPTY, NAMEROW, EMETHOD, EXTRAROW, PREFIX, TEMPMISSING, INOPTIONS, OUTOPTIONS, RGBMETHOD, SEPARATORS, SCOPE, IPREFIX, TRANSPOSE, UNICODE, COLUNICODENAMES, UNINAME, XLSCONTENT.
Parameters: FILE, OUTFILE, SHEETNAME, CELLRANGE, COLUMNS, ISAVE, START, END, ANCILLARY ROWSELECTION, COLSELECTION.

Method
The request is passed to the DATALOAD.DLL library which reads the foreign file and returns any valid data found in a temporary GEN or GSH file. The following file types are supported: Excel 2-5, 95, 97, 2000, XP, 2003, 2007-2013, Open Office, Lotus WK1, Quattro (WQ1, WB*, QPW), dBase 2-5, Paradox 3-9, Genstat GSH and GWB, SAS PC 6.03-12, 7-9, SAS Transport, SAS JMP, Minitab 8-17, Statistica 5 and 6, Systat, MStat, Instat, Epi-Info, SPSS/Win, Gauss Data/Matrix (PC/Win/Unix), MatLab, S+ (PC/Unix), Stata 4-8, StatGraphics, R data frames, Weka Attribute files, SigmaPlot 7-9, OSIRIS, Limdep, Comma delimited text files (*.CSV), Cornell Ecology format, MapQTL trait files (.QUA), ArcView/Info Shapefiles, MapInfo Exchange files, Windows Bitmap (*.BMP), Windows Sound (*.WAV), NMR Binary files and image files (JPG, GIF, TIF, PNG). The file type is worked out from the file contents, so the usual extension need not be used with the exception of the following file types which do not contain a unique signature: Epi-Info (.REC), S+ (.SDD), Paradox (.DB) and GRIB2 meteorological data files.. Any files not containing a unique file signature, but ending in these extensions, will be classified as above. Any other file extensions will attempted to be read as a comma, space or tab delimited text file.

There is a known problem that using the OUTTYPE=GEN inside a FOR loop (or another other procedure) ties up input channels until exiting the FOR loop. Thus it may exhaust the available input channels. Either use the OUTTYPE=GSH or set LOAD=no and write code to input the files created outside the loop (you will need to provide an output file name to do this).

Action with RESTRICT
Restrictions are not applicable to any of the parameters.

See also
Directive: SPLOAD.
Procedures: EXPORT, GRIBIMPORT.
Genstat Reference Manual 1 Summary section on: Input and output.

INSIDE

Determines whether points lie within a specified polygon (S.A. Harding).

Option

TOLERANCE = *scalar*	Value used for testing against zero; default 10^{-4}

Parameters

Y = *variates*	Y coordinates of points
X = *variates*	X coordinates of points
YPOLYGON = *variates*	Y coordinates of polygon
XPOLYGON = *variates*	X coordinates of polygon
INSIDE = *variates*	Indicate whether points are inside (1) the polygon, outside (−1) or on an edge (0)

Description

INSIDE takes a set of points whose *x* and *y* coordinates are specified by the X and Y parameters and determines which of these lie inside the polygon whose vertices are specified by the XPOLYGON and YPOLOGON parameters. This procedure is primarily intended for use with high-resolution graphics. It allows subsets of plotted points to be identified according to their spatial relationships so that they can be redrawn or deleted.

The output is in the form of a variate, specified by the INSIDE parameter. This will contain the value 1 for points that are located inside the polygon, 0 for those on an edge, and −1 for those outside the polygon. It can thus be used in RESTRICT, for example, to identify subsets of the values.

Usually the polygon will be defined by several points. Closure is assumed, so the last point need not be the same as the first. The polygon need not be convex. If only two points are given these are interpreted as diagonally opposite corners of a rectangle (thus maintaining compatibility with the "rubber-rectangle" type of input cursor of DREAD).

Option: TOLERANCE.
Parameters: Y, X, YPOLYGON, XPOLYGON, INSIDE.

Method

The method used is essentially that of Shimrat (1962). The algorithm counts the number of edges for which a point lies within the y-range and to the left. If this is an odd number the point must lie within the polygon. A separate check is made for points that lie on the boundary.

Action with RESTRICT

If either Y or X variate is restricted, only the restricted set of points is checked for inclusion in the polygon. Any points omitted by a restriction will be identified as lying outside the polygon. Restrictions are removed from YPOLYGON and XPOLYGON.

Reference

Shimrat, M. (1962). Position of point relative to polygon, CACM Algorithm 112. Communications of the ACM, August 1962.

See also

Procedures: DPOLYGON, PTSINPOLYGON.
Genstat Reference Manual 1 Summary sections on: Graphics, Spatial statistics.

JACKKNIFE

Produces Jackknife estimates and standard errors (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (estimates, vcovariance); default esti
DATA = *variates*, *factors* or *texts*	Data vectors from which the statistics are to be calculated
ANCILLARY = *any type*	Other relevant information needed to calculate the statistics
VCOVARIANCE = *symmetric matrix*	Saves the variance-covariance matrix for the statistics

Parameters

LABEL = *texts*	Texts, each containing a single line, to label the statistics
ESTIMATE = *scalars*	Saves the Jackknife estimate for each statistic
SE = *scalars*	Saves Jackknife estimates of the standard errors
PSEUDOVALUES = *variates*	Saves the Jackknife pseudo-values
ACCELERATION = *scalars*	Saves the acceleration parameter for bias-corrected and accelerated bootstrap confidence intervals

Description

The Jackknife provides a way of decreasing bias and obtaining standard errors in situations where the standard methods might be expected to be inappropriate. The basic form of the Jackknife method works by calculating the statistic (or statistics) of interest omitting each data value in turn. Thus, if there are n data values, n "partial estimates" $T_{-1} \ldots T_{-n}$ are obtained (where T_{-j} is the estimate omitting value j). These are combined with the estimate T obtained from all the data, to produce n pseudo-values:

$$P_j = n \times T - (n - 1) \times T_{-j} \; : \; j = 1 \ldots n$$

The Jackknife estimate of the statistic is given by the mean of the pseudo-values, and the standard error by the standard error of the mean of the pseudo-values.

The Jackknife can be shown to eliminate the term proportional to $1/n$ from a bias of the form

$$T = t + a/n + \mathrm{O}(1/n^2)$$

where t is the true value of the estimate and $\mathrm{O}(1/n^2)$ is a term of order one divided by the square of the number of observations (Quenouille 1956). However, it is not appropriate in all situations. In particular the statistic needs to be "smooth" (small changes in the data set should cause only small changes in the statistic); it will not work for example with medians or order statistics. Further details and advice are given by Miller (1974), Bissell & Ferguson (1975), Hinkley (1983) and Efron & Tibshirani (1993).

The data for JACKKNIFE are provided as a list of vectors (variates, factors or texts) using the DATA option. From this, new vectors are formed omitting each unit of the original vectors in turn, and a subsidiary procedure RESAMPLE is called to calculate the statistics. Other relevant information can be provided for passing to RESAMPLE, in any type of data structure, using the ANCILLARY option. To use JACKKNIFE, you need to provide a version of RESAMPLE to calculate the particular statistics that you require. The default RESAMPLE procedure, which accompanies JACKKNIFE in the library, merely prints details of the syntax (also described in the *Methods* Section).

A label should be provided for each statistic, using the LABEL parameter; by default, there is assumed to be a single statistic labelled simply as Statistic. The estimates, their standard errors and variates of corresponding pseudo-values for each statistic can be saved by the ESTIMATE, SE and PSEUDOVALUES parameters, respectively. Also, if there is more than one statistic, a variance-covariance matrix can be saved for the estimates using the VCOVARIANCE

option.

Printed output is controlled by the PRINT option, with settings estimates for the estimates and their standard errors, and vcovariance for the variance-covariance matrix; by default PRINT=estimates.

The jackknife is also required for the calculation of bias-corrected and accelerated confidence limits for bootstrap statistics (as given by the BOOTSTRAP procedure). The necessary acceleration quantities can be saved using the ACCELERATION parameter. For details see Efron & Tibshirani, 1993, Section 14.3.

Options: PRINT, DATA, ANCILLARY, VCOVARIANCE.
Parameters: LABEL, ESTIMATE, SE, PSEUDOVALUES, ACCELERATION.

Method

The original papers describing the Jackknife technique are by Quenouille (1949, 1956) and by Tukey (1958). Good expository accounts are provided by Hinkley (1983) or Bissell & Ferguson (1975).

JACKKNIFE needs a subsidiary procedure RESAMPLE to calculate the statistics of interest. RESAMPLE has an option, DATA, which is used to supply the data vectors (variates, factors or texts) from which the statistics are to be calculated. (On the first occasion that RESAMPLE is called, these will be the original vectors as supplied to JACKNIFE, in order to calculate the estimate *T*; subsequently, they will be new vectors containing all except one of the units.) Other relevant information can can be supplied through the ANCILLARY option, which corresponds to the ANCILLARY option of JACKKNIFE itself. RESAMPLE can be called by the BOOTSTRAP procedure, and it then also has an AUXILIARY option, but this is not relevant to JACKKNIFE.

There are two parameters: STATISTICS supplies a list of scalars to store the estimates of each statistic, and EXIT a list of scalars which should be set to zero or one according to whether or not each statistic could be estimated successfully with the supplied data vectors. If the value of EXIT is not calculated in RESAMPLE, JACKKNIFE assumes that the calculations succeeded. This example shows a version of RESAMPLE which calculates the correlation between two variates.

```
PROCEDURE [PARAMETER=pointer] 'RESAMPLE'
OPTION    'DATA',       " (I: variates, factors or texts) data
                          vectors from which to calculate the
                          statistics; no default"\
          'ANCILLARY'; " (I: any type of structure) other
                          relevant information needed to
                          calculate the statistics "\
          MODE=p; TYPE=!t(variate,factor,text),*;\
          SET=yes,no; LIST=yes; DECLARED=yes; PRESENT=yes
PARAMETER 'STATISTIC', " (O: scalars) to save the calculated
                          statistics "\
          'EXIT';       " (O: scalars) to save an exit code
                          to indicate failure (EXIT[i]=1) or
                          success (EXIT[i]=0) when calculating
                          each STATISTIC[i]"\
          MODE=p; TYPE='scalar'; SET=yes

CALCULATE STATISTIC[1] = CORRELATION(DATA[1]; DATA[2])
&         EXIT[1] = STATISTIC[1]==C('missing')

ENDPROCEDURE
```

Action with RESTRICT

If any of the data vectors is restricted, JACKKNIFE will use only the units that are not restricted for any of the vectors.

References

Bissell, A.F. & Ferguson, R.A. (1975). The jackknife – toy, tool or two-edged weapon. *The Statistician*, **24**, 79-100.

Efron, B. & Tibshirani, R.J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, London.

Hinkley, D. (1983). Jackknife methods. In: *Encyclopedia of Statistics, Volume 4* (ed: S. Kotz, N.L. Johnson & C.B. Read). Wiley, New York.

Miller, R.G. (1974). The jackknife – a review. *Biometrika*, **61**, 1-15.

Quenouille, M.H. (1949). Approximate tests of correlation in time series. *Journal of the Royal Statistical Society, Series B*, **11**, 18-44.

Quenouille, M.H. (1956). Notes on bias in estimation. *Biometrika*, **61**, 353-360.

See also

Procedures: BOOTSTRAP, APERMTEST, CHIPERMTEST, RPERMTEST.

JOIN

Joins or merges two sets of vectors together, based on the values of sets of classifying keys (C.F. Johnston & D.B. Baird).

Options

NINDEX = *scalar*	Number of index vectors in structures (up to 10); default 1
METHOD = *string token*	Type of join (inner, left, right, full); default full
REPEATS = *string token*	How to handle repeats of matches (combinations, single); default sing outputs one row per match
INCLUDE = *string token*	How to handle restrictions on the input vectors (all, nonrestricted); default all uses all the data rows
SORT = *string token*	Whether NEWVECTORS should be sorted on the index vectors (ascending, descending, unsorted); default unsorted keeps the same ordering as the input sets

Parameters

LEFTVECTORS = *pointer*	Pointer to a list of vectors in left set (keys and variables)
RIGHTVECTORS = *pointer*	Pointer to a list of vectors in right set (keys and variables)
NEWVECTORS = *pointer*	Pointer to a list of output vectors (keys and variables)

Description

This procedure can be used to produce a set of NEWVECTORS, which is the result of joining (or merging) two sets according to index (or key) vectors in each set. JOIN supports SQL style joins, as well as merges, as implemented in Genstat for Windows, SAS and SPSS.

The number of index vectors is given by the NINDEX option (up to 10). Each of LEFTVECTORS and RIGHTVECTORS is a pointer to NINDEX keys followed by any number of extra vectors. The NEWVECTORS parameter is a pointer to NINDEX keys followed by the total number of non-index vectors in the two input sets. The output order in NEWVECTORS will be the combined keys from the left and right sets, then the non-index vectors from the left set, followed by the non-index vectors from the right set. You need not have declared the pointer already; it will be declared automatically if necessary. The vectors may be variates, factors or texts. Warnings are given if the types of index vectors in each set do not match, although a factor can be matched with a text. Attempting to match a text with a factor or variate will result in a fault.

The METHOD option controls the type of join and determines which rows from each input set will be output. METHOD=inner outputs only those rows where the keys from both sets match. METHOD=left outputs all rows from the LEFTVECTORS set and only those rows from RIGHTVECTORS where the keys from both sets match. METHOD=right outputs all rows from the RIGHTVECTORS set and only those rows from LEFTVECTORS where the keys from both sets match. METHOD=full outputs all rows from both sets. Where keys do not match, missing values are inserted into the non-index vectors from the set without that key value.

The REPEATS option determines what happens when both input sets have repeats of the same matching key values. REPEATS=single outputs one row for each match, so that if there are M repeats in LEFTVECTORS and N repeats in RIGHTVECTORS, MAX(M,N) rows will be output. This is the same behaviour as the merge statements of SAS and SPSS and the Merge Spreadsheets menu of Genstat *for Windows*. REPEATS=combinations outputs all combinations of the repeats, giving M*N rows. This is equivalent to an SQL join and may produce very large output sets.

The INCLUDE=nonrestricted option allows the use of restrictions on the vectors in each input set to be used to subset the rows. The SORT=unsorted option allows the resulting vectors

to be returned in the original order of the input data set, or sorted on the key vectors in either ascending or descending direction.

Note: this procedure may take some time to complete for joins of large data sets. You should also ensure the data space is large enough for the resultant vectors, especially if using the option `REPEATS=combinations`.

Options: `NINDEX`, `METHOD`, `REPEATS`, `INCLUDE`, `SORT`.
Parameters: `LEFTVECTORS`, `RIGHTVECTORS`, `NEWVECTORS`.

Method
The `LEFTVECTORS` and `RIGHTVECTORS` are sorted by the index variables. If there are restrictions on the vectors in either input set, this is used to subset the input vectors if `INCLUDE=nonrestricted`. For each of the rows of the two sets, the keys are compared and output rows are appended according to the `METHOD` option. If repeats of the same matching key values in both sets occur and `REPEATS=combinations` the procedure loops through all combinations of the matching rows. The resulting vectors are then sorted into the order specified in the `SORT` option.

Action with `RESTRICT`
Any of the input vectors may be restricted. If `INCLUDE=nonrestricted`, only those rows which are not excluded by any restriction on vectors in each input set will be processed, otherwise the restrictions will be ignored.

See also
Directive: `EQUATE`.
Procedures: `APPEND`, `STACK`, `UNSTACK`, `VEQUATE`.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

KALMAN

Calculates estimates from the Kalman filter (A.I. Glaser).

Option

PRINT = *string tokens*	Controls printed output (xpredicted, xfiltered, deviance, residuals, gain, varpredictions, varfiltered, varresiduals); default *

Parameters

Y = *variates*, *matrices* or *pointers*	Time series data
YTRANSITIONMATRIX = *scalars*, *matrices* or *pointers*	Observation transition matrix, mapping the relationship between the current value of the state vector and the observation
YVCOVARIANCE = *scalars*, *symmetricmatrices* or *pointers*	Observation error covariance matrix
XSTATETRANSITIONMATRIX = *scalars*, *matrices* or *pointers*	State transition matrix, mapping the relationship between the current value of the state vector and its previous value
BXVCOVARIANCE = *scalars*, *matrices* or *pointers*	State noise coefficient matrix
XVCOVARIANCE = *scalars*, *symmetricmatrices* or *pointers*	State error covariance matrix
MEANINITIAL = *scalars*, *variates* or *matrices*	Initial value of the mean of the state vector
VARINITIAL = *scalars* or *symmetricmatrices*	Initial value of the variance-covariance matrix of the state vector
DEVIANCE = *scalars*	To save the deviance of the model
XPREDICTED = *matrices*	Saves the predicted (a priori) state estimate matrix
XFILTERED = *matrices*	Saves the filtered (a posteriori) state estimate matrix
RESIDUALS = *matrices*	Saves the matrix of residuals
GAIN = *pointers*	Saves the Kalman gain matrix at each iteration
VARPREDICTIONS = *pointers*	Saves the variances of the predicted state estimate matrix at each iteration
VARFILTERED = *pointers*	Saves the variances of the filtered state estimate matrix at each iteration
VARRESIDUALS= *pointers*	Saves the variances of the residuals at each iteration
SAVE = *pointers*	Save structure which provides information for use in DKALMAN

Description

KALMAN performs the iterations from the time-varying or time-invariant Kalman filter using a square-root covariance filter algorithm.

The parameters contain the components of the state space model where data (*Y*) are assumed to be linked to an unknown state vector (*X*), see *Method* for further details.

The Y parameter specifies the values of the observations as a variate, a matrix (where each row contains the values at a specific time point) or a pointer to a set of variates.

The YTRANSITIONMATRIX parameter maps the state vector (*X*) onto the observation vector (*Y*), with residuals assumed to come from a (multivariate) Normal distribution with mean of zero

and variance-covariance matrix YVCOVARIANCE.

The XSTATETRANSITIONMATRIX parameter gives the relationship between the state vector X at time t with that at time $t-1$. This is assumed to have residuals from a Normal distribution with mean zero and variance-covariance matrix specified by the BXVCOVARIANCE and XVCOVARIANCE parameters according to the equation

```
T(BXVCOVARIANCE) *+ XVCOVARIANCE *+ BXVCOVARIANCE
```

By default BXVCOVARIANCE is the identity matrix. A description of all the default combinations of these parameters is given in the *Method* Section, below.

The state-space parameters YTRANSITIONMATRIX, XSTATETRANSITIONMATRIX and BXVCOVARIANCE can be set to scalars, or to matrices, or to pointers containing scalars or matrices (if they are time-varying) where element t of the pointer contains the scalar or matrix at time t.

The covariance matrices, YVCOVARIANCE and XVCOVARIANCE, can be set to scalars or to symmetric matrices, or to pointers containing scalars or symmetric matrices (if they are time-varying) where element t of the pointer contains the scalar or symmetric matrix at time t.

You should only set XSTATETRANSITIONMATRIX, BXVCOVARIANCE and XVCOVARIANCE to scalars when the state vector (X) is of dimension one. Likewise YVCOVARIANCE should be set to a scalar only when the data vector (Y) is of dimension one, and YTRANSITIONMATRIX should be set to a scalar only when both Y and X are of dimension one.

You can simplify the input when YTRANSITIONMATRIX, YVCOVARIANCE, XSTATETRANSITIONMATRIX, BXVCOVARIANCE and/or XVCOVARIANCE are set to pointers, if there are only a few different values. You need then specify elements of the pointer only for the times when the values change (the omitted elements are assumed to be the same as the most recent previous value). For example, suppose there is a change at time 20 for the setting (say ystate) of the YTRANSITIONMATRIX parameter. You could then define ystate with elements only for times 1 and 20, as shown below.

```
POINTER [SUFFIXES=!(1,20); VALUES=matrix1,matrix20] ystate
```

(Note that you must always specify an element for time 1.)

When YTRANSITIONMATRIX, YVCOVARIANCE, XSTATETRANSITIONMATRIX, BXVCOVARIANCE or XVCOVARIANCE are set to pointers, then all elements of each pointer must be of the same type, e.g. if YTRANSITIONMATRIX is a pointer and its first element is a scalar, then all of the other elements of YTRANSITIONMATRIX must be scalars too.

Before running a Kalman filter, values must be defined for the mean and variance of the initial value of the state vector. These are supplied by the MEANINITIAL and VARINITIAL parameters respectively.

The PRINT option controls printed output, with settings:

xpredicted	predicted (a priori) state estimate matrix,
xfiltered	filtered (a posteriori) state estimate matrix,
deviance	deviance of the model,
residuals	matrix of residuals,
gain	Kalman gain matrix at each iteration,
varpredictions	variance of predicted state estimate matrix,
varfiltered	variance of filtered state estimate matrix, and
varresiduals	variance of the residuals.

By default nothing is printed.

The results can also be saved, using the parameters DEVIANCE, XPREDICTED, XFILTERED, RESIDUALS, GAIN, VARPREDICTIONS, VARFILTERED and VARRESIDUALS. The deviance is saved as a scalar. The XPREDICTED, XFILTERED and RESIDUALS parameters save matrices, where each row corresponds to an individual time point. The others parameters save pointers, suffixed from 1...n, where n is the number of time points.

The SAVE parameter saves various elements of the output for use by the DKALMAN procedure, which plots fitted and original values of the data.

Option: PRINT.
Parameters: Y, YTRANSITIONMATRIX, YVCOVARIANCE, XSTATETRANSITIONMATRIX, BXVCOVARIANCE, XVCOVARIANCE, MEANINITIAL, VARINITIAL, DEVIANCE, XPREDICTED, XFILTERED, RESIDUALS, GAIN, VARPREDICTIONS, VARFILTERED, VARRESIDUALS, SAVE.

Method
Kalman filtering is a method of analysing multi-dimensional time series which can be written in the state-space form:

$$Y_t = C_t X_t + V_t$$
$$X_t = A_t X_{t-1} + B_t W_t$$

where:

Y_t is the observed measurement vector (Y),
X_t is the state vector,
C_t is the observation transition matrix (YTRANSITIONMATRIX),
V_t is the observation error,
A_t is the state transition matrix (XSTATETRANSITIONMATRIX),
B_t is the state noise coefficient matrix (BXVCOVARIANCE) and
W_t is the state noise,

all measured at time t. When A_t, B_t and C_t are equal for all values of t, the model is assumed time invariant.

When BXVCOVARIANCE is not set it is assumed to be the identity matrix.

The observation error and state noise terms are assumed to be uncorrelated, with zero mean, and covariance matrices given by YVCOVARIANCE and XVCOVARIANCE respectively. When XVCOVARIANCE is not set, it is assumed to be the identity matrix.

The estimate of X_i given the observations Y_1 to Y_{i-1} is known as the predicted (a priori) state estimate matrix, and is usually denoted as $X_{i|i-1}$. Similarly, the estimate of X_i given the observations Y_1 to Y_i is known as the filtered (a posteriori) state estimate matrix and is usually denoted as $X_{i|i}$.

The initial values $X_{1|0}$ are assumed to be drawn from a multivariate Normal distribution with mean MEANINITIAL and variance VARINITIAL. If MEANINITIAL is not set, it is assumed to be zero. If VARINITIAL is unset then it is taken to be 100 times the identity matrix.

The NAG algorithms G13EBF (for time invariant matrices) or G13EAF (for time varying matrices) are used to update one iteration from the Kalman filter.

Action with RESTRICT
Input structures must not be restricted.

See also
Procedure: DKALMAN.
Genstat Reference Manual 1 Summary section on: Time series.

KAPLANMEIER

Calculates the Kaplan-Meier estimate of the survivor function (J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What output to print and whether to display the Kaplan-Meier estimate in a graph (estimate, mean, quantiles, summary, graph); default esti, grap
GRAPHICS = *string token*	Type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	General title for the graph; default *
WINDOW = *scalar*	Window number for the high-resolution graph; default 1
KEYWINDOW = *scalar*	Window number for the key (zero for no key); default 2
SCREEN = *string token*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
PROBABILITY = *scalar*	Probability level of the confidence interval for the Kaplan-Meier estimates; default 0.95
XLOWER = *scalar*	Lower bound for x-axis; default 0
XUPPER = *scalar*	Upper bound for x-axis; default * i.e. a value slightly larger than the maximum of the TIME parameter (or EVENT parameter if TIME is not set) is used
PLOT = *string tokens*	What additional plotting features to include (referenceline, censored); default * i.e. none
PERCENTILES = *variate* or *scalar*	Percentiles at which to estimate quantiles of survival times; default 25,50,75

Parameters

TIME = *variates*	Observed timepoints
CENSORED = *variates*	Variate specifying whether the corresponding element of TIME is censored (1) or not (0); default is to assume no censoring
GROUPS = *factors*	Factor specifying the different groups for which the survivor function is estimated
EVENT = *variates*	Saves the distinct TIME values when TIME is set; otherwise supplies an input variate specifying the endpoint of each interval
NDEATH = *variates*	Saves the number of deaths at each EVENT when TIME is set; otherwise supplies an input variate specifying the number of deaths in each interval
NATRISK = *variates*	Saves the number of units at risk at each EVENT when TIME is set; otherwise supplies an input variate with the number at risk in each interval
ESTIMATE = *variates*	Saves the Kaplan-Meier estimates of the survivor function
NEWGROUPS = *factors*	Saves the grouping of the EVENT, NDEATH, NATRISK and ESTIMATE variates when TIME is set

Description

Survival data are data in which the response variate is the lifetime of a component or the survival time of a patient. Typically these are censored, i.e. the survival time of some units is unknown at the end of the study. The survivor function F(t) is a key element in the analysis of survival

data. It is defined as the probability of an individual still surviving at time *t*. KAPLANMEIER calculates the Kaplan-Meier estimate of the survivor function for two different types of data.

The first type of data occurs when all timepoints are accurately observed. The observed timepoints or the timepoints at which censoring took place are then specified using the TIME parameter. The CENSORED variate contains values 0 and 1 to specify whether the corresponding element of TIME is censored (1) or not (0); if there was no censoring, this need not be set. The GROUPS parameter can be used to specify a factor to indicate different groups whose survivor functions are to be estimated separately. The distinct TIME values can be saved using the EVENT parameter, and the number of deaths and the number of units at risk at each individual EVENT can be saved using parameters NDEATH and NATRISK respectively. The Kaplan-Meier estimate can be saved with the ESTIMATE parameter. The NEWGROUPS parameter can save a factor indicating the group structure of the output variates.

The second type of data is relevant when the units are observed at the end of time-intervals. The exact times are then unknown and input should be specified using parameters EVENT, NDEATH, NATRISK. These specify the timepoints, number of deaths and number of risk at the end of each interval. The GROUPS parameter can again be used to request separate group estimates.

The PRINT option selects the output to be displayed with settings:

estimate	the events, number of deaths, number of units at risk and the Kaplan-Meier estimate with a confidence interval,
summary	summary of censored and uncensored observations,
quantiles	estimates quantiles of the distribution of survival times (observed timepoints only),
mean	mean and standard error (observed timepoints only),
graph	plots the Kaplan-Meier estimate against the time points.

The default is PRINT=estimates,graph.

The probability level for the Kaplan-Meier estimate confidence interval can be set using the PROBABILITY option; by default this is 0.95. Percentiles for estimating survival times can be set using the PERCENTILES option; by default this is 25,50,75. If PRINT=graph is set, then the PLOT option can be used to include censored observations and a reference line at $S(t)$=0.5 to indicate the median survival time. If GRAPHICS=highresolution different lines are drawn for different groups, whereas GRAPHICS=lineprinter produces separate graphs for the different groups. Lower and upper bounds for the x-axis can be set by options XLOWER and XUPPER, the TITLE option can specify a title for the plots. Options WINDOW and KEYWINDOW control the windows used for high-resolution graphs.

Options: PRINT, GRAPHICS, TITLE, WINDOW, KEYWINDOW, SCREEN, PROBABILITY, XLOWER, XUPPER, PLOT, PERCENTILES.
Parameters: TIME, CENSORED, GROUPS, EVENT, NDEATH, NATRISK, ESTIMATE, NEWGROUPS.

Method

When TIME is set, the Kaplan-Meier estimate is calculated according to equation (1.10) in Kalbfleisch & Prentice (1980). When TIME is not set, the Kaplan-Meier estimate is directly calculated from the variates specified by EVENT, NDEATH and NATRISK. If PERCENTILES includes the median (50) then a confidence interval is displayed for the median using the method described in Brookmeyer & Crowley (1982). The mean survival time is calculated by the formula

$$\mu = \sum_{i=1\ldots k} \{ S(t_{i-1}) \times (t_i - t_{i-1}) \}$$

where

k is the number of ordered death times,

$S(t_{i-1})$ is the Kaplan-Meier estimate of the survivor function at the $(i-1)^{th}$ death time,

t_i is the death time, where t_0 is defined to be zero

Its standard error is calculated using the formula:

$$se(\mu) = \sqrt{[\ (m/m-1)\ x\ \sum_{i=1...k-1}\ \{\ (A_i\ **\ (2/n_i))\ \times\ (n_i\ -\ d_i)\ \}\ ]}$$

where

$$m = \sum_{i=1...k}\ \{\ d_i\ \}$$

$$A_i = \sum_{j=1...k-1}\ \{\ S(t_{j-1})\ \times\ (t_{j+1}\ -\ t_j)\ \}$$

Action with `RESTRICT`

The input variates and factor `GROUPS` may be restricted identically. The Kaplan-Meier estimate is based only on the units not excluded by the restriction.

Reference

Brookmeyer, R. & Crowley, J. (1982). A confidence interval for the median survival time. *Biometrics*, **38**, 29-41.

Collett, D. (1994). *Modelling Survival Data in Medical Research*. Chapman & Hall. London.

Kalbfleisch, J.D. & Prentice, R.L. (1980). *The Statistical Analysis of Failure Time Data*. Wiley, New York.

See also

Procedures: `RLIFETABLE`, `RPHFIT`, `RPROPORTIONAL`, `RSTEST`, `RSURVIVAL`.

Genstat Reference Manual 1 Summary section on: Survival analysis.

KAPPA

Calculates a kappa coefficient of agreement for nominally scaled data (A.J. Rook).

Option

PRINT = *string token*	Whether to print kappa and its associated information (test); default test

Parameters

DATA = *tables*	Data sets, each consisting of an object × category table whose entries are the number of judges assigning the *i*th object to the *j*th category
STATISTIC = *scalars*	Save the value of kappa for each data table
VARIANCE = *scalars*	Save the corresponding variances

Description

The kappa coefficient provides a way of assessing the agreement between judges who have rated a set of *N* objects or subjects using a nominal scale: that is, each judge has allocated each object to one of *M* different categories. The data for KAPPA, specified by the DATA parameter, consists of an $N \times M$ table whose entries indicate the number of judges that have assigned the *i*th object to the *j*th category. This must not contain any missing values and all the row totals must be equal.

Kappa takes the value one when there is complete agreement and zero when there is none (except that expected by chance). The printing of the test statistic and its associated information is controlled by the PRINT option. With the default, test, the procedure prints the actual and expected proportion of times that the judges agree, the resulting value of kappa and its variance. When *N* is large, the sampling distribution of kappa is approximately Normal. The procedure thus also prints the value of kappa divided by the variance, and its probability assuming a Normal distribution. A warning is printed if *N* is less than 20.

The STATISTIC and VARIANCE parameters allow kappa and its variance to be saved, in scalars.

Option: PRINT. Parameters: DATA, STATISTIC, VARIANCE.

Method

The method used is that of Siegel & Castellan (1988, pages 284-291).

Reference

Siegel, S. & Castellan, N. J. (1988). *Nonparametric Statistics for the Behavioural Sciences (second edition)*. McGraw-Hill, Singapore.

See also

Procedure: GSTATISTIC.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

KCONCORDANCE

Calculates Kendall's Coefficient of Concordance; synonym CONCORD (S.J. Welham, N.M. Maclaren & H.R. Simpson).

Options

PRINT = *string tokens*	Output required (test, ranks): test produces the relevant test statistics, ranks produces the vector of mean ranks and the ranks for each sample; default test
GROUPS = *factor*	Defines the variable stored in each unit if only one variate is specified by DATA
STATISTIC = *scalar*	Scalar to save the coefficient of concordance
CHISQUARE = *scalar*	Scalar to save the chi-square approximation to the coefficient (calculated only if the sample size is at least 8)
MEANRANKS = *variate*	Variate to save the mean ranks for individuals over variables
DF = *scalar*	Scalar to save the degrees of freedom for CHISQUARE

Parameters

DATA = *variates*	List of variables to be compared, or a single variate containing the data for all the variables (the GROUPS option must then be set to indicate the variable recorded in each unit belongs)
RANKS = *variates*	Save the ranks of the variables

Description

Kendall's Coefficient of Concordance is a measure of association between K rankings on N individuals, i.e. a set of N individuals are ranked on each of K variables in turn, and these rankings are to be compared. The variables can be stored in separate variates and the DATA parameter set to list them all. Alternatively, all the data can be stored in a single variate, and the GROUPS option set to a factor to indicate which variable is recorded in each unit of the variate. (KCONCORDANCE then assumes that the individuals are recorded in the same order for each variable.)

Concord calculates the chi-square approximation to the statistic if the sample sizes are large enough (i.e. 8 or more). Otherwise, for $2<K<21$ and $2<N<8$, KCONCORDANCE looks up the probability from a stored table. The results of these calculations can be printed using the test setting of PRINT, or saved using the options STATISTIC (for the coefficient), CHISQUARE (for the chi-square statistic) and DF (degrees of freedom). The ranks setting of PRINT causes the vector of mean ranks (over all variates) and the ranks for each variate individually to be displayed, and these can be saved using the MEANRANKS option and the RANKS parameter.

Options: PRINT, GROUPS, STATISTIC, CHISQUARE, MEANRANKS, DF.
Parameters: DATA, RANKS.

Method

Kendall's Coefficient of Concordance, KC, is built up from the sum of ranks over the K variables for each individual, R_j ; j=1...N :

$$KC = \text{sum}\{ (R_j - R)\times(R_j - R) ; j=1...N \} / \{ K\times K\times N\times(N\times N - 1)/12 \}$$

where R is the mean of the set $\{ R_j ; j=1...N \}$.

If ties are present in the data, then the denominator of KC must be modified to avoid bias in the statistic. The adjusted denominator is:

$\{ K \times N \times (N \times N - 1)/12 - K \times \text{sum}\{ T_j ; j=1...N \} \}$

where T_j = is the sum over all ranks k in group j of $((t_k^3) - t_k)/12$, and t_k is the number of observations in the group with rank k. (See e.g. Siegel 1956, pages 229-238.)

The chi-square approximation for this statistic (valid only when $N \geq 8$) is $K \times (N-1) \times KC$ with $N-1$ degrees of freedom.

Action with `RESTRICT`

If any of the variates in `DATA` is restricted, the statistic is calculated only for the units not excluded by the restriction.

Reference

Siegel S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Procedures: `CMHTEST`, `FCORRELATION`, `KTAU`, `LCONCORDANCE`, `SPEARMAN`.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

KCROSSVALIDATION

Computes cross validation statistics for punctual kriging (D.A. Murray & R. Webster).

Options

PRINT = *string tokens*	Controls printed output (statistics, correlation); default stat
PLOT = *string token*	Whether to produce a scatter plot of the predicted against the true values (scatter); default * i.e. none
Y = *variate* or *scalar*	Y positions or interval (not needed for 2D regular data i.e. when DATA is a matrix)
X = *variate*	X positions (needed only for 2D irregular data)
YOUTER = *variate*	Variate containing 2 values to define the Y-bounds of the region to be examined (bottom then top); by default the whole region is used
XOUTER = *variate*	Variate containing 2 values to define the X-bounds of the region to be examined (bottom then top); by default the whole region is used
RADIUS = *scalar*	Maximum distance between target point and usable data
SEARCH = *string token*	Type of search (isotropic, anisotropic); default isot
MINPOINTS = *scalar*	Minimum number of data points from which to compute elements; default 7
MAXPOINTS = *scalar*	Maximum number of data points from which to compute elements; default 20
DRIFT = *string token*	Amount of drift (constant, linear, quadratic); default cons
YXRATIO = *scalar*	Ratio of Y interval to X interval
SAVE = *pointer*	Pointer containing model estimates saved from MVARIOGRAM

Parameters

DATA = *variates* or *matrices*	Observed measurements as a variate or, for data on a regular grid, as a matrix
ISOTROPY = *string tokens*	Form of variogram (isotropic, Burgess, geometrical); default isot
MODELTYPE = *string tokens*	Model fitted to the variogram (power, boundedlinear, circular, spherical, doublespherical, pentaspherical, exponential, besselk1, gaussian, cubic, stable, cardinalsine, matern); default *
NUGGET = *scalars*	The nugget variance
SILLVARIANCES = *scalars* or *variates*	Sill variances of the spatially dependent component
RANGES = *scalar* or *variates*	Ranges of the spatially dependent component
GRADIENT = *scalars* or *variates*	Slope of the unbounded component
EXPONENT = *scalars* or *variates*	Power of the unbounded component or power for the stable model
SMOOTHNESS = *scalar*	Value of ν parameter for the Matern model
PHI = *scalars* or *variates*	Phi parameters in anisotropic model (ISOTROPY = burg or geom)
RMAX = *scalars* or *variates*	Maximum gradient of an anisotropic model

RMIN = *scalars* or *variates*	Minimum gradient of an anisotropic model
MEASUREMENTERROR = *scalars*	Variance of measurement error
PREDICTIONS = *variates* or *matrices*	Saves the kriged estimates in matrices for 2D Regular data, otherwise in variates
VARIANCES = *variates* or *matrices*	Saves the estimation variances in matrices for 2D Regular data, otherwise in variates
STATISTICS = *variates*	Saves the cross validation statistics

Description

In geostatistics one way of choosing between plausible models for variograms is to use them for kriging, and see how well the kriging predicts the true values. The observed value of *z* at each sampling point in the data is omitted in turn from the whole set and predicted from the others. The predictions are compared with the true values to give a mean deviation or error, and the kriging variances are compared with the squared deviations to give a mean squared deviation ratio. This process is known as "cross-validation". The procedure KCROSSVALIDATION uses this principle of leave-one-out cross-validation.

The data are supplied, by the DATA parameter, in one of the two forms as for the KRIGE directive: i.e. in a matrix for data on a regular grid, or as a variate for irregularly scattered data together with the X and Y options set to variates to supply the spatial coordinates.

By default all data are considered when forming the kriging system. However, you may select a subset of the data by limiting the area to a rectangle defined by XOUTER and YOUTER options. Each of these should be set to a variate with two values to define lower and upper limits in the x (East-West) and y (North-South) directions respectively.

The minimum and maximum number of points for the kriging system are set by the MINPOINTS and MAXPOINTS options. There is a minimum limit of 3 for MINPOINTS and a maximum of 40 for MAXPOINTS, and MINPOINTS must be less than or equal to MAXPOINTS. The defaults are 7 and 20 respectively. You may select data points around the point to be kriged by setting the RADIUS option to the radius within which they must lie. If the variogram is anisotropic, the search may be requested to be anisotropic by setting option SEARCH to anisotropic; by default SEARCH=isotropic.

You can invoke universal kriging for two-dimensional data by setting the DRIFT option to linear or to quadratic, i.e. to be of order 1 or 2 respectively. The default is DRIFT=constant, to give ordinary kriging. For data in a regular grid that is not square, the ratio of the spacing in the *y* direction to that in the *x* direction should be given by the YXRATIO option. The default is 1.0 (i.e. square).

The variogram is specified by its type and parameters, as follows. The MODEL option may be defined to be set to either power, boundedlinear (one dimension only), circular, spherical, doublespherical, pentaspherical, exponential, besselk1 (Whittle's function), gaussian, cubic, stable (i.e. powered exponential; see Webster & Oliver 2001), cardinalsine or matern. All models may have a nugget variance, supplied using the NUGGET option; this is the constant estimated by MVARIOGRAM. You can specify the variance of any measurement error using the MEASUREMENTERROR parameter. The parameters of the power function (the only unbounded model) are defined by the GRADIENT and EXPONENT parameters. The parameter for the power of the stable model is supplied using the EXPONENT parameter. The parameter ν for the Matern model is supplied using the SMOOTHNESS parameter. The simple bounded models (i.e. all other settings of MODEL except doublespherical) require the SILLVARIANCES (the sill of the correlated variance) and RANGES parameters. The latter is strictly the correlation range of the boundedlinear, circular, spherical and pentaspherical models, while for the asymptotic models it is the distance parameter of the model. The doublespherical model requires SILLVARIANCES and RANGES to be set to

variates of length two, to correspond to the two components of the model.

The ISOTROPY parameter allows the variation to be defined to be either isotropic or anisotropic in one of two ways: either Burgess anisotropy (Burgess & Webster 1980) or geometric anisotropy (Webster & Oliver 1990). The anisotropy is specified by three parameters, namely PHI the angle in radians of the direction of maximum variation, RMAX the maximum gradient of the model, and RMIN the minimum gradient. In the current release only the power function may be anisotropic.

The predictions (or estimates) and variances can be saved using the PREDICTIONS and VARIANCES parameters. The cross-validation statistics can be saved using the STATISTICS parameter.

The PRINT option can be set to statistics to print the cross validation statistics or correlation to print the correlation between the predicted and true values. The PLOT option can be used to produce a plot of the predicted values against the true values.

Options: PRINT, PLOT, Y, X, YOUTER, XOUTER, RADIUS, SEARCH, MINPOINTS, MAXPOINTS, DRIFT, YXRATIO, SAVE.

Parameters: DATA, ISOTROPY, MODEL, NUGGET, SILLVARIANCES, RANGES, GRADIENT, EXPONENT, SMOOTHNESS, PHI, RMAX, RMIN, MEASUREMENTERROR, PREDICTIONS, VARIANCES, STATISTICS.

Method

The mean error is given by

$$\sum_{i=1\ldots n} \{ z(x_i) - zhat(x_i) \} / n$$

the mean squared error is

$$\sum_{i=1\ldots n} \{ z(x_i) - zhat(x_i) \}^2 / n$$

and the mean squared deviation ratio

$$\sum_{i=1\ldots n} \{ (z(x\_i) - zhat(x_i))^2 / sig^2(x_i) \} / n$$

Action with RESTRICT

The vectors involved in the analysis may be restricted as for KRIGE.

References

Burgess, T.M. & Webster, R. (1980). Optimal interpolation and isarithmic mapping of soil properties. I. The semi-variogram and punctual kriging. *Journal of Soil Science*, **31**, 315-331.

Webster, R. & Oliver, M.A. (1990). *Statistical Methods in Soil and Land Resource Survey*. Oxford University Press, Oxford.

Webster, R. & Oliver, M.A. (2001). *Geostatistics for Environmental Scientists*. Wiley, Chichester.

See also

Directives: FVARIOGRAM, FCOVARIOGRAM, KRIGE, MCOVARIOGRAM, COKRIGE.

Procedures: MVARIOGRAM, DVARIOGRAM, DCOVARIOGRAM, DHSCATTERGRAM.

Genstat Reference Manual 1 Summary section on: Spatial statistics.

KCSRENVELOPES

Simulates K function bounds under complete spatial randomness (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string tokens*	What to print (summary, monitoring); default summ, moni

Parameters

YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NPOINTS = *scalars*	How many points to generate in each simulation; no default – this parameter must be set
NSIMULATIONS = *scalars*	How many simulations of CSR to use; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
KLOWER = *variates*	Variates to receive the values of the lower bound of the K function
KUPPER = *variates*	Variates to receive the values of the upper bound of the K function
SEED = *scalars*	Seeds for the random numbers used in the simulations; default 0

Description

The K function, or reduced second-order moment function, relates to the distribution of the inter-event distances between all ordered pairs of events in a spatial point pattern (see Diggle 1983). The term complete spatial randomness (CSR) is used to represent the hypothesis that the overall density of events in a spatial point pattern is constant throughout the study region, and that the events are distributed independently and uniformly.

The K function for a completely random pattern is given by

$$K(s) = \pi \times s^2 .$$

(The K function for a clustered (regular) pattern will tend to be larger (smaller) than the values given by the above expression, at least for small distances.) The procedure KHAT can be used to obtain an approximately unbiased estimate of K(*s*) for an observed pattern which can be compared with the expected value under CSR given by the above expression. However, the variance of the estimate under the null hypothesis cannot be expressed in closed form, and so critical values for the estimated K function cannot be obtained analytically. This problem can be overcome by repeatedly simulating from the null hypothesis and estimating the K function for each simulated pattern. If NSIMULATIONS denotes the number of simulations used, then, for each value of *s*, the minimum (maximum) value of the estimated K function provides an approximate 100/(NSIMULATIONS+1) percent lower (upper) critical value for K(*s*).

The procedure KCSRENVELOPES computes lower and upper bounds (envelopes) for the K function under CSR. The data required by the procedure are the coordinates of a polygon in which to simulate CSR (specified by the parameters XPOLYGON and YPOLYGON), the number of points to generate in each simulation (specified using the parameter NPOINTS), the number of simulations to use (specified by the parameter NSIMULATIONS) and a vector of distances at which to calculate the EDF of K (specified by the parameter S). The SEED parameter allows a seed to be supplied for generating the random numbers for the simulations (thereby producing

reproducible results). If this is not supplied, the default of 0 initializes the random number generator (if necessary) from the system clock. The output of the procedure consists of two vectors, the first containing the minimum value obtained for K(*s*) for each distance in S, and the second containing the corresponding maximum values. The minimum and maximum values of the K function can be saved using the parameters KLOWER and KUPPER.

Printed output is controlled using the PRINT option. The settings available are monitoring (which prints a message to mark the start of each simulation) and summary (which prints the distances at which the K function is estimated under the heading S, together with the lower and upper bounds for the K function under the headings KLOWER and KUPPER).

Option: PRINT.
Parameters: YPOLYGON, XPOLYGON, NPOINTS, NSIMULATIONS, S, KLOWER, KUPPER, SEED.

Method
A procedure PTCHECKXY is used to check that XPOLYGON and YPOLYGON have identical restrictions. The SORT function is then used to create a variate containing the distances in S arranged in ascending order. (The original variate is left unchanged.) The procedures GRCSR and KHAT are called NSIMULATIONS times to calculate estimates of the K function under CSR. Finally, the VMINIMA and VMAXIMA functions are used to calculate the minimum and maximum values of the K function for each distance in S.

Action with RESTRICT
If XPOLYGON and YPOLYGON are restricted, only the subset of values specified by the restriction will be included in the calculations. The parameter S may also be restricted.

Reference
Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

See also
Procedures: FHAT, GHAT, KHAT, KLABENVELOPES, KSED, KSTHAT, KSTSE, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

KERNELDENSITY

Uses kernel density estimation to estimate the underlying density of a sample (P.W. Goedhart).

Options

PRINT = *string token*	What to print (integral, summary, monitoring, graph); default inte
METHOD = *string token*	Which automatic bandwidth selection method should be used when the BANDWIDTH option is not set (s1, s2, s3, sj); default sj
BANDWIDTH = *scalar* or *variate*	Which bandwidth value or values are to be used; default *
NGRIDEXPONENT2 = *scalar*	Defines the number of grid points as 2**NGRIDEXPONENT2; default 11
SAVEGRIDEXTENT = *scalar*	Defines the lower and upper limit of the interval on which the kernel density is saved; the default value of 4 uses the full interval on which the kernel density is calculated
NFOURIER = *scalar*	Defines the upper limit of the sample size for which the kernel density is calculated directly (when the sample size exceeds the setting of this option, the fast Fourier transform is used to calculate the kernel density); default 100
PROPORTION = *variate*	Proportions at which to calculate quantiles of the kernel density estimate; default !(0.025, 0.25, 0.5, 0.75, 0.975)
PLOT = *string tokens*	Specifies the graphs to be plotted (kerneldensity, histogram, sample); default kern, hist, samp
TITLE = *text*	General title(s) for the graph(s); default *
WINDOW = *scalar* or *variate*	Window number(s) for the graph(s); default 1
SCREEN = *string token*	Whether to clear the screen before plotting into the first window, or whether to or continue plotting on the old screen (clear, keep); default clea

Parameters

SAMPLE = *variates*	The sample for which to calculate the kernel density estimate
GRID = *variates*	Saves the grid of equidistant points at which the kernel density is calculated
DENSITY = *variates* or *pointers*	Saves the kernel density estimate
CUMULATIVE = *variates* or *pointers*	Saves the estimated cumulative distribution
QUANTILE = *variates* or *pointers*	Saves the quantiles calculated from the estimated cumulative distribution
SAVEBANDWIDTH = *scalars*	Saves the automatically selected bandwidths as specified by the METHOD option

Description

Kernel density estimation is a useful tool for exploring the unknown underlying distribution of a sample, see e.g. Silverman (1986) for a general introduction in density estimation. The kernel method constructs an estimate $f_h(t)$ of the true density function by placing a kernel function $K(t; x_i, h)$ over each observation x_i in the sample. The kernel function $K(t; x, h)$ is itself a density

function with location parameter x and scale parameter h, also called bandwidth in this context. The density estimate is then given by

$$f_h(t) = (1 / (n \times h)) \times \sum_{i=1\ldots n} K((t - x_i) / h) \quad (1)$$

where n denotes the sample size. It turns out that the choice of kernel function K is not very critical for the resulting estimate $f_h(t)$, see Section 3.3 of Silverman (1986). The Gaussian kernel is commonly used and is therefore adopted here as kernel function, i.e.

$$K(\mathrm{t}) = (1 / \sqrt{(2 \times \pi)}) \times \exp((-t^2) / 2) \quad (2)$$

For this choice of kernel function K, there is an efficient algorithm available for the calculation of $f_h(t)$. This algorithm employs the fast Fourier transform of the data.

The choice of bandwidth h is of crucial importance in kernel density estimation. A large value of h will give rise to an oversmoothed density estimate, while a small value of h will produce a very ragged density with many spikes at the observations. Silverman (1986) recommends examining kernel density estimates for several values of h, since this will highlight different features of the data. For automatic use of kernel density estimation, estimation of the bandwidth h from the data is very helpful. Silverman (1986) suggests the following normal-based estimates:

S1 = 1.06 × (standard deviation) × $n^{-1/5}$

S2 = 0.79 × (interquartile range) × $n^{-1/5}$

S3 = 0.90 × minimum(standard deviation, interquartile range/1.34) × $n^{-1/5}$

These estimates are popular due to their simplicity. Jones, Marron & Sheather (1996), who provide an extensive review of the many automatic methods for choosing the bandwidth, advise against these estimates. They recommend the method of Sheather & Jones (1991) for general purposes. This method, denoted below by SJ, is therefore the default method used in the `KERNELDENSITY` procedure.

The sample, for which to estimate the underlying density, must be specified by means of the `SAMPLE` parameter. The `METHOD` and `BANDWIDTH` options determine which bandwidths h are used. When the `BANDWIDTH` option is set to a scalar or variate, then these values are used for the bandwidth h. When the `BANDWIDTH` option is unset, the `METHOD` option determines which automatic bandwidth selection method is used. The default setting of the `METHOD` option is `sj`, which indicates that the method of Sheather & Jones (1991) is to be used. The automatically selected bandwidth can be saved by means of the `SAVEBANDWIDTH` parameter.

The kernel density estimate is calculated on an interval at a grid of equidistant points. The grid is returned using the `GRID` parameter, and the density estimate and corresponding cumulative density can be saved with the `DENSITY` and `CUMULATIVE` parameters. When the `BANDWIDTH` option is set to a variate, the `DENSITY` and `CUMULATIVE` parameters are pointers to variates: one variate for each bandwidth value. The number of grid points can be set using the `NGRIDEXPONENT2` option as `2**NGRIDEXPONENT2`. The lower and upper limit of the interval on which the kernel density is calculated are given by:

```
CALCULATE lower = MINIMUM(SAMPLE) - 4*MAXIMUM(BANDWIDTH)
CALCULATE upper = MAXIMUM(SAMPLE) + 4*MAXIMUM(BANDWIDTH)
```

This ensures that the integral of the kernel density will be very close to one. The `SAVEGRIDEXTENT` option can be used to save the grid and the (cumulative) density at a more limited interval defined by

```
CALCULATE lowsave = MINIMUM(SAMPLE) \
                    - SAVEGRIDEXTENT*MAXIMUM(BANDWIDTH)
CALCULATE uppsave = MAXIMUM(SAMPLE) \
                    + SAVEGRIDEXTENT*MAXIMUM(BANDWIDTH)
```

The setting of the `NFOURIER` option determines whether the kernel density is calculated directly by means of equation (1) or by employing the fast Fourier transform of the data. When the sample size n exceeds the setting of the `NFOURIER` option, the fast Fourier transform is used.

The parameter `QUANTILES` can be used to save quantiles of the kernel density estimate, for proportions specified by means of the `PROPORTION` option. When the `BANDWIDTH` option is set

to a variate, the QUANTILES are saved in a pointer containing a set of variates.

The PRINT option controls the output displayed by KERNELDENSITY. The integral setting prints the integral of the kernel density, which should be close to one, while the summary setting print summary statistics of the sample and of the kernel density estimate. The monitoring setting can be used to monitor the iterative bandwidth estimation method SJ. Finally, the setting graph produces a high-resolution plots of the kernel densities, superimposed over a rough histogram estimate of the density calculated as the proportion of the sample falling into CEILING(SQRT(number of samples))+1 equal intervals across the range of sample values. (There will be as many plots as there were bandwidths.) The sample values are also plotted, using the symbol +, along the bottom of the plots. The PLOT option controls which elements (kerneldensity, histogram, sample) are plotted. The TITLE option can provide a title for each graph. The WINDOW option specifies the windows to be used for the plots (default 1), and the SCREEN option controls whether or not the screen is cleared before plotting into the first window (default clear).

Options: PRINT, METHOD, BANDWIDTH, NGRIDEXPONENT2, SAVEGRIDEXTENT, NFOURIER, PROPORTION, PLOT, TITLE, WINDOW, SCREEN.
Parameters: SAMPLE, GRID, DENSITY, CUMULATIVE, QUANTILE, SAVEBANDWIDTH.

Method

The interquartile range is calculated by means of the TABULATE directive. For sample sizes larger than the setting of the NFOURIER option, the fast Fourier transform of the data is used. This employs the algorithm of Silverman (1982), with the modification of Jones & Lotwick (1984), using the FOURIER directive. The cumulative density is calculated by applying the trapezodial rule to the density. Quantiles of the estimated distribution are calculated with the INTERPOLATE directive applied to the cumulative distribution. The difference between the cumulative distribution calculated directly and by means of the Fourier transform, was found to be less than 0.0001 for samples of size 100 from a wide variety of mixed distributions.

The bandwidth selection method of Sheather & Jones (1991) requires solving of a complicated equation by means of an iterative method. The iterative method stops when the relative difference between subsequent estimates of the bandwidth is smaller than 0.0001. The implementation in Genstat is a transcription of a Fortran subroutine which was kindly made available by Jones (1991). The algorithm uses all squared pairwise differences of the sample, i.e. $(x_i - x_j)^2$ for all i, j. When the sample size is large there are too many such differences. The sampled values are then discretizised on a grid; i.e. all the sampled values are assigned to a binning interval and the expectation of this value is used instead of the exact value $(x_i - x_j)^2$. Assuming a uniform distribution of x_i and x_j over their respective binning intervals, the expectation is given by $d^2 (k^2 + 1/6)$, where d is the length of a binning interval and k is the number of intervals in between x_i and x_j. The data are discretizised when the sample size exceeds 500. The grid is refined in a loop until there are more than 500 bins with sampled values assigned to them. The relative difference between the Sheather & and Jones estimate calculated with exact squared pairwise differences and by discretization of the data, was found to be less than 0.0003 for samples of size 100 from a wide variety of mixed distributions.

The automatic bandwidth selection method of Sheather & Jones (1991) is implemented in a subsidiary procedure _KERNELSJ which is called by KERNELDENSITY. This has the following 5 options:

PRINT = *string token*	What to print (monitoring); default *
NTIMES = *scalar*	Maximum number of iterations; default 20
TOLERANCE = *scalar*	Convergence criterion; default 0.0001
BINSAMPLE = *scalar*	Defines the upper limit of the sample size for which the method uses exact squared pairwise differences of the

	sampled value (for sample sizes exceeding the setting of this option, discretization of the sample is used); default 500
NGRID = *scalar*	Defines the number of bins for discretization of the sampled values – the grid is refined in a loop until there are more than NGRID bins with sampled values assigned to them; default 500

It also has three parameters which must all be set

SAMPLE = *variates*	The sample for which to estimate the bandwidth
IQR = *scalar*	Interquartile range of the sample
SJ = *scalars*	Saves the estimated bandwidth using Sheather & Jones (1991)

Action with RESTRICT

The SAMPLE parameter can be restricted. The grid and kernel density estimate are then calculated using only those units that are in the restriction set.

References

Jones, M.C. & Lotwick, H.W. (1984). A remark on algorithm AS 176. Kernel density estimation using the fast Fourier transform. Remark AS R50. *Applied Statistics*, 33, 120-122.

Jones, M.C. (1991). Fortran subroutine SJEQD for the automatic bandwidth selection method of Sheather & Jones (1991). Personal communication.

Jones, M.C., Marron, J.S. & Sheather, S.J. (1996). Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, 11, 337-381.

Sheather, S.J. & Jones, M.C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B*, 53, 683-690.

Silverman, B.W. (1982). Kernel density estimation using the fast Fourier transform. Applied Statistics Algorithm AS 176. *Applied Statistics*, 31, 93-99.

Silverman, B.W. (1986). *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London.

See also

Directive: DISTRIBUTION.

Procedures: MSEKERNEL2D, PTKERNEL2D, PTK3D.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

KHAT

Calculates an estimate of the K function (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

`PRINT` = *string token*	What to print (`summary`); default `summ`

Parameters

`Y` = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
`X` = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
`YPOLYGON` = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
`XPOLYGON` = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
`S` = *variates*	Vectors of distances to use; no default – this parameter must be set
`KVALUES` = *variates*	Variates to receive the estimated values of the K function

Description

The K function, or reduced second-order moment function, relates to the distribution of the inter-event distances between all ordered pairs of events in a spatial point pattern (see Diggle 1983). The function is formally defined as the expected number of further events within distance s of an arbitrary event, divided by the overall density of events per unit area. An approximately unbiased estimator of K which incorporates corrections for edge effects can be obtained using the method of Ripley (1976). This estimator, denoted by $\hat{K}(s)$, is essentially an empirical distribution function (EDF) of weighted inter-event distances. The weight associated with an inter-event distance s derived from events at positions (x_1, y_1) and (x_2, y_2) is the reciprocal of the conditional probability that any event separated from the point (x_1, y_1) by the distance s will fall in the study region and so be observed.

The term complete spatial randomness (CSR) is used to represent the hypothesis that the overall density of events in a spatial point pattern is constant throughout the study region, and that the events are distributed independently and uniformly. Under CSR, the expected number of further events which lie within a distance s of an arbitrary event in the pattern is simply the area of a circle of radius s, multiplied by the overall density of events. Thus, the K function for a completely random pattern is given by

$$K(s) = \pi \times s^2 .$$

The K function for a clustered (regular) pattern will tend to be larger (smaller) than the values given by the above expression, at least for small distances.

The procedure KHAT calculates Ripley's (1976) estimator for K given the coordinates of a spatial point pattern (specified by the parameters `X` and `Y`), the coordinates of a polygon containing the points (specified by the parameters `XPOLYGON` and `YPOLYGON`) and a vector of distances at which to calculate the EDF of K (specified by the parameter `S`). The output of the procedure is a vector of estimates of K corresponding to the distances in `S`. The estimated K function can be saved using the parameter `KVALUES`.

Printed output is controlled using the `PRINT` option. The default setting of `summary` prints the distances at which the K function is estimated and the estimates themselves under the headings `S` and `KVALUES`.

Option: PRINT.
Parameters: Y, X, YPOLYGON, XPOLYGON, S, KVALUES.

Method
A procedure PTCHECKXY is called to check that X and Y have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The procedure then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. The SORT function is then used to create a variate containing the distances in S arranged in ascending order. (The original variate is left unchanged.) The procedure then calls a procedure PTPASS to call a Fortran program to calculate an edge-corrected estimate of the K function.

Action with RESTRICT
The variates X, Y, XPOLYGON, YPOLYGON and S may be restricted, as long as X has the same restriction as Y, and XPOLYGON has the same restriction as YPOLYGON. Only the subset of values specified by each restriction will be included in the calculations.

References
Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.
Ripley, B.D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, **13**, 255-266.

See also
Procedures: FHAT, GHAT, KCSRENVELOPES, KLABENVELOPES, KSED, KSTHAT, KSTSE, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

KLABENVELOPES

Gives bounds for K function differences under random labelling (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string tokens*	What to print (summary, monitoring); default summ, moni

Parameters

Y1 = *variates*	Vertical coordinates of the first spatial point patterns; no default – this parameter must be set
X1 = *variates*	Horizontal coordinates of the first spatial point patterns; no default – this parameter must be set
Y2 = *variates*	Vertical coordinates of the second spatial point patterns; no default – this parameter must be set
X2 = *variates*	Horizontal coordinates of the second spatial point patterns; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NSIMULATIONS = *scalars*	How many simulations of random labelling to use; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
KLOWER = *variates*	Variates to receive the values of the lower bound of the difference between the K functions
KUPPER = *variates*	Variates to receive the values of the upper bound of the difference between the K functions
SEED = *scalars*	Seeds for the random numbers used to generate the random labellings; default 0

Description

The K function, or reduced second-order moment function, relates to the distribution of the inter-event distances between all ordered pairs of events in a spatial point pattern (see Diggle 1983). The procedure KHAT can be used to obtain an approximately unbiased estimator of K(s) for an observed pattern, and this may be used to investigate the degree of clustering/regularity in the pattern. Patterns consisting of two different types of events may be separated into two patterns, one for each type of event. The difference between the K functions for the two univariate patterns may then be used to investigate whether the two types of events display similar degrees of clustering/regularity. (If the difference between the K functions is positive (negative) then the first pattern is more (less) strongly clustered than the second.)

The term random labelling is used to represent the hypothesis that the spatial distributions of different types of events within an overall pattern are completely random. Under random labelling, the difference between the K functions for different types of events is zero (Diggle & Chetwynd 1991). Critical values for the estimated difference between two K functions under random labelling may be obtained by repeatedly simulating from the null hypothesis, for example using the procedure GRLABEL. If NSIMULATIONS denotes the number of simulations used, then, for each value of s, the mimimum (maximum) value of the difference between the two K functions provides an approximate 100/(NSIMULATIONS+1) percent lower (upper) critical value for the true difference.

The procedure KLABENVELOPES computes lower and upper bounds (envelopes) for the difference between two K functions under random labelling. The data required by the procedure are the coordinates of two spatial point patterns (specified by the parameters X1, Y1, X2 and Y2), the coordinates of a polygon containing the points (specified by the parameters XPOLYGON and YPOLYGON), the number of simulations to use (specified by the parameter NSIMULATIONS) and a vector of distances at which to estimate the K functions (specified by the parameter S). The SEED parameter allows a seed to be supplied for generating the random numbers required to generate the random labelling (thereby producing reproducible results). If this is not supplied, the default of 0 initializes the random number generator (if necessary) from the system clock. The output of the procedure consists of two vectors, the first containing the minimum value obtained for the difference between the K functions for each distance in S (calculated by subtracting the K function for the second pattern from that of the first pattern), and the second containing the corresponding maximum values. The minimum and maximum values of the difference between the two K functions can be saved using the parameters KLOWER and KUPPER.

Printed output is controlled using the PRINT option. The settings available are monitoring (which prints a message to mark the start of each simulation) and summary (which prints the distances at which the K functions are estimated under the heading S, together with the lower and upper bounds for the difference between the K functions under the headings KLOWER and KUPPER).

Option: PRINT.
Parameters: Y1, X1, Y2, X2, YPOLYGON, XPOLYGON, NSIMULATIONS, S, KLOWER, KUPPER, SEED.

Method

A procedure PTCHECKXY is called to check that X1 and Y1 have identical restrictions. Similar checks are made on X2 and Y2, and XPOLYGON and YPOLYGON. The SORT function is then used to create a variate containing the distances in S arranged in ascending order. (The original variate is left unchanged.) The procedures GRLABEL and KHAT are then called NSIMULATIONS times to calculate estimates of the difference between the K functions for the two types of events under random labelling. Finally the VMINIMA and VMAXIMA functions are used to calculate the minimum and maximum values of the difference between the two K functions for each distance in S.

Action with RESTRICT

The variates X1, Y1, X2, Y2, XPOLYGON, YPOLYGON and S may be restricted as long as X1 has the same restriction as Y1, X2 has the same restriction as Y2 and XPOLYGON has the same restriction as YPOLYGON. Only the subset of values specified by each restriction will be included in the calculations.

References

Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

Diggle, P.J. & Chetwynd, A.G. (1991). Second-order analysis of spatial clustering. *Biometrics*, **47**, 1155-1163.

See also

Procedures: FHAT, GHAT, KCSRENVELOPES, KHAT, KSED, KSTHAT, KSTSE, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

KNEARESTNEIGHBOURS

Classifies items or predicts their responses by examining their *k* nearest neighbours (R.W. Payne).

Options

PRINT = *string tokens*	Printed output required (neighbours, predictions); default pred
SIMILARITY = *matrix* or *symmetric matrix*	Provides the similarities between the training and prediction sets of items
NEIGHBOURS = *pointer*	Pointer with a variate for each prediction item to save the numbers of its nearest neighbours in the training set
GROUPS = *factor*	Defines groupings to identify the training and prediction sets of items when SIMILARITY is a symmetric matrix
LEVTRAINING = *scalar* or *text*	Identifies the level of GROUPS or dimension of SIMILARITY that represents the training set; default 1
LEVPREDICTION = *scalar* or *text*	Identifies the level of GROUPS or dimension of SIMILARITY that represents the prediction set; default 2
METHOD = *string token*	How to calculate the prediction from a DATA variate (mean, median); default medi
MINSIMILARITY = *scalar*	Cut-off minimum value of the similarity for items to be regarded as neighbours; default 0.75
MINNEIGHBOURS = *scalar*	Minimum number of nearest neighbours to use; default 5
MAXNEIGHBOURS = *scalar*	Maximum number of nearest neighbours to use; default 10
SEED = *scalar*	Seed for the random numbers used to select neighbours when more than MAXNEIGHBOURS are available; default 0

Parameters

DATA = *variates* or *factors*	Data values for the items in the training set
PREDICTIONS = *variates* or *factors*	Saves the predictions

Description

KNEARESTNEIGHBOURS provides the data-mining technique known as *k-nearest-neighbour* classification. This allocates unknown items to a category, or it predicts their (continuous) responses, by looking at nearby items in a known data set. The known data set is usually called the *training set*, and we will call the unknown items the *prediction set*.

The SIMILARITY option provides a similarity matrix for KNEARESTNEIGHBOURS to use to determine the nearby items in the training set (or *nearest neighbours*) for each item in the prediction set. This can be a symmetric matrix with a row (and column) for every item in the combined set of training and prediction items. The GROUPS option must then be set to a factor with one level for the training items and another for the prediction items. By default the training set has level 1 and the prediction set has level 2, but these can be changed by the LEVTRAINING and LEVPREDICTION options. Matrices like these can be formed in a wide variety of ways, using mixtures of categorical and continuous data, by the FSIMILARITY directive. For example, if we have a factor Sex, and variates Age, Weight and Height whose values are known for both the training and prediction items, we could form a symmetric matrix Sim by

```
FSIMILARITY [SIMILARITY=Sim] Sex,Age,Weight,Height;\
            TEST=simplematching,3(euclidean)
```

However, Sim will contain unnecessary information, as we need the similarities between prediction and training items, but not between training items or between prediction items. So, for large data sets, it will be more efficient to form a (rectangular) between-group similarity matrix by setting the GROUPS option of FSIMILARITY. For example

```
FSIMILARITY [SIMILARITY=Gsim; GROUPS=Gfac] Sex,Age,Weight,Height;\
            TEST=simplematching,3(euclidean)
```

where Gfac is a factor with two levels, one for the training set (usually level 1), and the other for the prediction set (usually level 2). You then no longer need to set the GROUPS option of KNEARESTNEIGHBOUR. The LEVTRAINING and LEVPREDICTION options now specify the dimension of the similarity matrix (1 for rows, and 2 for columns) that correspond to the training and prediction data sets, respectively. (They still correspond to group levels though, as they are defined by the numbers of the respective levels of the GROUPS factor in FSIMILARITY.)

The MINSIMILARITY option sets a minimum value on the similarity between two items if they are to be regarded as neighbours (default 0.75). The MINNEIGHBOURS option specifies the minimum number of neighbours to try to find (default 5), and the MAXNEIGHBOURS option specifies the maximum number (default 10). The search for nearest neighbours for a particular prediction item works by finding the most similar item in the training set, and adding this (with any equally-similar training items) to the set of neighbours. If at least MINNEIGHBOURS have been found, the search stops. Otherwise it finds the next most similar items, and adds these to the set of neighbours, continuing until at least MINNEIGHBOURS have been found. If this results in more than MAXNEIGHBOURS neighbours, KNEARESTNEIGHBOURS makes a random selection from those that are least similar to the prediction item, so that the number of neighbours becomes MAXNEIGHBOURS. The SEED option specifies the seed for the random numbers that are used to make that selection. The default of zero continues an existing sequence of random numbers if any have already been used in this Genstat job, or initializes the seed automatically. The NEIGHBOURS option can save a pointer, containing variate for each prediction item storing the numbers of its neighbours within the training set.

Once the neighbours have been found, KNEARESTNEIGHBOURS can use these to form the predictions. The DATA parameter lists variates and/or factors containing values of the variables of interest for the items on the training set. The predictions can be saved using the PREDICTIONS parameter (in variates and/or factors to match the settings of the DATA parameter).

For a DATA factor, the category predicted for each item in the prediction set is taken to be the factor level that occurs most often amongst its nearest neighbours. If more than one level occurs most often, the choice is narrowed down by seeing which of the levels has the the most similar neighbours. If this still leaves more than one level, the choice is narrowed further by seeing which of the levels has neighbours with the highest mean similarity. Then, if even that does not lead to a single level, the final choice is made at random.

For a DATA variate, the METHOD option controls whether the prediction is made by the median (default) or the mean of the data values of the nearest neighbours of each prediction item.

Printed output is controlled by the PRINT option, with settings:

neighbours	to print the nearest neighbours, and
predictions	to print the predictions.

The default if PRINT=predictions.

So, to print predictions of blood pressure with a variate of training data Pressure, using the similarity matrix Gsim (as above) and default settings for the numbers of neighbours, we simply need to put

```
KNEARESTNEIGHBOURS [SIMILARITY=Gsim] Pressure
```

Options: PRINT, SIMILARITY, NEIGHBOURS, GROUPS, LEVTRAINING, LEVPREDICTION, METHOD, MINSIMILARITY, MINNEIGHBOURS, MAXNEIGHBOURS, SEED.
Parameters: DATA, PREDICTIONS.

See also
Directives: FSIMILARITY, ASRULES, NNFIT, RBFIT.
Procedures: BCLASSIFICATION, BCFOREST, BREGRESSION, SOM.
Genstat Reference Manual 1 Summary sections on: Data mining, Multivariate and cluster analysis.

KOLMOG2

Performs a Kolmogorov-Smirnoff two-sample test (S.J. Welham, N.M. Maclaren & H.R. Simpson).

Options

PRINT = *string tokens*	Output required (test, differences, ranks): test gives the test statistic, differences gives signed differences, and ranks produces the ranks for each sample; default test
GROUPS = *factor*	Defines the groups for a two-sample test if only the Y1 parameter is specified

Parameters

Y1 = *variates*	Identifier of the variate holding the first sample
Y2 = *variates*	Identifier of the variate holding the second sample
R1 = *variates*	Saves the ranks of the first sample
R2 = *variates*	Saves the ranks of the second sample
STATISTIC = *scalars*	Scalar to save the test statistic (the maximum absolute difference between the cumulative distribution functions)
CHISQUARE = *scalars*	Scalar to save the chi-square approximation to the test statistic
DIFFERENCES = *variates*	Variate to save the signed differences between the cumulative distribution functions

Description

The Kolmogorov-Smirnoff test assesses the similarity between the underlying distributions of the two samples, by comparing their cumulative distribution functions; the test statistic is the maximum absolute difference between the cumulative distribution functions. The samples can either be specified in two separate variates using the parameters Y1 and Y2. Alternatively, they can be given in a single variate, with the GROUPS option set to a factor to identify the samples. The GROUPS option is ignored when the Y2 parameter is set.

Output from the procedure is controlled by the PRINT option: test prints the relevant test statistic, differences prints the signed differences, and ranks prints a vector of ranks for each of the samples.

The test statistic and its chi-square approximation can be saved using the parameters STATISTIC and CHISQUARE respectively. The parameter DIFFERENCES can be used to save the differences between the cumulative distributions. The R1 and R2 parameters allow the ranks of the samples to be saved.

Options: PRINT, GROUPS.
Parameters: Y1, Y2, R1, R2, STATISTIC, CHISQUARE, DIFFERENCES.

Method

The Kolmogorov-Smirnoff two sample test is a test of the null hypothesis that the two samples arise from the same distribution, against the alternative that the underlying distributions are different. The test compares the two empirical cumulative distribution functions in order to try and detect differences in shape of the underlying distributions. The cumulative distribution functions S_1 and S_2 are formed by

$$S_k(X) = (\text{ number of scores in sample } k \le X) / (\text{ size of sample } k)$$

for k=1,2; and a suitable set of points X. The procedure uses the set of values taken by one or

other of the samples, i.e. {*X*: *X* is in DATA}. The maximum absolute difference

$$MD = \max(\text{abs} \{ S_1(X) - S_2(X) \})$$

is used as the basis for significance tests. The chi-square approximation (2 degrees of freedom) to this statistic is *CH*:

$$CH = 4 \times MD \times MD \times (n_1 \times n_2 / (n_1 + n_2))$$

where n_1, n_2 are the sizes of the samples. (See for example Siegel 1956, pages 127-136.)

Action with RESTRICT

The variates Y1 and Y2 can be restricted, and in different ways. KOLMOG2 uses only those units of each variate that are not excluded by their respective restrictions. Restrictions are also obeyed on Y1 and GROUPS, allowing RESTRICT to be used for example to limit the data to only two groups when the GROUPS factor has more than two levels.

Reference

Siegel, S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Directive: DISTRIBUTION.

Procedures: DPROBABILITY, EDFTEST.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

KRUSKAL

Carries out a Kruskal-Wallis one-way analysis of variance (S.J. Welham, N.M. Maclaren & H.R. Simpson).

Options

PRINT = *string tokens*	Output required (test, ranks): test produces the relevant test statistics, ranks produces a vector of ranks for each sample relative to the whole data set; default test
GROUPS = *factor*	Defines the sample membership if only one variate is specified by DATA
STATISTIC = *scalar*	Scalar to save the Kruskal-Wallis test statistic
MEANRANKS = *variate*	Variate to save the mean ranks of the samples
DF = *scalar*	Scalar to save the degrees of freedom for the statistic

Parameters

DATA = *variates*	List of variates containing the data for each sample, or a single variate containing the data from all the samples (the GROUPS option must then be set to indicate the sample to which each unit belongs)
RANKS = *variates*	Allow the ranks to be saved (relative to the combined data)

Description

KRUSKAL carries out a Kruskal-Wallis one-way analysis of variance on the ranks (relative to the whole data set) of a set of samples. The samples can be stored in different variates and supplied as a list in the DATA pointer. Alternatively, they can all be placed in a single variate, and the GROUPS option set to a factor to indicate the sample to which each unit belongs. Output from the procedure is controlled by the PRINT option: test (the default setting) prints the relevant test statistics, and ranks prints the vector of ranks for each sample.

The test statistic, vector of mean ranks and degrees of freedom can be saved using the STATISTIC, MEANRANKS and DF options, respectively. Parameter RANKS can be set to a variate, or variates, to store the ranks of the data relative to the whole data set.

Options: PRINT, GROUPS, STATISTIC, MEANRANKS, DF.
Parameters: DATA, RANKS.

Method

The Kruskal-Wallis One-Way Analysis of Variance is used to test the hypothesis that several (K) samples come from distributions with the same mean. The test statistic H, is formed by ranking the combined data set, then considering the sum of these ranks within each sample:

$$H = [\ (12 / N\times(N+1)) \times \sum_{j=1\ldots K} \{\ R_j\times R_j/n_j\ \}\ ] - 3\times(N+1)$$

where R_j is the sum of ranks for the jth sample,
n_j is the size of the jth sample, and
N is the size of the combined data set.

If ties are present in the data, then an adjustment to the statistic H is required:

$$\text{adjusted } H = H\ /(\ 1 - \sum_k \{\ t_k^3 - t_k\ \}/(N^3 - N)\ )$$

where t_k is the number of observations with rank k. (See for example Siegel 1956, pages 184-193.)

When there are at least five cases in each of the samples, H has approximately a Chi-square distribution on $K-1$ degrees of freedom. When this condition is not satisfied, and there are three

samples, KRUSKAL uses a table of calculated values of the distribution of the statistic.

Action with RESTRICT

The variates in DATA can be restricted, and in different ways. KRUSKAL uses only those units of each variate that are not excluded by their respective restrictions.

Reference

Siegel, S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Procedures: AONEWAY, APERMTEST, A2WAY, FRIEDMAN.

Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Analysis of variance.

KSED

Calculates the standard error for K function differences under random labelling (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	Controls printed output (summary); default summ

Parameters

Y1 = *variates*	Vertical coordinates of the first spatial point patterns; no default – this parameter must be set
X1 = *variates*	Horizontal coordinates of the first spatial point patterns; no default – this parameter must be set
Y2 = *variates*	Vertical coordinates of the second spatial point patterns; no default – this parameter must be set
X2 = *variates*	Horizontal coordinates of the second spatial point patterns; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of the polygons; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of the polygons; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
KSED = *variates*	Variates to receive the values of the standard error of the difference between the K functions for the two patterns under random labelling
VCOVARIANCE = *symmetric matrices*	Saves the variance-covariance matrix
VK1 = *variates*	Saves the variance of Khat for first spatial point pattern
VK2 = *variates*	Saves the variance of Khat for second spatial point pattern
VK12 = *variates*	Saves the covariance of Khat for the two samples

Description

The K function, or reduced second-order moment function, relates to the distribution of the inter-event distances between all ordered pairs of events in a spatial point pattern (see Diggle 1983). The procedure KHAT can be used to obtain an approximately unbiased estimator of K(s) for an observed pattern, and this may be used to investigate the degree of clustering/regularity in the pattern. Patterns consisting of two different types of events may be separated into two patterns, one for each type of event. The difference between the K functions for the two univariate patterns may then be used to investigate whether the two types of events display similar degrees of clustering/regularity. (If the difference between the K functions is positive (negative) then the first pattern is more (less) strongly clustered than the second.)

The term random labelling is used to represent the hypothesis that the spatial distributions of different types of events within an overall pattern are completely random. The expected value of the difference between two K functions under random labelling is zero. The standard error of the estimated difference can be obtained using the method of Diggle & Chetwynd (1991).

The procedure KSED calculates the standard error for the difference between two K functions under random labelling. The data required by the procedure are the coordinates of two spatial point patterns (specified by parameters X1, Y1, X2 and Y2), the coordinates of a polygon containing the points (specified by the parameters XPOLYGON and YPOLYGON) and a vector of distances at which to estimate the K functions (specified by the parameter S). The standard error

for the difference between the two K functions for each distance in S can be saved using the parameter KSED. The VCOVARIANCE parameter can be used to save the variance-covariance matrix for the difference between the K functions for the two patterns. The variances for the two K functions can be saved using the VK11 and VK12 parameters. The covariance for the K function of the two point patterns can be saved using the VK12 parameter.

Printed output is controlled using the PRINT option. The default setting of summary prints the distances at which the standard error is calculated and the values of the standard error under the headings S and KSED. The variance and covariance for the two K functions are also displayed under the headings VK11, VK22 and VK12.

Option: PRINT.
Parameters: Y1, X1, Y2, X2, YPOLYGON, XPOLYGON, S, KSED, VCOVARIANCE, VK1, VK2, VK12.

Method
A procedure PTCHECKXY is called to check that X1 and Y1 have identical restrictions. Similar checks are made on X2 and Y2, and on XPOLYGON and YPOLYGON. The procedure then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. The SORT function is then used to create a variate containing the distances in S arranged in ascending order. (The original variate is left unchanged.) The procedure then calls APPEND to combine the horizontal coordinates for both patterns, and again to combine the vertical coordinates. The coordinates of the closed polygon, the sorted values of S and the combined coordinates for the two patterns are then passed to the Fortran program using a procedure PTPASS. This program calculates the variance-covariance matrix for the difference between the K functions for the two patterns and the variance of the K function for each sample. Finally, the standard error for the difference between the two K functions is obtained using the CALCULATE directive by taking the square root of the values on the diagonal of the variance-covariance matrix.

Action with RESTRICT
The variates X1, Y1, X2, Y2, XPOLYGON, YPOLYGON and S may be restricted as long as X1 has the same restriction as Y1, X2 has the same restriction as Y2, and XPOLYGON has the same restriction as YPOLYGON. Only the subset of values specified by each restriction will be included in the calculations.

References
Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.
Diggle, P.J. & Chetwynd, A.G. (1991). Second-order analysis of spatial clustering. *Biometrics*, **47**, 1155-63.

See also
Procedures: FHAT, GHAT, KHAT, KSTHAT, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

KSTHAT

Calculates an estimate of the K function in space, time and space-time (D.A. Murray, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	Controls printed output (summary); default summ

Parameters

Y = *variates*	Vertical coordinates of the spatial point patterns; no default – this parameter must be set
X = *variates*	Horizontal coordinates of the spatial point patterns; no default – this parameter must be set
TIMES = *variates*	Times for each event
YPOLYGON = *variates*	Vertical coordinates of the polygons; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of the polygons; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
TVALUES = *variates*	Time scales for the analysis
TLOWER = *variates*	Lower temporal domain
TUPPER = *variates*	Upper temporal domain
KS = *variates*	Saves the spatial K function estimates
KT = *variates*	Saves the spatial K function estimates
KST = *variates*	Saves the space-time K function estimates

Description

For data that consist of locations and times of events within a specified spatial region and time-period, it is often of interest to examine whether events that are relatively close in space are also relatively close in time. Data that have events both close in space and time are said to exhibit space-time clustering. KSTHAT provides a method for describing this space-time interaction using an extension of the second-order methods for purely spatial point patterns to the spatial-temporal setting. KSTHAT calculates an estimate of the second-order reduced moment measure, or K function, in space, time and space-time. The K function, or reduced second-order moment function, relates to the distribution of the inter-event distances between all ordered pairs of events in a spatial point pattern (see Diggle 1983). The function is formally defined as the expected number of further events within distance s of an arbitrary event, divided by the overall density of events per unit area. The space-time K function is defined as the number of further events occurring within distance s and time t of an arbitary event, divided by the expected number of events per unit space per unit time (see Diggle et al 1995). The K function for a spatial-temporal homogeneous Poisson process, in which the spatial and temporal components are independent homogeneous Poisson processes is given by

$$K(s,t) = 2\pi \times s^2 t.$$

This represents the volumne of a cylinder with base radius s and height 2t. Assuming that the spatial and temporal component processes are independent the space-time K function factorizes as follows

$$K(s,t) = K1(s) \times K2(t)$$

where K1(s) is the spatial K function and K2(t) is the temporal K function.

The procedure KSTHAT calculates space-time K given the coordinates of a spatial point pattern (specified by the parameters X and Y), and the times for each of the events (specified by TIMES). The coordinates of a polygon containing the spatial points are specified by the parameters

XPOLYGON and YPOLYGON, and the parameter S is used to supply the vector of distances at which to calculate the spatial K function. The TLOWER and TUPPER parameters specify the start and finish of the temporal range. The TVALUES parameter is used to supply the vector of times at which to calculate the temporal K function. The outputs of the procedure are vectors of estimates of the spatial and temporal K function corresponding to the distances in S and times in TIMES. The estimated spatial and temporal K functions can be saved using the parameters KS and KT respectively. The KST parameter can be used to save a matrix of the space-time K function.

Printed output is controlled using the PRINT option. The default setting of summary prints the distances at which the spatial K function is estimated along with the estimates, the times at which the temporal K function is estimated along with the estimates and the space-time K function estimates.

Option: PRINT.
Parameters: Y, X, TIMES, YPOLYGON, XPOLYGON, S, TVALUES, TLOWER, TUPPER, KS, KT, KST.

Method

A procedure PTCHECKXY is called to check that X, Y and TIMES have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The procedure then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. The SORT function is then used to create variates containing the distances in S and time in TVALUES arranged into ascending order. (The original variates are left unchanged.) The procedure then calls a procedure PTPASS to call a Fortran program to calculate an estimate of the space-time K function.

References

Diggle, P,J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic press, London.

Diggle, P.J., Chetwynd, A.G., Haggkvist, R. & Morris, S.E. (1995). Second-order analysis of space-time clustering. *Statistical Methods in Medical Research*, **4**, 124-136.

See also

Procedures: FHAT, GHAT, KHAT, KSTMCTEST, KSTSE, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

KSTMCTEST

Performs a Monte-Carlo test for space-time interaction (D.A. Murray, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string token*	Controls printed output (statistic, rank); default stat, rank
PLOT = *string token*	Whether to produce a plot of the test statistic (histogram); default hist
NTIMES = *scalar*	Number of simulations for Monte-Carlo test; default 49
SEED = *scalar*	Seed for random number generator; default 0 continues from previous generation or uses system clock

Parameters

Y = *variates*	Vertical coordinates of the first spatial point patterns; no default – this parameter must be set
X = *variates*	Horizontal coordinates of the first spatial point patterns; no default – this parameter must be set
TIMES = *variates*	Times for each event
YPOLYGON = *variates*	Vertical coordinates of the polygons; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of the polygons; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
TVALUES = *variates*	Time scales for the analysis
TLOWER = *variates*	Lower temporal domain
TUPPER = *variates*	Upper temporal domain
STATISTIC = *scalars*	Saves the Monte-Carlo statistic

Description

For data that consist of locations and times of events within a specified spatial region and time-period, it is often of interest to examine whether events that are relatively close in space are also relatively close in time. Data that have events both close in space and time are said to exhibit space-time clustering. KSTMCTEST performs a Monte-Carlo test for space-time interaction using a sum of the residuals as a test statistic. For a given number of simulations the procedure randomly permutes the times of set of points, and computes the sum of differences between the space-time K function and the product of the spatial and temporal K functions. The first simulation represents the observed value of the test statistic.

The data required by the procedure are the coordinates of a spatial point pattern (specified by the parameters X and Y), and the times for each of the events (specified by TIMES). The coordinates of a polygon containing the spatial points are specified by the parameters XPOLYGON and YPOLYGON, and the S parameter is used to supply the vector of distances at which to calculate the spatial K function. The TLOWER and TUPPER parameters specify the start and finish of the temporal range. The TVALUES parameter is used to supply the vector of times at which to calculate the temporal K function. The output of the procedure are vectors of estimates of the spatial and temporal K function corresponding to the distances in S and times in TIMES. The values of the test statistic can be saved using the STATISTIC parameter.

The NTIMES option allows you to specify the number of simulations and the SEED option allows you to set a randomization seed. By default SEED=0, so the random numbers will continue any existing sequence, used earlier in the Genstat program. The setting

PLOT=histogram can be used to produce a histogram of the test statistics with the value for the data indicated with a vertical line.

Printed output is controlled using the PRINT option. The default setting of summary prints the test statistic for the Monte-Carlo test, and rank prints how the test statistic for the data ranks with the simulations.

Options: PRINT, PLOT, NSIM, SEED.
Parameters: Y, X, TIMES, YPOLYGON, XPOLYGON, S, TVALUES, TLOWER, TUPPER, STATISTIC.

Method

A procedure PTCHECKXY is called to check that X, Y and TIMES have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The procedure then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. For each simulation the procedure randomly permutes the times of the set of points and then calls a procedure PTPASS to call a Fortran program to calculate an estimate of the space-time K functions. The difference between the space-time K function and the product of the spatial and temporal K functions is then calculated.

Action with RESTRICT

The variates X, Y, TIMES, XPOLYGON, YPOLYGON, S and TVALUES may be restricted, as long as X, Y and TIMES have the same restriction, and XPOLYGON has the same restriction as YPOLYGON. Only the subset of values specified by each restriction will be included in the calculations.

References

Diggle, P,J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

Diggle, P.J., Chetwynd, A.G., Haggkvist, R. & Morris, S.E. (1995). Second-order analysis of space-time clustering. *Statistical Methods in Medical Research*, **4**, 124-136.

See also

Procedures: FHAT, GHAT, KHAT, KSTHAT, KSTSE, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

KSTSE

Calculates the standard error for the space-time K function (D.A. Murray, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	Controls printed output (summary); default summ

Parameters

Y = *variates*	Vertical coordinates of the spatial point patterns; no default – this parameter must be set
X = *variates*	Horizontal coordinates of the spatial point patterns; no default – this parameter must be set
TIMES = *variates*	Times for each event
YPOLYGON = *variates*	Vertical coordinates of the polygons; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of the polygons; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
TVALUES = *variates*	Time scales for the analysis
TLOWER = *variates*	Lower temporal domain
TUPPER = *variates*	Upper temporal domain
SE = *variates*	Saves the standard errors

Description

For data that consist of locations and times of events within a specified spatial region and time-period, it is often of interest to examine whether events that are relatively close in space are also relatively close in time. Data that have events both close in space and time are said to exhibit space-time clustering. KSTSE calculates the standard error for the space-time K function.

The data required by the procedure are the coordinates of a spatial point pattern (specified by the parameters X and Y), and the times for each of the events (specified by TIMES). The coordinates of a polygon containing the spatial points are specified by the parameters XPOLYGON and YPOLYGON, and the parameter S is used to supply the vector of distances at which to calculate the spatial K function. The TLOWER and TUPPER parameters specify the start and finish of the temporal range. The TVALUES parameter is used to supply the vector of times at which to calculate the temporal K function. The output of the procedure are vectors of estimates of the spatial and temporal K function corresponding to the distances in S and times in TIMES. The standard errors of the space-time interaction can be saved using the SE parameter.

Printed output is controlled using the PRINT option. The default setting of summary prints the standard errors for the space-time clustering.

Option: PRINT.
Parameters: Y, X, TIMES, YPOLYGON, XPOLYGON, S, TVALUES, TLOWER, TUPPER, SE.

Method

A procedure PTCHECKXY is called to check that X, Y and TIMES have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The procedure then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. The SORT function is then used to create variates containing the distances in S and time in TVALUES arranged into ascending order. (The original variates are left unchanged.) The procedure then

calls a procedure PTPASS to call a Fortran program to calculate an estimate of the space-time clustering standard errors.

References

Diggle, P,J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic press, London.

Diggle, P.J., Chetwynd, A.G., Haggkvist, R. & Morris, S.E. (1995). Second-order analysis of space-time clustering. *Statistical Methods in Medical Research*, **4**, 124-136.

See also

Procedures: FHAT, GHAT, KHAT, KSTMCTEST, KSTHAT, K12HAT.

Genstat Reference Manual 1 Summary section on: Spatial statistics.

KTAU

Calculates Kendall's rank correlation coefficient τ (R.W. Payne & D.B. Baird).

Options

PRINT = *string tokens*	Output required (correlations, probabilities); default corr, prob
GROUPS = factor	Defines the sample membership if only one variate is specified by DATA
CORRELATIONS = *scalar* or *symmetric matrix*	Scalar to save the rank correlation coefficient if there are two samples, or symmetric matrix to save the coefficients between all pairs of samples if there are several
PROBABILITIES = *scalar* or *symmetric matrix*	Scalar to save the probability for the correlation coefficient if there are two samples, or symmetric matrix to save the probabilities for all pairs of samples if there are several
NORMAL = *scalar* or *symmetric matrix*	Scalar to save a transformation of tau that approximately follows a Normal distribution with mean zero and variance if there are two samples, or symmetric matrix to save the transformation for all pairs of samples if there are several

Parameter

DATA = *variates*	List of variates containing the data for each sample, or a single variate containing the data from all the samples (the GROUPS option must then be set to indicate the sample to which each unit belongs)

Description

KTAU calculates Kendall's rank correlation coefficient (known as τ i.e. tau) between pairs of samples. The samples can be stored in different variates and supplied in a list with the DATA parameter. Alternatively, they can all be placed in a single variate, and the GROUPS option set to a factor to indicate the sample to which each unit belongs.

The PRINT option controls the printed output, with settings:

correlations	to print the correlations between the samples; and
probabilities	to print the corresponding probabilities (calculated under the assumption, or null hypothesis, that there is no association between the samples).

By default these are both printed.

The CORRELATIONS option allows the correlations to be saved, in a scalar if there are only two samples or in a symmetric matrix if there are three or more. Similarly, the probabilities can be saved using the PROBABILITIES option. Also, you can use the NORMAL option to save a transformation of τ that approximately follows a Normal distribution with mean zero and variance; this provides reasonably accurate probabilities when the number of units *N* is no smaller than 8 (see Kendall 1948).

Options: PRINT, GROUPS, CORRELATIONS, PROBABILITIES, NORMAL.
Parameter: DATA.

Method

Kendall's rank correlation coefficient τ is a measure of association between the rankings of two variables measured on N individuals. It is calculated as

$$\tau = S / \sqrt{(NC_1 \times NC_2)}$$

S is defined as the sum of

$$\texttt{SIGN}(x_i - x_j) \times \texttt{SIGN}(y_i - y_j)$$

over all pair of distinct units i and j. NC_1 and NC_2 are the number of valid comparisons (removing ties and missing values) that can be made amongst the first and second set of samples, respectively. (See Siegel 1956, pages 213-223.)

The transformation of τ into a Normal random variable is given by

$$\tau / \sqrt{((2 \times (2 \times N + 5)) / (9 \times N \times (N - 1)))}$$

The probabilities are calculated using procedure PRKTAU.

Action with RESTRICT

If any of the variates in DATA is restricted, the statistic is calculated only for the set of units not excluded by the restriction.

References

Kendall, M.G. (1948). *Rank Correlation Methods*. Griffin, London.

Siegel, S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Procedures: PRKTAU, CMHTEST, FCORRELATION, KCONCORDANCE, LCONCORDANCE, SPEARMAN.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

KTORENVELOPES

Gives bounds for the bivariate K function under independence (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string tokens*	What to print (summary, monitoring); default summ, moni

Parameters

Y1 = *variates*	Vertical coordinates of the first spatial point patterns; no default – this parameter must be set
X1 = *variates*	Horizontal coordinates of the first spatial point patterns; no default – this parameter must be set
Y2 = *variates*	Vertical coordinates of the second spatial point patterns; no default – this parameter must be set
X2 = *variates*	Horizontal coordinates of the second spatial point patterns; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NSIMULATIONS = *scalars*	How many simulations of independence to use; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
KLOWER = *variates*	Variates to receive the values of the lower bound of the bivariate K function
KUPPER = *variates*	Variates to receive the values of the upper bound of the bivariate K function
SEED = *scalars*	Seeds for the random numbers used to generate the random shifts; default 0

Description

The bivariate K function, or reduced second-order moment function, relates to the distribution of inter-event distances in a spatial point pattern consisting of two different types of events (see Diggle 1983). For independent processes, the bivariate K function is given by

$$K_{12}(s) = \pi \times s^2 .$$

(The bivariate K function for positively (negatively) correlated processes will tend to be larger (smaller) than the values given by the above expression, at least for small distances). The procedure K12HAT can be used to obtain an approximately unbiased estimate of $K_{12}(s)$ for two observed patterns which can be compared with the expected value under independence given by the above expression. However, the variance of the estimate under the null hypothesis cannot be expressed in closed form, and so critical values for the estimated K function cannot be obtained analytically. This problem can be overcome by repeatedly simulating from the null hypothesis and estimating the K function for each simulation. If NSIMULATIONS denotes the number of simulations used, then, for each value of *s*, the minimum (maximum) value of the estimated K function provides an approximate 100/(NSIMULATIONS+1) percent lower (upper) critical value for $K_{12}(s)$.

The established method for simulating independent patterns whilst retaining the observed degree of clustering/regularity in the univariate patterns is to perform a random toroidal shift of one observed pattern whilst holding the other fixed. This method is due to Lotwick and

Silverman (1982). Random toroidal shifts can be performed using the procedure GRTORSHIFT.

The procedure KTORENVELOPES computes lower and upper bounds (envelopes) for the bivariate K function under independence. The data required by the procedure are the coordinates of two spatial point patterns (specified by the parameters X1, Y1, X2 and Y2), the coordinates of a polygon containing the points (specified by the parameters XPOLYGON and YPOLYGON), the number of simulations to use (specified by the parameter NSIMULATIONS) and a vector of distances at which to estimate the K function (specified by the parameter S). The simulations of independence are generated by performing random toroidal shifts of the second pattern whilst holding the first pattern fixed. The SEED parameter allows a seed to be supplied for generating the random numbers required to generate the random shifts (thereby producing reproducible results). If this is not supplied, the default of 0 initializes the random number generator (if necessary) from the system clock. The output of the procedure consists of two vectors, the first containing the minimum value obtained for $K_{12}(s)$ for each distance in S, and the second containing the corresponding maximum values. The minimum and maximum values of the bivariate K function can be saved using the parameters KLOWER and KUPPER.

Printed output is controlled using the PRINT option. The settings available are monitoring (which prints a message to mark the start of each simulation) and summary (which prints the distances at which the K function is estimated under the heading S, together with the lower and upper bounds for the K function under the headings KLOWER and KUPPER).

Option: PRINT.
Parameters: Y1, X1, Y2, X2, YPOLYGON, XPOLYGON, NSIMULATIONS, S, KLOWER, KUPPER, SEED.

Method

A procedure PTCHECKXY is called to check that X1 and Y1 have identical restrictions. Similar checks are made on X2 and Y2, and XPOLYGON and YPOLYGON. The SORT function is then used to create a variate containing the distances in S arranged in ascending order. (The original variate is left unchanged). The procedure PTBOX is used to generate a bounding box for the polygon specified by XPOLYGON and YPOLYGON. The procedures GRTORSHIFT and K12HAT are then called NSIMULATIONS times to calculate estimates of the bivariate K function under independence (using the bounding box as the toroidal region for the random shifts of the second pattern). Finally, the VMINIMA and VMAXIMA functions are used to calculate the minimum and maximum values of the bivariate K function for each distance in S.

Action with RESTRICT

The variates X1, Y1, X2, Y2, XPOLYGON, YPOLYGON and S may be restricted as long as X1 has the same restriction as Y1, X2 has the same restriction as Y2, and XPOLYGON has the same restriction as YPOLYGON. Only the subset of values specified by each restriction will be included in the calculations.

References

Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

Lotwick, H.W. & Silverman, B.W. (1982). Methods for analysing spatial processes of several types of points. *Journal of the Royal Statistical Society, Series B*, **44**, 406-413.

See also

Procedures: FHAT, GHAT, KHAT, KSTHAT, K12HAT.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

K12HAT

Calculates an estimate of the bivariate K function (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

Y1 = *variates*	Vertical coordinates of the first spatial point patterns; no default – this parameter must be set
X1 = *variates*	Horizontal coordinates of the first spatial point patterns; no default – this parameter must be set
Y2 = *variates*	Vertical coordinates of the second spatial point patterns; no default – this parameter must be set
X2 = *variates*	Horizontal coordinates of the second spatial point patterns; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
S = *variates*	Vectors of distances to use; no default – this parameter must be set
KVALUES = *variates*	Variates to receive the estimated values of the bivariate K functions

Description

The bivariate K function, or reduced second-order moment function, relates to the distribution of inter-event distances in a spatial point pattern consisting of two different types of events (see Diggle 1983). Suppose that the two types of events are classified as type j (j = 1, 2), then there are four bivariate K functions, $K_{ij}(s)$ (i, j = 1, 2), each defined as the expected number of type j events within distance s of an arbitrary type i event, divided by the overall density of type j events per unit area. Each $K_{ij}(s)$ corresponds exactly to the function $K(s)$ for a univariate process, and can be used to investigate whether an observed pattern is random, clustered or regular (see the procedure KHAT). The two remaining functions, $K_{12}(s)$ and $K_{21}(s)$, can be used to investigate whether the spatial patterns of the two types of events are independent or positively / negatively correlated.

Assuming that the bivariate process is stationary and isotropic (meaning that its properties are invariant under translations and rotations of the coordinate space), then $K_{12}(s) = K_{21}(s)$. An approximately unbiased estimator for $K_{12}(s)$ which incorporates corrections for edge effects can, therefore, be obtained by taking a weighted sum of separate estimators for $K_{12}(s)$ and $K_{21}(s)$ (these being analogous to the edge-corrected estimator for the K function of a univariate process – see the procedure KHAT). The final estimator for $K_{12}(s)$, due to Lotwick & Silverman (1982), is given by

$$\hat{K}_{12}(s) = (n_2 \times \tilde{K}_{12}(s) + n_1 \times \tilde{K}_{21}(s)) / (n_1 + n_2),$$

where $\tilde{K}_{12}(s)$ and $\tilde{K}_{21}(s)$ are the separate estimators for $K_{12}(s)$ and $K_{21}(s)$, and n_j (j = 1, 2) is the number of type j events.

For independent processes, the expected number of type j events which lie within a distance s of an arbitrary type i event is simply the area of a circle of radius s, multiplied by the overall density of events. Thus, the bivariate K function for independent processes is given by

$$K_{12}(s) = \pi \times (s^2).$$

The bivariate K function for positively (negatively) correlated processes will tend to be larger

(smaller) than the values given by the above expression, at least for small distances.

The procedure K12HAT calculates Lotwick & Silverman's (1982) estimator for $K_{12}(s)$ given the coordinates of two spatial point patterns (specified by the parameters X1, Y1, X2 and Y2), the coordinates of a polygon containing the points (specified by the parameters XPOLYGON and YPOLYGON) and a vector of distances (specified by the parameter S). The output of the procedure is a vector of estimates of $K_{12}(s)$ corresponding to the distances in S. The estimated bivariate K function can be saved using the parameter KVALUES.

Printed output is controlled using the PRINT option. The default setting of summary prints the distances at which the bivariate K function is estimated and the estimates themselves under the headings S and KVALUES.

Option: PRINT.
Parameters: Y1, X1, Y2, X2, YPOLYGON, XPOLYGON, S, KVALUES.

Method

A procedure PTCHECKXY is called to check that X1 and Y1 have identical restrictions. Similar checks are made on X2 and Y2, and XPOLYGON and YPOLYGON. Procedure PTCLOSEPOLYGON is called to close the polygon specified by XPOLYGON and YPOLYGON. The SORT function is then used to create a variate containing the distances in S arranged in ascending order. (The original variate is left unchanged.) The procedure then calls a procedure PTPASS to call a Fortran program to calculate the bivariate K function.

Action with RESTRICT

The variates X1, Y1, X2, Y2, XPOLYGON, YPOLYGON and S may be restricted, as long as X1 has the same restriction as Y1, X2 has the same restriction as Y2 and XPOLYGON has the same restriction as YPOLYGON. Only the subset of values specified by each restriction will be included in the calculations.

References

Diggle, P.J. (1983). *Statistical Analysis of Spatial Point Patterns*. Academic Press, London.

Lotwick, H.W. & Silverman, B.W. (1982). Methods for analysing spatial processes of several types of points. *Journal of the Royal Statistical Society, Series B*, **44**, 406-413.

See also

Procedures: FHAT, GHAT, KHAT, KSTHAT, KTORENVELOPES.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

LCONCORDANCE

Calculates Lin's concordance correlation coefficient (R.W. Payne & M.S. Dhanoa).

Options

PRINT = *string token*	Controls printed output (concordance); default conc
GROUPS = *factor*	Defines the sets of measurements when they are all supplied in a single DATA variate
CONCORDANCE = *scalar* or *variate*	Saves Lin's the concordance coefficient
LOWER = *scalar* or *variate*	Saves the lower confidence limit for the coefficient
UPPER = *scalar* or *variate*	Saves the upper confidence limit for the coefficient
CORRELATION = *scalar* or *variate*	Saves the correlation coefficient
CB = *scalar* or *variate*	Saves the bias correction factor
ZTRANSFORMATION = *scalar* or *variate*	Saves the Z transformation of the coefficient
ZSD = *scalar* or *variate*	Saves the standard deviation of the Z transformation
CIPROBABILITY = *scalar*	Defines the size of the confidence interval; default 0.95 i.e. 95%
REFERENCELEVEL = *scalar* or *text*	Defines the set of measurements to be used as the control if there are more than two variates or groups; default 1

Parameter

DATA = *variates*	List of variates specifying the sets of measurements to be compared, or a single variate containing all the measurements (the GROUPS option must then be set to indicate the set to which each unit belongs)

Description

Lin's concordance correlation coefficient measures how well a new set of observations reproduce an original set. So, for example, it can be used to assess the effectiveness of new instruments or measurement methods.

The coefficient is calculated by multiplying two components. The first is the ordinary Pearson correlation coefficient, which essentially assesses the linear relationship between the two sets of measurements. However, for the second set to reproduce the first, the slope of the line relating the two sets should be one, and the line should go through the origin. These other aspects are assessed by the second component, which is known as C_b.

The measurements are supplied using the DATA parameter. You can set this to a list of variates, one for each measurement. Alternatively, you can put them all into a single variate, and set the GROUPS option to a factor to identify which measurement is stored in each unit of the variate. (LCONCORDANCE then assumes that the individuals that were measured are recorded in the same order within each set of measurements.) If there are more than two sets of measurements, LCONCORDANCE takes one of these as the control (i.e. the standard) set, and compares the others with this. By default the control is first variate if DATA has been set to a list of variates, or the set corresponding to the reference level of the GROUPS factor (see the FACTOR directive) if there was a single variate. However, you can define a different control by setting the REFERENCELEVEL option, to a scalar to indicate the number of the variate within the list of DATA variates of the level of the GROUPS factor. Alternatively, if the GROUPS factor has labels, you can set REFERENCELEVEL to a text.

Lin (1989, 2000) has shown that, if the coefficient is given an inverse hypobolic tangent transformation (i.e. a Z-transformation), the result has an approximate Normal distribution. LCONCORDANCE uses this to produce a confidence interval for the coefficient. The size of the

interval is specified by the CIPROBABILITY option; the default is 0.95 (i.e. 95%).

By default, the concordance coefficient, the lower and upper confidence limits, the correlation coefficient and Cb are printed. However, you can set option PRINT=* to suppress this. The CONCORDANCE, LOWER, UPPER, CORRELATION, CB, ZTRANSFORMATION and ZSD parameters allow the coefficient and all the associated information to be saved.

Options: PRINT, GROUPS, CONCORDANCE, LOWER, UPPER, CORRELATION, CB, ZTRANSFORMATION, ZSD, CIPROBABILITY, REFERENCELEVEL.
Parameter: DATA.

Method

The coefficient ρ_c is derived by Lin (1989) by considering how well the relationship between the measurements is represented by a line through the origin at an angle of 45 degrees (as would be generated if the two measurements generated identical results):

$$\rho_c = 1 - d_c^2 / d_u^2$$

where d_c^2 is the expected squared perpendicular deviation from the line, and d_u^2 is the expected squared perpendicular deviation from the line when the measurements are uncorrelated.

This can be written as

$$\rho_c = \rho \times C_b$$

The term ρ is the standard Pearson product-moment correlation coefficient, while C_b is a bias correction factor which is calculated by

$$C_b = 2 / (v + 1/v + u^2)$$
$$v = s_1 / s_2$$
$$u = (m_1 - m_2) / \sqrt{(s_1 \times s_2)}$$

where m_i and s_i (i = 1,2) are the mean and standard deviation of the i^{th} set of measurements.

The Z-transformation is

$$Z = 0.5 \times (\log(1 + \rho_c) / \log(1 - \rho_c))$$

The standard deviation of the Z-transformed coefficient is calculated as defined by Lin (2000).

Action with RESTRICT

If any of the DATA variates is restricted, the coefficient is calculated only for the units not excluded by the restriction.

References

Lin, L.I. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, **45**, 255-268.

Lin, L.I. (2000). A note on the concordance correlation coefficient. *Biometrics*, **56**, 324-325.

See also

Procedures: BLANDALTMAN, SLCONCORDANCE, CMHTEST, FCORRELATION, KCONCORDANCE, KTAU, SPEARMAN.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

LIBEXAMPLE

Accesses examples and source code of library procedures (R.W. Payne).

No options

Parameters

PROCEDURE = *texts*	Single-valued texts indicating the procedures about which the information is required
EXAMPLE = *texts*	Identifiers of text structures to store the example for each procedure
SOURCE = *texts*	Identifiers of text structures to store the source code of each procedure

Description

LIBEXAMPLE allows you to obtain an example of the use of any procedure in the Genstat Procedure Library, also to access the source code of any procedure, so that you can see how it works, or modify it. The names of procedures for which examples or source code are required should be listed, in quotes, using the PROCEDURE parameter. The EXAMPLE parameter can be used to specify the identifier of a text to store each example, and the SOURCE parameter to specify texts to store the source code. The examples can then be run (as macros) using the operator ##. Thus,

```
LIBEXAMPLE 'PERCENT'; EXAMPLE=%Ex
##%Ex
```

would put an example of how to use PERCENT into the text %Ex, and then run it.

Options: none. Parameters: PROCEDURE, EXAMPLE, SOURCE.

Method

The examples are read from the Genstat Examples folder, which is usually located alongside the bin folder that holds the Genstat executable program. The source code is held in a backing-store file whose name is supplied by procedure LIBFILENAME. This file is opened on the first available backing-store channel; if all the channels are in use, the procedures stops with a diagnostic. After the code has been brought back from backing store, the file is closed.

See also

Directive: HELP.
Procedures: LIBFILENAME, LIBHELP, LIBVERSION.

LIBFILENAME

Supplies the names of information files for library procedures (R.W. Payne).

No options

Parameters

FILENAME = *texts*	Text in which to store the name of the backing-store file containing the required information
CONTENTS = *string tokens*	Indicates which file is required (procedures, adesign, afraction, acyclic, agenerator); default proc

Description

The source code of the procedures in the Genstat Procedure Library is stored in a backing-store file for use by the LIBEXAMPLE procedure. Other backing-store files store information that is used by some of the Genstat procedures for design of experiments. When a procedure needs to access any of this information, it calls LIBFILENAME to ascertain the name and location of the relevant file. (It then opens the file on the first free backing-store channel, reads the required information, and closes the file again.)

Options: none. Parameters: FILENAME, CONTENTS..

Method

The procedure contains a text structure containing the various filenames, and the POSITION function of CALCULATE is used to set FILENAME to the appropriate one.

Action with RESTRICT

Any restriction on the FILENAME text will be cancelled.

See also

Directive: HELP.
Procedures: LIBEXAMPLE, LIBHELP, LIBVERSION.

LIBHELP

Provides help information about library procedures (R.W. Payne).

No options

Parameter

PROCEDURE = *texts*	Single-valued texts indicating the procedures about which the information is required; if this is not set, information is given about LIBHELP itself

Description

LIBHELP provides information about procedures in the Genstat Procedure Library. It has a parameter, called PROCEDURE, which you use to indicate the procedures for which you want information; if PROCEDURE is not specified, information is given about LIBHELP itself. The names of the procedures should be given in quotes: for example

```
LIBHELP 'GLMM'
```

will obtain information about the procedure GLMM.

Options: none.
Parameter: PROCEDURE.

Method

In Genstat *for Windows*, LIBHELP opens the Windows on-line help file at the appropriate page.

See also

Directive: HELP.
Procedures: LIBEXAMPLE, LIBFILENAME, LIBVERSION.

LIBSOURCE

Obtains the source code of a Genstat procedure, PC Windows only (R.W. Payne).

No options

Parameters

PROCEDURE = *texts*	Procedure names
SOURCE = *texts*	Texts to store the source code of each procedure
STATEMENT = *texts*	Saves a command to obtain the source of each procedure (useful if the name has been specified in response to questions from PROCEDURE)

Description

LIBSOURCE provides a convenient way of accessing the source code of a procedure from the Genstat Procedure Library. The name of the procedure is specified by the PROCEDURE parameter. Alternatively, if you are running Genstat interactively, you can type just

```
LIBSOURCE
```

and Genstat will list the types of procedure, followed (once you have selected the type) by a list of the procedures themselves. In fact, this is what happens if you select Help followed by Procedure Source from the menu bar of Genstat *for Windows*. If, you wish to access the same example later, the STATEMENT parameter allows you to save a Genstat text structure containing a command setting the PROCEDURE parameter as required.

When you are running Genstat interactively, the procedure is put into a new text window. When Genstat is running in batch, the example is printed to the output window. You can also save the source code in a Genstat text structure, using the SOURCE parameter.

Options: none.
Parameters: PROCEDURE, SOURCE, STATEMENT.

See also

Procedure: LIBEXAMPLE.
Genstat Reference Manual 1 Summary section on: Program control.

LIBVERSION

Provides the name of the current Genstat Procedure Library (R.W. Payne).

Option

PRINT = *string token*	Controls printed output (release); default rele

Parameter

RELEASENAME = *text*	Text in which to store the name of the currently available release of the Genstat Procedure Library

Description

The Genstat Procedure Library is updated independently of releases of the main Genstat program and the current release thus may not be immediately apparent. Consequently LIBVERSION is provided to allow users to obtain the name of the currently available release. The name is printed by default, but you can set option PRINT=* to suppress this. The RELEASENAME parameter allows the name, 'Genstat Procedure Library Release ...' to be saved.

Options: none.
Parameter: RELEASENAME.

Method

RELEASENAME is formed by an ordinary TEXT declaration.

Action with RESTRICT

Any restriction on the RELEASENAME text will be cancelled.

See also

Directive: HELP.
Procedures: LIBEXAMPLE, LIBFILENAME, LIBHELP.

LINDEPENDENCE

Finds the linear relations associated with matrix singularities (J.H. Maindonald).

Option

PRINT = *string tokens*	Printed output (dependent, coefficients); default depe

Parameters

DATA = *symmetric matrices*	Specifies the positive semi-definite matrix for which the information is required
COEFFICIENTS = *matrices*	Stores the coefficients of the linear dependencies

Description

Procedure LINDEPENDENCE takes a positive semi-definite matrix S (e.g. a matrix formed as $X'X$), and identifies any columns of S that are a linear combination of earlier columns. It determines the linear relations involved, and stores these in the columns of the matrix specified by the COEFFICIENTS parameter.

In more mathematical terms the output, stored as columns of COEFFICIENTS, is a basis for the null space of a positive semi-definite matrix S. If $S = X'X$, then this will also be a basis for the column space of X.

The first parameter, DATA, specifies the symmetric matrix S for which the information is required. The columns of the COEFFICIENTS matrix store the linear relations. This matrix will be defined automatically if it has not been declared earlier.

Printed output information on either which columns are dependent and/or what the coefficients for the dependencies are can be requested with the settings dependent and coefficients of the PRINT option. By default the dependent columns are printed.

Option: PRINT. Parameters: DATA, COEFFICIENTS.

Method

The matrix function CHOLESKI is used to determine a lower triangular matrix L such that $LL' = S$. Zeros on the diagonal of L identify columns of S that are a linear combination of earlier columns. The corresponding columns of L' form a matrix H. The algorithm then replaces zeros on the diagonal of L' by ones, to give the matrix T, and solves the equation $T B = H$. Finally it identifies in each column of H the element that was originally on the diagonal of L, and sets each such element to −1. For further details, see Maindonald (1984) page 105.

Warning – if S is inaccurately formed, e.g. using single precision calculations, there is a risk that it will not be detected as singular, or that it will be detected as not positive semi-definite.

Reference

Maindonald, J.H. (1984). *Statistical Computation*. Wiley, New York.

See also

Procedure: POSSEMIDEFINITE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

LORENZ

Plots the Lorenz curve and calculates the Gini and asymmetry coefficients (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (gini, lorenz, asymmetry); default gini, lore, asym
PLOT = *string token*	Controls graphical output (curve); default curv
TITLE = *string*	Title for the graph; default uses the identifier of the DATA variate
NBOOT = *scalar*	Number of samples to make to construct the bootstrap confidence intervals; default 100
SEED = *scalar*	Seed for the random number generator used to construct the bootstrap samples; default 0 i.e. continue an existing sequence of random numbers or, if none, initialize the generator automatically
CIPROBABILITY = *scalar*	Probability for the bootstrap confidence interval; default 0.95

Parameters

DATA = *variates*	Specifies sets of data values
GINI = *scalars*	Saves the Gini coefficient for each DATA variate
ASYMMETRY = *scalars*	Saves the asymmetry coefficient for each DATA variate

Description

The Lorenz curve provides a graphical representation of the inequality of a sample of numbers. In economics the numbers could be the annual incomes of a group of people, or in ecology they could be population sizes of a set of species of animal or plant. The y-coefficients for the curve are formed by sorting the numbers, calculating their cumulative totals, and then dividing these by the grand total. The x-coefficients are simply the numbers 0, 1, ... *n*, where *n* is the size of the sample. If the numbers are all equal, the curve will form a straight line, known as the line of equality, running from the origin to the point (1, 1). Inequalities amongst the numbers cause the curve to lie below the line of equality.

The Gini coefficient is the area between the line of equality and the Lorenz curve area, divided by area under the line of equality. So, a value close to zero indicates near equality, while a value near to one shows a high amount of inequality. The asymmetry coefficient assesses the amount of asymmetry of the Lorenz curve. The axis of symmetry for the curve is the line from (1, 0) to (0, 1). The coefficient is less than one if the point where the Lorenz curve is parallel to the line of equality lies below the axis of symmetry, and greater than one if it lies above the axis.

The numbers whose equality is to be studied are specified, in a variate, by the DATA parameter. Their Gini and asymmetry coefficients can be saved, in scalars, using the GINI and ASYMMETRY parameters respectively.

Printed output is controlled by the PRINT option, with settings:

asymmetry	prints the coefficient of asymmetry,
gini	prints the Gini coefficient,
lorenz	prints the coordinates of the Lorenz curve.

By default, these are all printed.

The procedure can also print bootstrap confidence intervals for the Gini and asymmetry coefficients. The probability level for the interval is specified by the CIPROBABILITY option; the default of 0.95 gives 95% intervals. The NBOOT option specifies how many bootstrap samples to take (default 100). If you do not want the confidence intervals, you should set NBOOT=0. The SEED option specifies the seed to use in the random number generator used to construct the

bootstrap samples. The default value of zero continues an existing sequence of random numbers or, if the generator has not yet been used in this run of Genstat, it initializes the generator automatically.

By default curve is plotted, but you can set `PLOT=*` to suppress the plot. The `TITLE` option can supply a title for the graph.

Options: `PRINT`, `PLOT`, `TITLE`, `NBOOT`, `SEED`, `CIPROBABILITY`.
Parameters: `DATA`, `GINI`, `ASYMMETRY`.

Method

The Gini coefficient is calculated using the equation

Gini = $\sum\{ ((2 \times i - n - 1) * \texttt{Dsort}) / (\text{mean}(\texttt{DATA}) \times n^2) \}$

where n is the sample size, `Dsort` are the sorted numbers.

The asymmetry coefficient is given by

```
Asymmetry = Fmu + Lmu
```

with *Fmu* and *Lmu* defined by

$Fmu = (m + d) / n$

$Lmu = (\texttt{CDsort}_m + d \times \texttt{Dsort}_{m+1}) / \texttt{CDsort}_n$

where m is index of the largest number less than mean(`DATA`),

```
CDsort = CUMULATE(Dsort),
```

and

$d = (\text{mean}(\texttt{DATA}) - \texttt{Dsort}_m) / (\texttt{Dsort}_{m+1} - \texttt{Dsort}_m)$

The bootstrap confidence intervals are generated using the `BOOTSTRAP` procedure.

Action with `RESTRICT`

`LORENZ` takes account of any restrictions on the `DATA` variate.

See also

Genstat Reference Manual 1 Summary section on: Ecological data.

LRIDGE

Does logistic ridge regression (A.I. Glaser).

Options

PRINT = *string token*	What output to print (correlation, crossvalidation, ridge, scaledridge, standarderrors); default corr
PLOT = *string tokens*	What graphs to plot (correlation, ridgetrace, buildup); default * i.e. none
LINK = *string token*	Link function (logit, probit, complementaryloglog); default logi
DISPERSION = *scalar*	Value of the dispersion parameter; default 1
TERMS = *formula*	Explanatory model
FACTORIAL = *scalar*	Limit on number of factors/covariates in a model term; default 3
LAMBDA = *variate* or *scalar*	Values for the ridge parameter lambda
CROSSVALIDATION = *string token*	Whether to use cross-validation to find an optimal value of lambda (yes, no); default no
NCROSSVALIDATIONGROUPS = *scalar*	Number of groups for cross-validation; default 10
CVMETHOD = *string token*	Which method to use for cross-validation (deviance, squarederror, countingerror); default devi
SEED = *scalar*	Seed for random numbers to use in cross-validation; default 0

Parameters

Y = *variates*	Response variate
NBINOMIAL = *scalars* or *variates*	Number of binomial trials for each unit; default 1
YVALIDATION = *variates*	Response variate for validation
XVALIDATION = *pointers*	Explanatory variables for validation
XDATA = *pointers*	Pointer containing the original explanatory variables in the same order as in XVALIDATION; default takes the variables in the order in which they occur in TERMS
NVALIDATION = *variates* or *scalars*	Number of binomial trials for the units of each YVALIDATION variate; default 1
BESTLAMBDA = *scalars*	Saves the optimal lambda value from cross-validation
CVSTATISTICS = *matrices*	Saves the cross-validation statistics
RESIDUALS = *variates*	Saves residuals when LAMBDA is a scalar
FITTEDVALUES = *variates*	Saves fitted values when LAMBDA is a scalar
ESTIMATES = *variates*	Saves parameter estimates when LAMBDA is a scalar
SE = *variates*	Saves standard errors of the parameter estimates when LAMBDA is a scalar
DEVIANCE = *scalars*	Saves the residual deviance when LAMBDA is a scalar
LINEARPREDICTOR = *variates*	Saves the linear predictor when LAMBDA is a scalar

Description

Procedure LRIDGE fits a logistic ridge regression model based on penalized likelihood inference, as explained in the *Method* section. The response variate is specified by the Y parameter. The NBINOMIAL parameter defines the number of binomial trials for each unit, with a default of one. If NBINOMIAL is greater then one, LRIDGE forms a modified copy of the data set in which each

of the original observations is expanded into its underlying individuals (i.e. to have binary responses either one or zero).

The model to fit is defined by the TERMS option. The FACTORIAL option sets a limit on the number of variates and/or factors in the model terms generated from the TERMS model formula, as in the FIT directive. The LINK option defines the link function. This can be either logit (the default), probit or complementary-log-log. The DISPERSION option specifies the dispersion parameter in the usual way i.e. the default is to fix the parameter at one, or you can set DISPERSION=* to use a dispersion parameter estimated from the residual deviance.

Printed output is controlled by the PRINT option, with settings:

correlation	prints the correlations between the explanatory variables in the TERMS formula,
crossvalidation	prints the cross-validation results, with optimal lambda value,
ridge	prints the ridge coefficients on the original scale,
scaledridge	prints the ridge coefficients for the standardized data, and
standarderrors	includes standard errors with coefficients printed by the ridge or scaledridge settings.

Graphical output is controlled by the PLOT option:

ridgetrace	produces coefficient estimates against lambda, showing how they decrease as lambda increases,
buildup	plots coefficient values against the coefficients divided by their maximum values, showing the relative decrease as lambda increases, and
correlation	uses the DCORRELATION procedure to produce a graphical representation of the correlation matrix for elements in TERMS.

The LAMBDA option allows you to define the values to try for the ridge parameter lambda (see *Method*). By default LRIDGE takes a range of values between 0 and 1. If you have set LAMBDA to a single value, you can save results from the analysis using the RESIDUALS, FITTEDVALUES, ESTIMATES, DEVIANCE and LINEARPREDICTOR parameters. Note that the residuals are simple residuals, rather than standardized residuals.

LRIDGE can use cross-validation to find an optimal value of lambda. The YVALIDATION, XVALIDATION and NVALIDATION parameters allow you to supply an independent data set for validation. The YVALIDATION parameter specifies the response variate, the NVALIDATION parameter specifies the corresponding numbers of binomial trials (default 1), and the XVALIDATION supplies a pointer containing values for the explanatory variables. LRIDGE needs to match the validation explanatory variables with the original variables in TERMS. You can define the correspondence explicitly by setting the XDATA parameter to a pointer containing the original variables in the same order as the corresponding variables in the XVALIDATION pointer. If XDATA is not set, LRIDGE forms the original list using the CLASSIFICATION of the FCLASSIFICATION directive. The order of variables should easily be predictable for straightforward TERMS models, but it is safest to specify XDATA explicitly for complicated models.

If you do not have an independent data set, LRIDGE can do the validation by selecting subsets of the original data set. The NCROSSVALIDATIONGROUPS option defines the number of subsets (default 10). The data set (modified to contain binary responses, as explained above, if NBINOMIAL is greater than one) is divided into that number of roughly equal-sized subsets. The model is fitted to the data set with each of these parts removed, in turn, and the prediction error is calculated for the omitted subset based on that fit. The method for calculating the prediction error is specified by the CVMETHOD option:

deviance	uses the deviance function (defined as twice the difference

between the maximum log-likelihood and that achieved under the validation data),

`squarederror` takes the sum of the squared differences between the validation data and the expected values, and

`countingerror` counts the number of "wrong" predictions in the validation data, i.e. if the value of the validation data was 1 and the expected probability was less than 0.5, the prediction would be considered to be wrong.

The calculation of the prediction error is repeated for every value of the `LAMBDA` option. The value that minimizes the mean prediction error is taken as the optimal lambda, and can be saved by the `BESTLAMBDA` parameter. (You could then use `LRIDGE` again, with `LAMBDA` set to that value, and use the parameters `RESIDUALS`, `FITTEDVALUES` etc. to save information from the optimal analysis.)

Options: `PRINT`, `PLOT`, `LINK`, `DISPERSION`, `TERMS`, `FACTORIAL`, `LAMBDA`, `CROSSVALIDATION`, `NCROSSVALIDATIONGROUPS`, `CVMETHOD`, `SEED`.

Parameters: `Y`, `NBINOMIAL`, `YVALIDATION`, `XVALIDATION`, `XDATA`, `NVALIDATION`, `BESTLAMBDA`, `CVSTATISTICS`, `RESIDUALS`, `FITTEDVALUES`, `ESTIMATES`, `SE`, `DEVIANCE`, `LINEARPREDICTOR`.

Method

Logistic ridge regression is carried out as described by le Cessie & van Houwelingen (1992). The usual log-likelihood for logistic regression is extended to include a penalty on the sum of squares of the parameter estimates β, namely $\lambda \times \sqrt{\{\sum \beta^2\}}$. When the ridge parameter, lambda, is equal to zero, the parameter estimates will be the usual maximum-likelihood estimates, whereas as lambda tends to infinity all of the parameters tend towards zero. The penalty term is applied by setting the `RIDGE` option of the `TERMS` directive. The columns of the design matrix in `TERMS` are standardized. However, estimated coefficients are available for both the standardized and unstandardized data.

Action with `RESTRICT`

There must be no restrictions.

Reference

le Cessie, S. & van Houwelingen, J.C. (1992). Ridge estimators in logistic regression. *Applied Statistics*, **41**, 191-202.

See also

Procedures: `RIDGE`, `RLASSO`.

Genstat Reference Manual 1 Summary section on: Regression analysis.

LRVSCREE

Prints a scree diagram and/or a difference table of latent roots (P.G.N. Digby).

Options

PRINT = *string tokens*	Printed output (scree, differences); default scre
PLOT = *string token*	What to plot in high-resolution graphics (scree); default scre
TITLE = *text*	Title for the graph; default * i.e. none
WINDOW = *scalar*	Window to use for the graph; default 1

Parameters

ROOTS = *LRVs* or *any numerical structures*	Latent roots to be displayed; if an LRV is supplied the trace will also be extracted from it
TRACE = *scalars*	Supplies or saves the total of the latent roots
DIFFERENCES = *pointers*	Contains 3 variates to save the difference table

Description

Procedure LRVSCREE displays a set of latent roots in a convenient form. The input to the procedure is a set of latent roots (ROOTS), either as an LRV or any structure with numerical values. Optionally a scalar (TRACE) can be specified, either to supply or to save the total of the latent roots.

Printed output is controlled by the PRINT option. The setting scree produces a scree diagram, annotated with the latent roots on their original scale and expressed both as per-thousandths of the total and as cumulated per-thousandths. The setting differences prints these quantities as a table, together with the first three differences among the per-thousandth values; i.e. the first difference column gives the differences from each per-thousandth to the next, the second difference column gives differences among the first-difference values, and so on. Large first-difference values indicate latent roots ocurring prior to large declines in the scree diagram. Large second and third differences mark the locations of series of two or more latent roots of similar magnitude, which can be thought of as plateaus on the scree diagram. Large positive, or negative, second differences indicate the first, or last, latent root of a plateau. Large negative third differences occur at the last latent root of one plateau that is followed by another plateau. See the example for illustration.

By default the scree diagram is also plotted in high-resolution graphics but this can be suppressed by setting option PLOT=*. The TITLE option can supply a title for the plot, and the WINDOW option specifies which window is used (by default window 1).

The DIFFERENCES parameter allows a pointer to be specified to contain three variates storing the columns of the difference table.

Options: PRINT, PLOT, TITLE, WINDOW.
Parameters: ROOTS, TRACE, DIFFERENCES.

Action with RESTRICT

Not relevant: LRVSCREE deals primarily with diagonal matrices or LRVs. If the latent roots are supplied in a variate, any restriction on the variate will be ignored.

See also
Directives: CVA, PCP, PCO.
Procedure: QEIGENANALYSIS.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Graphics.

LSIPLOT

Plots least significant intervals, saved from SEDLSI (M.C. Hannah).

Options

WINDOW = *scalar*	Window in which to plot the graph
TITLE = *text*	Title for the graph; default 'Estimates with LSIs by Treatment'
YTITLE = *text*	Title for the y-axis; default 'Estimates'

Parameters

LSI = *pointers*	Defines the least significant intervals
†SYMBOL = *texts* or *scalars*	Symbol to use to plot each set of estimates
†CSYMBOL = *texts* or *scalars*	Colour for each symbol
†SMSYMBOL = *scalars*	Multiplier to use in the calculation of the size of each symbol
†SMLABEL = *scalars*	Multiplier to use in the calculation of the size of the labels in each plot

Description

Least significant intervals (*LSI*s) are used for comparing a set of estimates (e.g. predicted means from ANOVA or regression) graphically, especially when their SEDs differ. LSIs are intervals (or error bars) that are designed to overlap where there is no significant difference between estimates, and to be disjoint (i.e. not to overlap) where there are significant differences.

LSIs can be calculated by the SEDLSI procedure and saved, in a pointer, using its LSI parameter. This pointer can then be be supplied as input to LSIPLOT, using its own LSI parameter, to plot the intervals on a later occasion.

LSIPLOT has an option WINDOW to specify the window in which to plot the LSIs. By default a window is defined internally, within LSIPLOT, to fill the whole screen. The TITLE option allows you to supply a title for the plot (default 'Estimates with LSIs by Treatment'), and the YTITLE option supplies a title for the y-axis (default 'Estimates').

The SYMBOL parameter specifies the symbol to use to plot the estimates; by default, this is a circle. The CSYMBOL parameter specifies the colour (default black). The SMSYMBOL and SMLABEL parameters specify the multipliers to use when calculating the sizes of the symbols and the labels, instead of the default values calculated by the procedure.

Options: WINDOW, TITLE, YTITLE.
Parameters: LSI, SYMBOL, CSYMBOL, SMSYMBOL, SMLABEL.

See also

Procedures: SEDLSI, SED2ESE.
Genstat Reference Manual 1 Summary section on: Graphics.

LSPLINE

Calculates design matrices to fit a natural polynomial or trigonometric L-spline as a linear mixed model (S.J. Welham).

Options

KMETHOD = *string token*	Method for constructing the set of knots (equal, quantile, given); default equa
NSEGMENTS = *scalar*	Specifies the number of segments between boundaries; default * obtains a value automatically
INKNOTS = *variate*	Provides the set of internal knots when KMETHOD=given
CORE = *string token*	The form of core function to use; (cossin, intcossin, lincossin, intercept, linear, quadratic) default linc
PERIOD = *scalar*	Defines the period for trigonometric functions (not required for polynomial splines)
LOWER = *scalar*	Specifies the lower boundary when KMETHOD=equal; default takes the minimum value in X
UPPER = *scalar*	Specifies the upper boundary when KMETHOD=equal; default takes the maximum value in X
ORTHOGONALIZETO = *variate*	Variate to use to get an orthogonalized basis; default * i.e. orthogonalization with respect to X
SCALING = *scalar*	Scaling of the XRANDOM terms (automatic, none); default auto

Parameters

X = *variates*	The explanatory variate for which the spline values are required
XFIXED = *matrices*	Saves the design matrix to define the fixed terms (excluding the constant) for fitting the L-spline
XRANDOM = *matrices*	Saves the design matrix to define the random terms for fitting the L-spline
KNOTS = *variates*	Saves the internal knots and boundaries used to form the basis for the spline
PX = *variates*	Specifies x-values at which predictions are required
PFIXED = *matrices*	Saves the design matrix for the fixed terms (excluding the constant) for the spline at the prediction points
PRANDOM = *matrices*	Saves the design matrix for the random terms for the spline at the prediction points

Description

This procedure generates the fixed and random terms required to fit a polynomial or trigonometric L-spline as a linear mixed model, using REML estimation of the smoothing parameter (Welham *et al.*, 2006). The explanatory variate values at which the spline is to be calculated are specified, in a variate, by the X parameter.

The KMETHOD option specifies how to choose the set of knots for the penalized spline, using settings:

equal	splits the range of X into segments of equal length (default),
quantiles	defines the set of knots in terms of equally-spaced quantiles of X,
given	indicates that the knots will be supplied, in a variate, by the

INKNOTS option.

The number of segments or quantiles for the `equal` and `quantile` settings is specified by the `NSEGMENTS` option. If this is unset, the number is determined automatically as

$$\min([p/4], 35) + 1$$

(Ruppert 2002) where p is the number of unique values of the variate `X` and $[r]$ denotes the integer part of the number r. The lower and upper boundaries for equal segments are specified by the `LOWER` and `UPPER` options, respectively, taking the minimum and maximum values of `X` as their defaults. The set of knots used to form the spline basis can be saved using the `KNOTS` parameter.

The form of the core functions used to generate the L-spline, i.e. the form that will remain unpenalized, is specified using the `CORE` option. The settings `cossin`, `intcossin` and `lincossin` define trigonometric L-splines, with a (known) period defined by the `PERIOD` option. These have the following forms:

`cossin`	$a_1 \cos(kx) + a_2 \sin(kx)$
`intcossin`	$a_0 + a_1 \cos(kx) + a_2 \sin(kx)$
lincossin	$a_0 + a_1 + a_2 \cos(kx) + a_3 \sin(kx)$

where

$$k = 2\,\pi\,/\,\texttt{PERIOD}$$

and the a_i are unknown coefficients.

The settings `intercept`, `linear` and `quadratic` define polynomial splines, as follows:

`intercept`	natural linear spline with a single constant core function,
`linear`	natural cubic spline with linear core functions, and
`quadratic`	natural quintic spline with quadratic core functions.

If the distinct values of `X` are used as knots, the function will produce a basis for a polynomial smoothing spline; with fewer knots, the spline is a low-rank approximation equivalent to the O-splines described by Ormerod & Wand (2008). A different parameterization of the basis is used here, but the fitted spline will be the same.

The `ORTHOGONALIZETO` option specifies a variate to use in orthogonalization. The set of random spline terms will then be orthogonal to the fixed terms when evaluated at the specified values. For most data sets, it is recommended to set `ORTHOGONALIZETO` to the variate `X` (the default). The random terms will then be orthogonal to the fixed terms, and fitted values corresponding to the fixed model will represent the whole of the polynomial trend in the fitted spline. For very large data sets, this calculation can be onerous and can be approximated by making the two bases orthogonal at the knots. No orthogonalization is carried out if `ORTHOGONALIZETO` is set to a scalar value (eg. `ORTHOGONALIZETO=0`).

The L-spline terms are saved as two matrices. The terms required to be fitted as fixed terms can be saved using the `XFIXED` parameter. For `CORE=cossin`, the constant should be omitted from the model by setting option `CONSTANT=omit` in `VCOMPONENTS`. For `CORE=intercept`, there are no additional fixed terms, and so `XFIXED` is ignored. The terms to be fitted as random can be saved using the `XRANDOM` parameter.

The random terms can be scaled so that, for a random spline matrix `Z`,

```
TRACE(Z *+ T(Z)) = NROWS(Z)
```

This ensures that the average contribution of `Z` to the variance of an observation is equal to one, and hence the overall contribution from the term is equal to the spline variance component. This is highly recommended as it removes computational instabilities due to the intrinsic scale of the matrix `K` (see *Method*), and improves interpretability of the spline variance component. This scaling is imposed by default, but can be avoided by setting option `SCALING=none`. For L-splines, use of some scaling is strongly recommended, as the unscaled matrix can be so large or small that the associated variance component appears aliased and cannot be estimated.

The L-spline terms required for prediction can be saved using the `PXFIXED` and `PXRANDOM` parameters. The `PX` parameter defines the set of x-values at which the predictions are to be made.

Options: KMETHOD, NSEGMENTS, INKNOTS, CORE, PERIOD, LOWER, UPPER, ORTHOGONALIZETO, SCALING.
Parameters: X, XFIXED, XRANDOM, KNOTS, PX, PXFIXED, PXRANDOM.

Method

The L-spline with core functions $\sum \tau_i d_i(x)$ and r knots, evaluated on variate X, minimizes the penalized sum of squares

$$(y - X\tau - K_x c)' R^{-1} (y - X\tau - K_x c) + \lambda c' K c$$

where

- X is a design matrix containing the core functions evaluated at x, with associated unknown parameters τ;
- K_x is a design matrix containing r L-spline basis functions evaluated at the knots, with associated unknown effects c; and
- K is an $r \times r$ symmetric matrix of L-spline basis functions evaluated at the knots (for details see references in Welham *et al.* 2006)

The matrix K can be transformed to full rank as

$$H = C' K C$$

where the matrix C contains the eigenvectors of XX' corresponding to zero eigenvalues, with corresponding transformation of the matrix functions as

$$K_u = K_x C$$

This is translated to a set of independent random effects via post-multiplication by $H^{-1/2}$.

The penalized sum of squares is reformulated as the estimating equations from a mixed model of the form

$$y = X\tau + Z u + e$$

where

$Z = K_x C H_m^{-1/2}$

- u is a set of r independently and identically distributed random effects with $\text{var}(u) = \sigma_s^2 I$
- e is a vector of residual errors with $\text{var}(e) = \sigma^2 R$

Fitting this mixed model with known λ set equal to σ^2/σ_s^2 produces estimates that minimize the penalized sum of squares. In addition, we can estimate the smoothing parameter using REML via the variance component σ_s^2. This can be generalized straightforwardly to mixed models with additional fixed and random terms.

The implementation in this procedure allows the random design matrix to be orthogonalized with respect to the fixed design matrix at a given variate. For orthogonalization with respect to the variate X, this is achieved by using random design matrix

$$Z^* = (I - X(X'X)^{-1}X')Z$$

The entirety of the polynomial trend is then captured by the fixed model. Orthogonalization with respect to a variate t is calculated as

$$Z^* = Z - X(T'T)^{-1}T' K_t C H_m^{-1/2}$$

where T is a matrix holding the core functions evaluated at t, and K_t is a matrix of L-spline basis functions evaluated at t. No orthogonalization is carried out if ORTHOGONALIZETO is set to a scalar value (e.g. ORTHOGONALIZETO=0).

When the random matrix is scaled so that $\text{trace}(Z^*Z^{*\prime})$ is equal to the number of row of Z^*, the average contribution of the spline term to the variance of each unit ($\sigma_s^2 \times \text{diag}(Z^*Z^{*\prime})$) is equal to σ_s^2. This makes the spline variance component value directly comparable with the residual variance.

Note that the constant function is not included in the fixed design matrix generated by PENSPLINE, as this term is added automatically to the linear mixed model by the default option setting, CONSTANT=estimate, in the VCOMPONENTS statement.

The design matrices for use in prediction are calculated by evaluating the same set of basis functions at the predict points.

Action with RESTRICT
Input structures must not be restricted.

References
Ruppert, D. (2002). Selecting the number of knots for penalised splines. *Computational & Graphical Statistics*, **11**, 735-757.
Wand, M.P. & Ormerod, J.T. (2008). On semiparametric regression with O'Sullivan penalised splines. *Australian & New Zealand Journal of Statistics*, **50**, 179-198.
Welham, S.J., Cullis, B.R., Kenward, M.G., Thompson, R. (2006). The analysis of longitudinal data using mixed model L-splines. *Biometrics*, **62**, 392-401.

See also
Directive: VCOMPONENTS.
Procedures: SPLINE, NCSPLINE, PENSPLINE, PSPLINE, RADIALSPLINE, TENSORSPLINE.
Function: SSPLINE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

LVARMODEL

Analyses a field trial using the Linear Variance Neighbour model (D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output (data, effects, sed, residuals, variances); default effe, sed, vari
METHOD = *string token*	Indicates which version of the LV model to use (full, reduced); default full
LAMBDA = *scalar*	Number between 0 and 1 which defines the value for the variance parameter λ (if METHOD=full and LAMBDA=0, the value is estimated by REML); default 0
VARMETHOD = *string token*	Specifies which estimator of residual variance to use to calculate the sed's of treatment effects (RMS2, GLS) default RMS2
TOLERANCE = *scalar*	Defines the precision to which the variance parameter λ should be estimated; default 0.01

Parameters

Y = *variates*	Y-values (usually plot yields) row by row
TREATMENTS = *factors*	Treatment factor for each y-variate
BLOCKS = *factors*	Block factor, defining groups of plots to be de-trended independently
UNITS = *factors*	Unit-within-block factor, defining the order of plots within each block
EFFECTS = *tables*	Saves the estimated treatment effects from each analysis
SED = *matrices* or *symmetric matrices*	Saves the estimated standard errors of differences between treatments
WNOISE = *variates*	Saves the estimated white noise component
TREND = *variates*	Saves the estimated trend component
COMPONENTS = *variates*	Saves the estimated variance components: the tuning parameter λ, and either the variance of the random walk innovations (λ<0.9) or the white noise variance (λ≥0.9)

Description

LVARMODEL analyses a field trial, whose plots are in lines of equal length, using the Linear Variance (LV) Neighbour analysis (Williams 1986). The LV model is equivalent to the extended First Difference model of Besag & Kempton (1986). The model allows for local trends within a row, and the analysis attempts to remove these trends by using a form of smoothing. In the full LV model, the degree of smoothing is estimated from the data; alternatively the reduced model, corresponding to the ordinary First Difference (FD) model of Besag & Kempton (1986), applies a full linear de-trending to the data.

The LV model specifies the data as the sum of three components: the treatment effects, a trend component which is a random walk process, and a residual white noise component. The full Linear Variance plus Incomplete Block model of Williams (1986) has an additional random component for incomplete blocks, These can be fitted as a fixed effect, by treating each block as a separate line of plots.

The variable to be analysed (normally a plot yield) is specified in a variate, using the Y parameter. The factor defining the treatments on the plots is specified using the TREATMENTS parameter. The BLOCKS parameter specifies the block factor, defining the groups of plots that are to be de-trended separately, and the UNITS parameter specifies the units-within-blocks factor

defining the order of the plots within each block. For example, if the plots are on a rectangular grid and trends are to be removed along rows, the BLOCKS and UNITS factors would be the row and column factors, respectively. If BLOCKS and UNITS are not set, the plots are assumed to be in a single line (and specified sequentially down the line). The procedure can handle missing values in the y-variate but not in the TREATMENTS, BLOCKS or UNITS factors.

The other parameters allow information to be saved from the analysis: EFFECTS for the table of estimated treatment effects; SED for the standard errors of differences between treatments effects (in either a matrix or a symmetric matrix); WNOISE for the estimated white noise (in a variate); TREND for trend component (in a variate); and COMPONENTS for the two variance parameters. The first variance component is the parameter λ. For $\lambda<0.9$ the second component is the variance of the innovations in the random walk. If $\lambda \geq 0.9$ the second component saved is the variance of the white noise component, as the random walk component disappears in the limit as λ tends to one.

Printed output is controlled by the PRINT option with the following settings: data - y-values and treatments in a tabular form; effects estimated treatment effects; sed standard errors of differences of effects; variance estimates of λ and the white noise variance; and residuals trend and white noise components.

The METHOD option controls the form of LV model to be fitted. By default setting of full causes the full LV model to be fitted, with the variance parameters of the model estimated by Residual Maximum Likelihood (REML); see Gleeson & Cullis (1987). The variance parameters used, λ and κ, are those given by Baird and Mead (1991). The parameter λ is known as the tuning parameter, as it controls the degree of smoothing used in eliminating trend effects from the data. It is related to the parameter α of Besag & Kempton (1986), by the relationship

$$\lambda = \alpha / (1 + \alpha)$$

Alternatively, specifying METHOD=reduced fits the reduced form of the LV model, that is the FD model. This is equivalent to putting $\lambda = 0$.

The option LAMBDA allows the value of the tuning parameter to be set at a fixed value, which must lie between 0 and 1. By default LAMBDA=0, which for METHOD=full causes the value to be estimated as described above.

The option VARMETHOD controls the estimator used for the estimating the variance of the residual white noise component. There are two possibilities: the normal generalized least-squares estimator GLS, and an estimator based on the second differences of the errors RMS2 (Besag & Kempton 1986). The simulation study of Baird & Mead (1991) showed the standard errors of treatment effects based on RMS2 to be approximately valid under randomization for a wide range of error models. When the estimated value of λ was not close to zero, the standard errors based on GLS were found to be approximately unbiased and more efficient than those based on RMS2 for the LV model. However the standard errors based on GLS could be seriously biased in some situations for the FD model or when λ was close to zero. Thus the default for VARMETHOD is RMS2.

Finally, the TOLERANCE option specifies the precision to which λ should be estimated.

Options: PRINT, METHOD, LAMBDA, VARMETHOD, TOLERANCE.
Parameters: Y, TREATMENTS, BLOCKS, UNITS, EFFECTS, SED, WNOISE, TREND, COMPONENTS.

Method

The model is fitted in a similar manner to that outlined in Besag & Kempton (1986), but the variance components have the parameterization used by Baird & Mead (1991) and are fitted by residual maximum likelihood (Gleeson & Cullis 1987) rather than maximum likelihood; also see Baird (1987). The optimization of the likelihood is done by golden section search on the profile likelihood for λ. Residuals are constructed by creating the smoothing matrix *S* that corresponds to the LV model fitted (Green *et al*. 1985).

The procedure uses a large amount of data space and computer time when the tuning parameter is estimated by REML. The speed is proportional to the number of rows multiplied by the square of the numbers of columns.

Action with `RESTRICT`

The procedure ignores any restrictions, for example, on `Y`, `TREATMENTS`, `BLOCKS` and `UNITS`.

References

Baird, D.B. (1987). A Genstat 5 procedure for a First Difference analysis. *Genstat Newsletter*, **19**, 40-47.

Baird, D.B. & Mead, R. (1991). The empirical efficiency and validity of two neighbour models. *Biometrics*, **47**, 1473-1487.

Besag, J.E. & Kempton R.A. (1986). Statistical analysis of field experiments using neighbouring plots. *Biometrics*, **42**, 231-251.

Gleeson, A.C. & Cullis, B.R. (1987). Residual maximum likelihood estimation of a neighbour model for field experiments. *Biometrics*, **43**, 277-288.

Green, P.J., Jennison, C. & Seheult. A.H. (1985). Analysis of field experiments by least squares smoothing. *Journal of the Royal Statistical Society, Series B*, **47**, 299-315.

Williams, E.R. (1986). A neighbour model for field experiments. *Biometrika*, **73**, 279-287.

See also

Directive: `VSTRUCTURE`.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

MAANOVA

Does analysis of variance for a single-channel microarray design (R.W. Payne & D.B. Baird).

Options

`PRINT` = *string tokens*	Controls printed output (`summary`, `monitoring`); default `*` i.e. none
`TREATMENTSTRUCTURE` = *formula*	Treatment formula for the analysis; if this is not set, the default is taken from the setting (which must already have been defined) of the `TREATMENTSTRUCTURE` directive
`BLOCKSTRUCTURE` = *formula*	Block formula for the analysis; if this is not set, the default is taken from any existing setting specified by the `BLOCKSTRUCTURE` directive and if neither has been set the design is assumed to be unstratified (i.e. to have a single error term)
`COVARIATE` = *variates*	Defines any covariates
`FACTORIAL` = *scalar*	Limit on the number of factors in a treatment term
`SAVETERMS` = *formula*	Treatment terms for which to save information; if this is not set, information is saved for all the treatment terms
`REPLICATION` = *pointer*	Pointer to tables saving the replication of the `SAVETERMS`
`SPREADSHEET` = *string tokens*	What results to save in spreadsheets (`aov`, `means`, `vcmeans`, `effects`, `vareffects`, `seeffects`, `contrasts`, `secontrasts`, `tcontrasts`, `prcontrasts`); default `*` i.e. none
`CONTRASTSLIMIT` = *scalar*	Limit on the order of a contrast of a treatment term; default 4
`DEVIATIONSLIMIT` = *scalar*	Limit on the number of factors in a treatment term for the deviations from its fitted contrasts to be retained in the model; default 9

Parameters

`Y` = *variates* or *pointers*	Y-variates for each analysis
`PROBES` = *factors* or *texts*	Defines the probe information for each analysis
`SLIDES` = *factors* or *texts*	Defines the slide information for each analysis
`CHECK` = *texts* or *variates*	Slide ID's that can be compared with the labels or levels of the `SLIDES` factor to ensure that the slide order is correct in each analysis
`IDS` = *texts*	Saves the probes names that have been generated to label the rows of the output structures from each analysis
`RESIDUALS` = *matrices*	Saves the residuals
`FITTEDVALUES` = *matrices*	Saves the fitted values
`MEANS` = *pointers*	Pointer to a matrix for each of the `SAVETERMS`, saving the means from each analysis
`VCMEANS` = *pointers*	Pointer to matrices saving variances and covariances for the means
`EFFECTS` = *pointers*	Pointer to matrices saving effects
`VAREFFECTS` = *pointers*	Pointer to variates saving unit variances for effects
`SEEFFECTS` = *pointers*	Pointer to variates saving effective standard errors of effects
`DF` = *pointers*	Pointer to variates saving degrees of freedom

SS = *pointers*	Pointer to variates saving sums of squares
MS = *pointers*	Pointer to variates saving mean squares
RDF = *pointers*	Pointer to variates saving degrees of freedom for the residual corresponding to each of the SAVETERMS
RSS = *pointers*	Pointer to variates saving residual sums of squares
RMS = *pointers*	Pointer to variates saving residual mean squares
VR = *pointers*	Pointer to variates saving variance ratios
PRVR = *pointers*	Pointer to variates saving probabilities for the variance ratios
CONTRASTS = *pointers*	Pointer to matrices saving estimates of contrasts
SECONTRASTS = *pointers*	Pointer to matrices saving standard errors of contrasts
TCONTRASTS = *pointers*	Pointer to matrices saving t-statistics for contrasts
PRCONTRASTS = *pointers*	Pointer to matrices saving probabilities for t-statistics of contrasts

Description

Procedure MAANOVA provides analysis of variance for microarray experiments with single-channel data. The experiment is assumed to consist of several slides, each of which represents a unit of the design. The BLOCKSTRUCTURE and TREATMENTSTRUCTURE options can specify block and treatment formulae (as in ordinary ANOVA) to define the models for the analysis of variance. If the TREATMENTSTRUCTURE option is not set, MAANOVA will use the model already defined by the TREATMENTSTRUCTURE directive, or will fail if that too has not been set. Similarly, if the BLOCKSTRUCTURE option is not set, MAANOVA will use the model (if any) previously defined by the BLOCKSTRUCTURE directive; these can both be omitted if there is only one error term (i.e. if the design is unstratified). The lengths of the block and treatment factors should be the same as the number of slides (and MAANOVA will give a failure diagnostic if this is not so). The FACTORIAL option sets a limit on the number of factors in a treatment term, as in the ANOVA directive. Similarly the CONTRASTSLIMIT and DEVIATIONSLIMIT options operate as the CONTRASTS and DEVIATIONS options of ANOVA.

The COVARIATE option can list any covariates for the analyses; if this is unset, the default is taken from any existing setting defined by the COVARIATE directive. The lengths of the covariates should be the same as the number of slides.

Each slide contains data on a (large) number of probes or genes. MAANOVA does a between-slide analysis of the data on each probe. So, it uses the mean value for any probe observations that are replicated within a slide, and prints a warning if the replication of any probe differs from slide to slide. The data from the slides are specified by the Y, PROBES and SLIDES parameters, and can be in either a stacked or an unstacked representation. With stacked data, the observations from all the slides are supplied by the Y parameter in a single variate. The SLIDES factor indicates the slide on which each observation was made, and the PROBES factor specifies the probe. With unstacked data, the Y parameter supplies a pointer with a variate for each slide, the PROBES factor or text specifies the probes (which must be in the same order on every slide), and the SLIDES factor can be omitted or may be a text defining the labels for each slide. The CHECK parameter can supply a text or variate to be compared with the labels or levels of the SLIDES factor to verify that the slides have been specified in the correct order.

The RESIDUALS and FITTEDVALUES parameters can save the residuals and fitted values, respectively, in a matrix with a row for each probe. The REPLICATION option saves a pointer containing the replication tables for the SAVETERMS. Parameters MEANS and EFFECTS save tables of means and effects from the analysis of each probe. The information is stored in a pointer with a matrix for each of the SAVETERMS. The matrices have a row for each probe, and the columns are labelled to show how they correspond to the cells of the table. (Note that their ordering is the same as the order in which the contents of the REPLICATION table is stored.)

Similarly SEEFFECTS saves effective standard errors for the effects, and VCMEANS saves the variances and covariances of the means. VAREFFECTS saves a pointer of variates storing the unit variances of the effects, obtained by the VARIANCE parameter of AKEEP. Parameters DF, SS, MS, RDF, RSS, RMS, VR and PRVR store information from the analysis of variance table, in pointers with a variate for each term and a unit for each probe. DF store the number of degrees of freedom for the relevant term (and probe), SS stores sums of squares, MS stores mean squares, VR stores variance ratios, and PRVR the corresponding probabilities. Similarly the RDF parameter stores the number of degrees of freedom for the appropriate residual for the term, RSS stores the residual sums of squares, and RMS the residual mean square.

Printed output is controlled by the PRINT option, with settings:

monitoring	to print a running total of the number of probes that have been analysed, and
summary	to print a summary of the significance levels found for the probes for each of the SAVETERMS.

The SPREADSHEET option allows you to save various output components in spreadsheets.

Options: PRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATE, FACTORIAL, SAVETERMS, REPLICATION, SPREADSHEET, CONTRASTSLIMIT, DEVIATIONSLIMIT.

Parameters: Y, PROBES, SLIDES, CHECK, IDS, RESIDUALS, FITTEDVALUES, MEANS, VCMEANS, EFFECTS, VAREFFECTS, SEEFFECTS, DF, SS, MS, RDF, RSS, RMS, VR, PRVR, CONTRASTS, SECONTRASTS, TCONTRASTS, PRCONTRASTS.

Method

The analyses are performed by the ANOVA directive.

Action with RESTRICT

If any of the y-variates is restricted, the analysis will involve only the units not excluded by the restriction.

See also

Procedures: AFFYMETRIX, FDRBONFERRONI, FDRMIXTURE, MABGCORRECT, MAEBAYES, MAREGRESSION, MARMA, MAROBUSTMEANS, MAVDIFFERENCE, MAVOLCANO, QNORMALIZE, AYPARALLEL.

Genstat Reference Manual 1 Summary section on: Microarray data.

MABGCORRECT

Performs background correction of Affymetrix slides (D.B. Baird).

Options

PRINT = *string token*	What to print (quantiles); default quan
METHOD = *string token*	Method of establishing grid background (mean, quantile); default mean
WEIGHTING = *string token*	Weighting method to use (affymetrix, distance); default affy
POWER = *scalar*	Power applied to distance; default 2 i.e. square
SMOOTH = *scalar*	Smoothing parameter applied to weights; default 100

Parameters

DATA = *variates* or *pointers*	Data values
SLIDES = *factors* or *texts*	Defines the slides
ROWS = *factors*	Defines the rows within each slide
COLUMNS = *factors*	Defines the columns within each slide
NEWDATA = *variates* or *pointers*	Saves the corrected values; if unset, they replace the original DATA values

Description

MABGCORRECT performs background correction of Affymetrix slides (or chips). The chip is divided into 16 zones in a 4 × 4 grid, and each spot has a weighted average of these 16 levels removed from it. The levels used are controlled by the METHOD options, with settings:

means	the means of the values below the 2% quantile are used as the background levels; and
quantiles	the actual 2% quantiles are used as the background levels.

The WEIGHTING option controls how the background levels are combined before removing them from each spot:

affymetrix	the weights are $1/(d^n + s)$, and
distance	the weights are $1/(\min(d^n, s)$,

where d is the distance from the spot to the zone centroid, the power n is defined by the POWER option (default 2), and the smoothing parameter s is defined by the SMOOTH option (default 100).

The data values are supplied by the DATA parameter, either as a variate, or as a pointer containing a variate for each slide. If DATA specifies a variate containing data for several slides, the SLIDES parameter must supply a factor to index the slides, and the ROWS and COLUMNS parameters supply factors to identify the rows and columns within the slides. If DATA specifies a pointer, the SLIDES parameter can be omitted, or it can supply a text to label the slides in the pointer. The ROWS and COLUMNS parameters then supply factors to identify the rows and columns for an individual slide, and all the slides must have the same layout. The corrected values can be saved in a variate or pointer, supplied by the NEWDATA parameter; if this is not set, the corrected values replace the original values in DATA.

Options: PRINT, METHOD, WEIGHTING, POWER, SMOOTH.
Parameters: DATA, SLIDES, ROWS, COLUMNS, NEWDATA.

See also

Procedures: AFFYMETRIX, FDRBONFERRONI, FDRMIXTURE, MAANOVA, MAEBAYES, MAREGRESSION, MARMA, MAROBUSTMEANS, MAVDIFFERENCE, MAVOLCANO, QNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MACALCULATE

Corrects and transforms two-colour microarray differential expressions (D.B. Baird).

Options

PRINT = *string token*	What to print (summary); default summ
BMETHOD = *string token*	How to correct for spot foreground for background values (subtract, smooth, none); default subtracts REDBACKGROUND and GREENBACKGROUND if set
TRANSFORMATION = *string token*	Type of transformation to apply to the red/green ratios (log, glog); default log
MINIMUM = *scalar*	Minimum value per channel; if RSDBACKGROUND and GSDBACKGROUND are supplied, this is the multiplier of these per spot, default 0
PERSPOTMINIMUM = *string token*	Use a single minimum value per spot rather than per slide (yes, no); default no
CONSTANTVALUE = *scalar*	Constant to add to red and green foreground values; default 0
DF = *scalar*	Degrees of freedom to use for loess smoothing of background; default 20

Parameters

RFOREGROUND = *variates* or *pointers*	Red foreground values per spot
GFOREGROUND = *variates* or *pointers*	Green foreground values per spot
RBACKGROUND = *variates* or *pointers*	Red background values per spot
GBACKGROUND = *variates* or *pointers*	Green background values per spot
RSDBACKGROUND = *variates* or *pointers*	Standard deviation of red background
GSDBACKGROUND = *variates* or *pointers*	Standard deviation of green background
SLIDES = *factors* or *texts*	Defines the slide to which each spot belongs for smoothing, or per slide minima
ROWS = *factors*	Defines the row position of each spot for background smoothing
COLUMNS = *factors*	Defines the column position of each spot for background smoothing
LOGRATIOS = *variates* or *pointers*	Saves the differential expression per spot
INTENSITIES = *variates* or *pointers*	Saves the intensity of each spot
RCORRECTED = *variates* or *pointers*	Saves the corrected red values per spot
GCORRECTED = *variates* or *pointers*	Saves the corrected green values per spot

Description

MACALCULATE calculates log-ratios from a two channel microarray. There are options for background correction and to set the channels to a minimum value to avoid large variation when the values in either channels get close to zero. The ratios are logged as the variation usually increases with the size of the mean. However, although the variation may be more uniform with the logged ratios, it may still show considerable heterogeneity with different levels of intensity.

The RFOREGROUND and GFOREGROUND parameters supply the values of the red/Cy5 and green/Cy3 foreground for each spot. (Cy5 and Cy3 are the technical names for the red and green fluorescent dyes.) These may be single variates. The SLIDES parameter then supplies a factor indicating the slide from which each spot was read. Alternatively, they may be pointers, containing a variate for each slide. The SLIDES parameter can then be omitted, or it can supply a text giving a label for each slide. The RBACKGROUND, GBACKGROUND, RSDBACKGROUND and GSDBACKGROUND parameters must be in the same format as the RFOREGROUND parameter.

The BMETHOD option controls the method to use for background correction, with settings:

none	no corrections;
subtract	subtract the red and green backgrounds, which must then be supplied by the RBACKGROUND and GBACKGROUND parameters;
smooth	perform a two-dimensional loess smoothing of the background values, and uses these for the correction: the ROWS and COLUMNS parameters must then supply factors to define the row and column positions of each spot.

If BMETHOD is unset, the red and green backgrounds are subtracted if available. The DF option specifies the number of degrees of freedom for the smoothing; if this is unset, the default is to use the square root of the number of spots observer per slide. Note that smoothing can be time consuming for large slides.

Log-ratios are undefined where spots may have foreground levels that are below their background levels, as you cannot take the log of a negative number. If this happens for both red and green, then there is no valid information on the level of differential expression and the log-ratio must be set to a missing value. However, there may be some useful information when one channel is above background and the other below and, if these log-ratios too are set to missing values, probes with a high level of differential expression may be missed. Some image analysis packages set a very large constant log-ratio for this, but this can bias the results as no differentiation is made between cases where the other colour is just above background, and other where the other colour is significantly above background.

You can set the MINIMUM option to a positive value to apply minimum values to the colours (for the default value of zero, no minimum values are applied). In the simplest situation, any red or green value less than MINIMUM is set to MINIMUM. For the alternatives, you need to supply variates giving the standard deviations of the red and green backgrounds around each spot, using the RSDBACKGROUND and GSDBACKGROUND parameters. Then, if option PERSPOTMINIMUM=yes, the background standard deviation for each colour is multiplied by MINIMUM, so that the minimum value depends on the quality of the background around each spot. With the default, PERSPOTMINIMUM=no, MINIMUM is multiplied by the background standard deviation for each colour, averaged within each slide.

The CONSTANTVALUE option can be used to reduce variability at the low end of the intensity range, as often the ratios become unstable as the foreground gets close to the background level. Adding a constant to both red and green will stabilize the log-ratios at low intensities, but with a potential loss of sensitivity for detecting differential expression. CONSTANTVALUE would normally be a positive value, although a negative value could be used if it were thought that the image analysis package was underestimating the background values. By default CONSTANTVALUE=0.

The LOGRATIOS parameter can save the calculated log-ratio or generalized log-ratio transformation (as requested by the TRANSFORMATION option) of the red and green values for each spot. The INTENSITIES parameter can save the intensity of each spot:

$$\textit{Intensity} = (\text{Log}(\textit{Red}) + \text{Log}(\textit{Green})) / (2 \times \text{Log}(2))$$

The RCORRECTED and GCORRECTED parameters save the corrected red and green values (i.e. after the application of any background corrections, minimum values and constants) for each

spot. If they have already been defined, the data structures supplied by LOGRATIOS, INTENSITIES, RCORRECTED and GCORRECTED must have the same type as that specified by the RFOREGROUND parameter (i.e. variates if RFOREGROUND is a variate, and pointers if RFOREGROUND is a pointer).

Options: PRINT, BMETHOD, TRANSFORMATION, MINIMUM, PERSPOTMINIMUM, CONSTANTVALUE, DF.

Parameters: RFOREGROUND, GFOREGROUND, RBACKGROUND, GBACKGROUND, RSDBACKGROUND, GSDBACKGROUND, SLIDES, ROWS, COLUMNS, LOGRATIOS, INTENSITIES, RCORRECTED, GCORRECTED.

See also

Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MAESTIMATE, MAHISTOGRAM, MAPCLUSTER, MAPLOT, MASCLUSTER, MASHADE, MAVOLCANO, MA2CLUSTER, MNORMALIZE.

Genstat Reference Manual 1 Summary section on: Microarray data.

MADESIGN

Assesses the efficiency of a two-colour microarray design (D.B. Baird).

Options

PRINT = *string tokens*	What to print (design, sed, secontrasts, vcovariance, summary); default desi, sed, seco, vcov, summ
DYEBIASMETHOD = *string token*	Whether to estimate dye bias effects (estimate, omit); default esti
SPREADSHEET = *string tokens*	What results to put in spreadsheets (sed, secontrasts, vcovariance); default sed, seco

Parameters

RED = *factors*	Targets on red dye
GREEN = *factors*	Targets on green dye
XCONTRASTS = *matrices*	Contrasts to estimate
SED = *symmetric matrices*	Saves standard errors of differences
VCOVARIANCE = *symmetric matrices*	Saves variance and covariances of treatments
SECONTRASTS = *symmetric matrices*	Saves standard errors of contrasts specified in XCONTRASTS

Description

MADESIGN assesses the efficency of a two-colour microarray design. The RED and GREEN parameters must supply factors defining which treatments are to be allocated to the red and green dyes of each slide, and the XCONTRASTS parameter can supply a matrix defining the contrasts of interest. The DYEBIASMETHOD option indicates whether dyebias is also to be estimated; by default DYEBIASMETHOD=esti.

The SED parameter can supply a symmetric matrix to save the standard errors of differences between treatment means that would arise from the design, assuming a residual mean square of one. The VCOVARIANCE parameter can save a symmetric matrix with variances and covariances of the treatment means, and the SECONTRASTS can save a variate eith the standard errors of the contrasts. The PRINT option controls which of these are printed, and the SPREADSHEET option allows you to put them into Genstat spreadsheets.

Options: PRINT, DYEBIASMETHOD, SPREADSHEET.
Parameters: RED, GREEN, XCONTRASTS, SED, VCOVARIANCE, SECONTRASTS.

See also

Procedures: AGBIB, AGLOOP, AGREFERENCE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MAEBAYES

Modifies t-values by an empirical Bayes method (D.B. Baird).

Options

PRINT = *string tokens*	What to print (estimates); default esti
PLOT = *string tokens*	What to plot (phistograms, thistograms, pvalues, tvalues); default * i.e. nothing
DATATYPE = *string token*	Type of data specified by the DATA parameter when it is a variate (means, tvalues); default tval
METHOD = *string token*	Type of test to use to form probability values (twosided, greaterthan, lessthan); default twos
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameters

DATA = *pointers* or *variates*	Pointers of variates or variates of means or t-values to be summarized
SD = *variates*	Supplies standard deviations of the data when DATA is a variate of means or t-values
DF = *variates* or *scalars*	Supplies degrees of freedom when DATA is a variate of means or t-values
SD0 = *scalars*	Saves the estimated prior standard deviation
DF0 = *scalars*	Saves the estimated number of degrees of freedom assigned to the prior standard deviation
TMODIFIED = *variates*	Saves the modified t-values
SDMODIFIED = *variates*	Saves the shrunken SD values
PMODIFIED = *variates*	Saves the modified probability values

Description

In a microarray experiment, as hundreds and often thousands of probes are being processed in parallel, there is a loss of power if you consider the variation of each probe in isolation. If this parallelism is used between the genes to gain extra information on the variation of an individual probe, then more powerful tests of the level of differential expression of a probe can be obtained. To do this, a prior distribution of the standard deviations (or equivalently the variances) over the probes is assumed. In particular, it is assumed that the reciprocal of the variance, s_p^2, of each probe is distributed as a multiple of a chi-square distribution with d_0 degrees of freedom, i.e. $1/s_p^2$ is distributed as $1/(d_0 \times s_0^2) \times \text{Chisquare}(d_0)$.

If the parameters of this distribution, the prior degrees of freedom d_0 and standard deviation s_0 are estimated, more information can be gained on an individual probe, by shrinking it towards the prior by an amount that depends on the amount of information in the standard deviation s_p of the probe (in this case its degrees of freedom d_p). The modified standard deviation $\tilde{s}_p$ is then given by:

$$\tilde{s}_p = \sqrt{((d_0^2 \times s_0^2 + d_p^2 \times s_p^2) / (d_0 + d_p))}$$

A modified t-test can then be performed using the modified standard deviation with $d_0 + d_p$ degrees of freedom. The method can also produce the probability values for tests that the differential expression differs from zero. The METHOD option selects the type of test i.e. two-sided, or for values greater than or less than zero (the default is two-sided).

The DATA parameter can supply a pointer containing one variate per slide, with the probes in the same position within each variate. The means and standard deviations are then be calculated from the raw data. Alternatively, DATA can supply a variate containing means or t-values for each

probe. The DATATYPE option should then indicate which of these has been given, the SD parameter should supply a variate containing the standard deviations for each probe, and the DF parameter should supply a variate with the numbers of degrees of freedom.

The estimated prior number of degrees of freedom d_0 and standard deviation s_0 can be saved, in scalars, by the D0 and S0 parameters. The TMODIFIED parameter can supply a variate to save the modified t-values, the SDMODIFIED parameter can save the shrunken SD values, and the PMODIFIED parameter can save the modified probability values.

By default, the estimates are printed, but this can be suppressed by setting option PRINT=*. The PLOT option controls what plots are produced, with settings:

phistograms	two histograms showing the modified and raw probabilities plotted on the same scale;
thistograms	two histograms showing the modified and raw t-values plotted on the same scale;
pvalues	a scatter plot of modified versus raw probabilities; and
tvalues	a scatter plot of modified versus t-values.

By default, nothing is plotted. You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE specifies then supplies a template for the file names.

Options: PRINT, PLOT, DATATYPE, METHOD, DEVICE, GRAPHICSFILE.
Parameters: DATA, SD, DF, SD0, DF0, TMODIFIED, SDMODIFIED, PMODIFIED.

Reference

Smyth, G.K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, **3**, No. 1, Article 3.
http://www.bepress.com/sagmb/vol3/iss1/art3

See also

Procedures: AFFYMETRIX, FDRBONFERRONI, FDRMIXTURE, MAANOVA, MABGCORRECT, MAREGRESSION, MARMA, MAROBUSTMEANS, MAVDIFFERENCE, MAVOLCANO, QNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MAESTIMATE

Estimates treatment effects from a two-colour microarray design (D.B. Baird).

Options

PRINT = *string tokens*	What to print (design, summary, monitoring); default desi, summ, moni
DYEBIASMETHOD = *string token*	Whether to estimate dye bias effects (estimate, omit); default esti
SPREADSHEET = *string tokens*	What results to put in spreadsheets (estimates, df, rsd, dyebias, seestimates, tvalues, probabilities, contrasts, secontrasts, tcontrasts, prcontrasts); default esti, df, rsd, dyeb, sees, tval, prob, cont, seco, tcon, prco

Parameters

LOGRATIOS = *variates* or *pointers*	Log-ratios
PROBES = *factors* or *texts*	Probes for the log-ratios
SLIDES = *factors* or *texts*	Slides for the log-ratios
REDTREATMENTS = *factors*	Targets on red dye for slides
GREENTREATMENTS = *factors*	Targets on green dye for slides
CHECK = *texts* or *variates*	Slide ID's of the red and green treatments for a check matching the slide order with the labels or levels of SLIDE
XCONTRASTS = *matrices*	Contrasts to estimate
IDPROBES = *texts*	Saves the probe names for each output row
DF = *variates*	Saves degrees of freedom for t-values
RSD = *variates*	Saves the residual standard deviation
DYEBIAS = *variates*	Saves estimated dye swap bias effects
ESTIMATES = *pointers*	Saves the estimates
SEESTIMATES = *pointers*	Saves the standard errors of the estimates
TVALUES = *pointers*	Saves t-values of the estimates
PROBABILITIES = *pointers*	Saves probabilities for the t-values
CONTRASTS = *pointers*	Saves estimates of the contrasts
SECONTRASTS = *pointers*	Saves the standard errors of the contrasts
TCONTRASTS = *pointers*	Saves t-values for the contrasts
PRCONTRASTS = *pointers*	Saves probabilities for the contrasts

Description

MAESTIMATE estimate effects from the within-slide differences between targets (or treatments). This information is contained in the log-ratios. Usually, these log-ratios will have normalized using the MNORMALIZE procedure. MAESTIMATE uses analysis of variance with a pooled error across the targets (i.e. treatments) for each probe (or gene). The normalization of each slide effectively removes the block effects, so the log-ratios now reflect the differences between treatments on each slide, and the constant represents the dye-swap effect for the probe. The DYEBIASMETHOD option controls whether or not the dye biases are estimated, and the XCONTRASTS parameter allows you to specify a matrix defining contrasts to estimate between treatments.

The log-ratios are supplied by the DATA parameter. If these are in a single variate, the SLIDE parameter should supply a factor to index the slides, and the PROBES parameter should index the probes or genes. Alternatively, you can supply a pointer containing a variate for each slide. The SLIDES parameter can then be omitted, or it can supply a text with an entry for each slide. The

PROBES parameter can supply either a factor or a text, defining the probes on a single slide, and all slides must have a common layout.

The REDTREATMENTS parameter should supply a factor to indicate the target assigned to the red dye. This is assumed to be the channel on the top of the log-ratios. This factor must have the same number of values as the number of levels of the Slides factor. Similarly, the GREENTREATMENTS parameter should supply a factor to indicate the target assigned to the green dye. The CHECK parameter can supply a text or variate identifying the slide in each unit of the REDTREATMENTS and GREENTREATMENTS factors. This can then be used to check that these units match the slides according to the labels or levels of the SLIDE factor. If the labels of the slides and check factor match, but are in a different order, the treatment factors will be sorted into the correct order, and a warning is given.

The other parameters allow you to save results from the analysis, and the SPREADSHEET option allows these to be put into Genstat spreadsheets.

Options: PRINT, DYEBIASMETHOD, SPREADSHEET.

Parameters: LOGRATIOS, PROBES, SLIDES, REDTREATMENTS, GREENTREATMENTS, CHECK, XCONTRASTS, IDPROBES, DF, RSD, DYEBIAS, ESTIMATES, SEESTIMATES, TVALUES, PROBABILITIES, CONTRASTS, SECONTRASTS, TCONTRASTS, PRCONTRASTS.

See also

Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAHISTOGRAM, MAPCLUSTER, MAPLOT, MASCLUSTER, MASHADE, MAVOLCANO, MA2CLUSTER, MNORMALIZE.

Genstat Reference Manual 1 Summary section on: Microarray data.

MAHISTOGRAM

Plots histograms of microarray data (D.B. Baird).

Options

SLIDES = *factor* or *text*	Defines the slides when the DATA variate contains data from more than one slide
SLIST = *variate* or *text*	Subset of slides to plot; default * i.e. all
NGROUPS = *scalar*	Number of groups into which to classify the DATA units; default 100
COLOUR = *text* or *scalar*	Colour to use for the bars of the histogram; default 'red'
TRANSFORMATION = *string token*	Whether to transform data to logarithms base 2 (log2, none); default none
SCALING = *string token*	Whether to use a common scale when not using Trellis plots (common, none); default comm
NROWS = *scalar*	Number of rows on a page in a trellis plot
NCOLUMNS = *scalar*	Number of columns on a page in a trellis plot
TITLE = *text*	Title for the graph
YTITLE = *text*	Title for the y-axis
XTITLE = *text*	Title for the x-axis
ARRANGEMENT = *string token*	Whether to use trellis or single plots when the DATA variate contains data from more than one slide (single, trellis); default trel
WINDOW = *scalar*	Window number for the graphs; default 3
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *
YMINIMUM = *scalar*	Minimum value on the y-axis of the histogram
YMAXIMUM = *scalar*	Maximum value on the y-axis of the histogram
XMINIMUM = *scalar*	Minimum value on the x-axis of the histogram
XMAXIMUM = *scalar*	Maximum value on the x-axis of the histogram

Parameter

DATA = *variates* or *pointers*	Data values to plot

Description

The data values are supplied using the DATA parameter. If you have data from several slides, you can set DATA either to a list of variates, or to a pointer to the variates (one for each slide), or to a single variate containing the data from all the slides. When multi-slide data are in a single DATA variate, the SLIDES option must supply a factor to identify the slides. If they are in a pointer, the SLIDES option can be omitted, or it can supply a text to identify the slides. By default, a histogram is produced for the data from every slide, but you can set option SLIST to a variate or text to define a subset of the slides to plot.

The NGROUPS option defines the number of groups into which to classify the DATA variate, i.e. the number of bars in each histogram (default 100). The COLOUR option defines the colour to use for the bars. You can set option TRANSFORMATION=log2 to transform the DATA values to logarithms base 2 before plotting. The ARRANGEMENT option controls whether the histograms for the slides are plotted singly, in separate frames, or together in a trellis arrangement (the default) when the DATA variate contains data from more than one slide. The SCALING option controls whether a common scale is used for the single plots, while the NROWS and NCOLUMNS options specify the numbers of rows and columns on the page in a trellis plot.

The TITLE, YTITLE and XTITLE options can supply titles for the graphs, the y-axes and the x-axes, respectively. The WINDOW option specifies the window to use (by default 3), and the KEYWINDOW option can specify a window for a key (by default there is none). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names. The YMAXIMUM, YMAXIMUM, XMINIMUM and XMAXIMUM options can be used to set the lower and upper limits on the y and x-axes of the histograms.

Options: SLIDES, SLIST, NGROUPS, COLOUR, TRANSFORMATION, SCALING, NROWS, NCOLUMNS, TITLE, YTITLE, XTITLE, ARRANGEMENT, WINDOW, DEVICE, GRAPHICSFILE, YMAXIMUM, YMAXIMUM, XMINIMUM, XMAXIMUM.
Parameter: DATA.

Action with RESTRICT
MAHISTOGRAM takes account of any restrictions on DATA or SLIDES.

See also
Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAESTIMATE, MAPCLUSTER, MAPLOT, MASCLUSTER, MASHADE, MAVOLCANO, MA2CLUSTER, MNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MANNWHITNEY

Performs a Mann-Whitney U test (S.J. Welham, N.M. Maclaren & H.R. Simpson).

Options

†PRINT = *string tokens*	Output required (test, ranks, hodgeslehmann, confidence); default test
METHOD = *string token*	Type of test required (twosided, greaterthan, lessthan); default twos
GROUPS = *factor*	Defines the samples for a two-sample test if the Y2 parameter is not set
CIPROBABILITY = *scalar*	Probability for the confidence interval for the median difference between the samples; default 0.95
CONTROL = *scalar* or *text*	Identifies the control group against which to make comparisons if GROUPS is set; default uses the reference level of GROUPS

Parameters

Y1 = *variates*	Identifier of the variate holding the first sample if Y2 is set, or both samples if Y2 is unset (the GROUPS option must then also be set)
Y2 = *variates*	Identifier of the variate holding the second sample
R1 = *variates*	Saves the ranks of the first sample if Y2 is set, or both samples if Y2 is unset
R2 = *variates*	Saves the ranks of the second sample if Y2 is set
STATISTIC = *scalars* or *tables*	Saves the test statistics *U*
PROBABILITY = *scalars* or *tables*	Probability values for the test statistics
SIGN = *scalars* or *tables*	Saves indicators: 1 if the first sample scores the highest ranks on average, 0 otherwise
†HODGESLEHMANN = *scalars* or *tables*	Saves the Hodges-Lehmann estimates for the differences in location of the two samples (i.e. the median differences between the samples)
LOWER = *scalars* or *tables*	Saves lower confidence values for the Hodges-Lehmann estimates
UPPER = *scalars* or *tables*	Saves upper confidence values for the Hodges-Lehmann estimates

Description

The Mann-Whitney *U* test is a test for differences in location between two samples. The data for the samples can be stored in two separate variates, and supplied by the parameters Y1 and Y2. Alternatively, they can be stored in a single variate, supplied by Y1, with the GROUPS option set to a factor to identify which unit belongs to each sample. The GROUPS option is ignored when the Y2 parameter is set. If GROUPS has more than 2 levels, each group is compared against a control group. You can define which level (or label) of GROUPS represents the control by setting the CONTROL option to a scalar or text. If CONTROL is not set, the reference level of GROUPS is used.

MANNWHITNEY calculates the test statistic *U*, along with its its associated probability value. An exact probability is calculated (using procedure PRMANNWHITNEYU) if the size of either sample is less than 51 and the statistic *U* is less than 10000; otherwise a Normal approximation is used. The statistic and the probability can be saved using the STATISTIC and PROBABILITY parameters respectively. Parameter SIGN holds an indicator which takes the value 1 if the ranks

in the first sample are higher on average than those in the second sample, and takes the value 0 otherwise. Usually STATISTIC, PROBABILITY and SIGN will save scalars, but they will save tables classified by the GROUPS factor when GROUPS is set to a factor with more than two levels. The ranks (with respect to the combined data set) for each sample can be saved using the R1 and R2 parameters.

Printed output is controlled by the PRINT option, with settings

test	test statistic and probability,
ranks	ranks (with respect to the whole data set) for each sample,
hodgeslehmann	Hodges-Lehmann estimate of the difference in the locations of the samples, with confidence limits, and
confidence	synonym of hodgeslehmann.

The probability for the confidence limits is specified by the CIPROBABILITY option; the default, of 0.95, gives a 95% interval. The calculation of the interval may be slow when there are ties amongst the values, as essentially MANNWHITNEY then has to invert the probability function. The Hodges-Lehmann estimates can be saved by the HODGESLEHMANN parameter. The lower and upper confidence values can be saved by the LOWER and UPPER parameters, respectively.

By default a two-sided test is done (to assess that samples are unequal) but the METHOD option can be set to greaterthan to test that the first sample is greater than the than the second, or lessthan to test that it is smaller.

Options: PRINT, METHOD, GROUPS, CIPROBABILITY.
Parameters: Y1, Y2, R1, R2, STATISTIC, PROBABILITY, SIGN, HODGESLEHMANN, LOWER, UPPER.

Method

The Mann-Whitney (or Wilcoxon) U-test is a two-sample test of location difference: i.e. a test of the null hypothesis that the two samples arise from distributions with the same mean vs. the alternative that the distribution means differ.

The test statistic U is formed using ranks found from the combined data set, and is taken to be the smaller of U_1 and U_2, where

$$U_k = n_1 \times n_2 + n_k \times (n_k+1) / 2 - R_k ; \; k=1,2$$

and n_k is the size of sample k, R_k is the sum of ranks for sample k. This score U_k can be interpreted as the number of times a rank score in the other sample precedes a score in sample k in the ranking. So the sample with the lowest score has, on average, smaller rank scores.

The PRMANNWHITNEYU procedure is used to calculate exact values of the probability for the test statistic when the size of either sample is less than 51 and the statistic U is less than 10000; otherwise a Normal approximation is used:

$$\text{Normal} = (n_1 \times n_2 / 2 - U) / \sqrt{\{ n_1 \times n_2 \times (n_1+n_2+1) / 12 \}}$$

If ties are present, the standard error of the Normal approximation (i.e. the denominator) must be calculated by:

$$\sqrt{\{ n_1 \times n_2 / (N \times (N-1)) \times ((N^3-N) / 12 - \sum_k T_k) \}}$$

where $T_k = (t_k^3 - t_k)/12$ and t_k is the number of observations with rank k. (See for example Siegel 1956, pages 116-127.)

The Hodges-Lehmann estimate is calculated as the median of all the differences between pairs of units (with one unit from each sample).

Action with RESTRICT

The variates Y1 and Y2 can be restricted, and in different ways. MANNWHITNEY uses only those units of each variate that are not excluded by their respective restrictions. Restrictions are also obeyed on Y1 and GROUPS, allowing RESTRICT to be used for example to limit the data to only

two groups when the GROUPS factor has more than two levels.

Reference

Siegel, S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Procedure: PRMANNWHITNEYU, SMANNWHITNEY, SIGNTEST, TTEST, WILCOXON.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

MANOVA

Performs multivariate analysis of variance and covariance (R.W. Payne & G.M. Arnold).

Options

PRINT = *string tokens*	Printed output required from the multivariate analysis of covariance (ssp, tests, permutationtest); default test
APRINT = *string tokens*	Printed output from the univariate analyses of variance of the y-variates (as for the ANOVA PRINT option); default *
UPRINT = *string tokens*	Printed output from the univariate unadjusted analyses of variance of the y-variates (as for the ANOVA UPRINT option); default *
CPRINT = *string tokens*	Printed output from the univariate analyses of variance of the covariates (as for the ANOVA CPRINT option); default *
TREATMENTSTRUCTURE = *formula*	Treatment formula for the analysis; if this is not set, the default is taken from the setting (which must already have been defined) by the TREATMENTSTRUCTURE directive
BLOCKSTRUCTURE = *formula*	Block formula for the analysis; if this is not set, the default is taken from any existing setting specified by the BLOCKSTRUCTURE directive and if neither has been set the design is assumed to be unstratified (i.e. to have a single error term)
COVARIATES = *variates*	Covariates for the analysis; by default MANOVA uses those listed by a previous COVARIATE directive (if any)
FACTORIAL = *scalar*	Limit on the number of factors in a treatment term
LRV = *pointer*	Contains elements first for the treatment terms and then the covariate term (if any), allowing the LRV's to be saved from one of the analyses; if a term is estimated in more than one stratum, the LRV is taken from the lowest stratum in which it is estimated
FPROBABILITY = *string token*	Printing of probabilities for F statistics (no, yes); default no
SELECTION = *string tokens*	Which test statistics to print when PRINT=test (lawleyhotellingtrace, pillaibartletttrace, roysmaximumroot, wilkslambda}; default lawl, pill, roys, wilk
NTIMES = *scalar*	Number of permutations to make when PRINT=perm; default 999
EXCLUDE = *factors*	Factors in the block model of the design whose levels are not to be randomized
SEED = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically

Parameter

Y = *variates*	Y-variates for an analysis

Description

Procedure MANOVA performs multivariate analysis of variance or covariance. The data variates are specified by the Y parameter.

The model for the design is specified by options of the procedure. TREATMENTSTRUCTURE specifies a model formula to define the treatment terms in the analysis; if this is unset, MANOVA will use the model already defined by the TREATMENTSTRUCTURE directive, or will fail if that too has not been set. BLOCKSTRUCTURE defines the underlying structure of the design, and MANOVA will use the model (if any) previously defined by the BLOCKSTRUCTURE directive if this is not set; these can both be omitted if there is only one error term (i.e. if the design is unstratified). The COVARIATES option specifies any covariates; by default MANOVA will take those already listed (if any) by the COVARIATE directive. The FACTORIAL option can be used to set a limit on the number of factors in the terms generated from the treatment formula.

The LRV option allows a pointer to be saved containing an LRV structure for each treatment term. When covariates have been specified, the pointer will also contain a final LRV structure for the covariate term. If a term is estimated in more than one stratum, the LRV is taken from the stratum that occurs last in the BLOCKTERMS pointer. The structures in the LRV hold the canonical variate loadings, roots and trace for the respective term.

The PRINT option indicates the output required from the multivariate analysis of covariance, with settings ssp to print the sums of squares and products matrices, tests to print the various test statistics, and permutationtest to calculate probabilities for the test statistics using a permutation test.

The SELECTION option controls which test statistics are given when PRINT=tests. The available statistics are Wilks' Lambda (with approximate F test), the Pillai-Bartlett trace, Roy's maximum root test and the Lawley-Hotelling trace. The default is to print them all.

By default, when PRINT=perm, MANOVA makes 999 random permutations and determines the probability of each test statistic from its distribution over these randomly generated datasets. The NTIMES option allows you to request another number of allocations, and the SEED option allows you to specify the seed to use for the random numbers used to make the permutations. The permutations are done by the RANDOMIZE directive, using the block model defined by the BLOCKSTRUCTURE option. The EXCLUDE option allows you to restrict the randomization so that one or more of the factors in the block model is not randomized. The most common situation where this is required is when one of the treatment factors involves time-order, which cannot be randomized.

The APRINT, UPRINT and CPRINT control output from the univariate analyses of each of the y-variates, corresponding to ANOVA options PRINT, UPRINT and CPRINT, respectively. FPROBABILITY controls whether or not probabilities are produced for F-ratios and for Chi-square variables in the analysis; by default these are omitted.

Options: PRINT, APRINT, UPRINT, CPRINT, TREATMENTSTRUCTURE, BLOCKSTRUCTURE, COVARIATES, FACTORIAL, LRV, FPROBABILITY, SELECTION, NTIMES, EXCLUDE, SEED.
Parameter: Y.

Method

The relevant theory, with formulae and references for the test statistics, can be found in Chatfield & Collins (1986, Chapter 9). The procedure analyses the data variates by ANOVA first as y-variates, and then as covariates in order to obtain the SSP matrices. The SSP matrices are then adjusted for the covariates, using matrix manipulation in CALCULATE, and LRV decompositions are done, before the test statistics are calculated (again using CALCULATE).

Action with `RESTRICT`
If any of the y-variates is restricted, the analysis will involve only the units not excluded by the restriction.

Reference
Chatfield, C. & Collins, A.J. (1986). *Introduction to Multivariate Analysis (revised edition)*. Chapman & Hall, London.

See also
Procedures: `RMULTIVARIATE`, `MVAOD`.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Repeated measurements.

MANTEL

Assesses the association between similarity matrices (J.W. McNicol, E.I. Duff & D.A. Elston).

Options

PRINT = *string token*	Controls printed output (test); default * i.e. none
METHOD = *string token*	The type of metric by which to compare the distance matrices (correlation, rankcorrelation, mantel); default corr
NPERMUTATIONS = *scalar*	The number of permutations of the units in the second distance matrix X on which the significance of the correlation between Y and X is to be based; default 100

Parameters

Y = *symmetric matrices*	The first distance or similarity matrix: the order of the units of this matrix is held fixed
X = *symmetric matrices*	The second distance or similarity matrix: the rows of X are permuted to allow the significance of the correlation between Y and X to be assessed
SEED = *scalars*	Random number seed for the permutations; default set by RANDOMIZE
M = *scalars*	Association between Y and X
MPERMUTED = *variates*	Associations between Y and the permuted X's
CUPROB = *scalars*	The proportion of MPERMUTED values greater than or equal to M
YOFFDIAGONAL = *variates*	Variate to save the off-diagonal elements of the distance/similarity matrix Y
XOFFDIAGONAL = *variates*	Variate to save the off-diagonal elements of the distance/similarity matrix X

Description

The extent to which two similarity/distance matrices describe the same relationships among the units can be measured by comparing their off-diagonal elements. The metrics to be used can be selected using the METHOD option: product-moment correlation (correlation), rank correlation (rankcorrelation) and SUM(X*Y) (Mantel). The last of these is the metric originally proposed by Mantel (1967). If the metric rankcorrelation is selected, the data are restricted to non-missing units and Spearman's rank correlation is used.

The significance of the association is assessed by a permutation test. The rows/columns of the second matrix are permuted at random and the association is recalculated for each permutation. Significance is estimated by the percentage of the permutations with association less/more than or equal to that of the original association.

If the number of random permutations, specified by the NPERMUTATIONS option, is set to a number greater than or equal to the total number of distinct permutations $d!$, where d is the dimension of the symmetric matrices, the full randomization test is implemented. Otherwise the rows/columns of the second matrix are permuted at random without regard to the duplication of specific permutations. By default, 100 permutations are done. The SEED parameter can supply a seed for the random numbers used to generate the random permutations. By default SEED=0, so the random numbers will continue any existing sequence, used earlier in the Genstat program, or be initialised by the RANDOMIZE directive.

The two matrices to be compared are specified by the Y and X parameters. The M parameter allows the value of the statistic for the original matrices to be saved, the MPERMUTED parameter saves the values from the permuted matrices, and the CUPROB parameter saves the proportion of

the permuted associations that are greater than the association between the original matrices. The off-diagonal elements of the matrices, on which the calculations are based, can be saved as variates using the XOFFDIAGONAL and YOFFDIAGONAL parameters.

The PRINT option can be set to test to print the values of M and CUPROB; by default there is no output.

Options: PRINT, METHOD, NPERMUTATIONS.
Parameters: Y, X, SEED, M, MPERMUTED, CLPROB, YOFFDIAGONAL, XOFFDIAGONAL.

Method

The off-diagonal elements of the symmetric matrices are transferred to variates by EQUATE, and the association is derived by CALCULATE for methods correlation and Mantel, and by SPEARMAN for rankcorr. If the full randomization test is used, all possible permutations of the rows of the second matrix are generated by PERMUTE. Otherwise a random set of permutations is generated by permuting an index to the rows of the matrix using RANDOMIZE. The permutations are then performed using CALCULATE, with the permuted indices as a qualified identifier.

References

Mantel, N. (1967). The detection of disease clustering and a generalized regression approach. *Cancer Research*, **27**, 209-220.

Manly, B.F.J. (1991). *Randomization and Monte Carlo Methods in Biology*. Chapman & Hall, London.

See also

Procedure: ECANOSIM.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

MAPCLUSTER

Clusters probes or genes with microarray data (D.B. Baird).

Options

PRINT = *string tokens*	What to print (cluster, groups, summary); default clus
PLOT = *string tokens*	What to plot (dendrogram, groups, meangroups); default dend, grou
METHOD = *string token*	Type of clustering to use (hierarchical, kmeans); default hier
DMETHOD = *string token*	Distance method to use for hierarchical clustering (euclidean, cityblock); default eucl
LMETHOD = *string token*	What type of link to use in hierarchal clustering (singlelink, nearestneighbour, completelink, furthestneighbour, averagelink, mediansort, groupaverage); default aver
CRITERION = *string token*	Criterion to use in forming groups when LMETHOD=kmeans (sums, predictive, within, Mahalanobis); default sums
NGROUPS = *scalar*	Number of groups to form when LMETHOD=kmeans
GTHRESHOLD = *scalar*	Grouping threshold for forming groups from the dendrogram; default *
PERCENT = *scalar*	Percentage of the probes/genes to use; default 100
DTITLE = *text*	Title for the dendrogram
GTITLE = *text*	Title for the groups plot
ARRANGEMENT = *string token*	Whether to use a trellis or single plot (single, trellis); default trel
WINDOW = *scalar*	Window number for the graphs; default 3
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *
SPREADSHEET = *string token*	What results to put in spreadsheets (top%probes); default * i.e. none

Parameters

DATA = *variates* or *pointers*	Data values (i.e. log-ratios)
SLIDES = *factors*, *texts* or *variates*	Identifies the slides
PROBES = *factors*, *texts* or *variates*	Identifies the probes or genes
SIMILARITY = *symmetric matrices*	Saves the pair-wise similarities between probes or genes when METHOD=hier
GROUPS = *factors*	Saves the group membership for each probe
AMALGAMATIONS = *matrices*	Saves the probe or gene amalgamation data when METHOD=hier

Description

MAPCLUSTER clusters probes (which may be thought of as representing genes) together on the similarity of their responses over a number of slides or target effects. The METHOD option specifies whether the clustering is hierarchical, or non-hierarchical using the k-means algorithm. A range of clustering criteria are available for each method (options DMETHOD, LMETHOD and CRITERION). The probes are grouped together so that the responses of each group are similar,

with the groups as distinct as possible. For the hierarchical clustering, the allocation to groups is specified by using the GTHRESHOLD option to provide a threshold for the levels of similarity within a group. The dendrogram is then cut at this level, generating an unknown number of groups. For the k-means algorithm, the number of groups must be specified using the NGROUPS option. The group membership can be saved by the GROUPS parameter.

The log-ratios are supplied by the DATA parameter. If these are in a single variate, the SLIDE parameter should supply a factor to index the slides, and the PROBES parameter should index the probes or genes. Alternatively you can supply a pointer containing a variate for each slide. The slides factor is then not required; if it is given it should just have one entry for each slide in the order of the variates in the pointer. The PROBES factor is that for a single slide, and all slides must have a common layout.

The PLOT option allows you to plot a dendrogram for the hierarchical cluster analyses, but for a large number of probes this is less useful as individual probes cannot be read. The responses of each probe across the targets/slides can also be plotted in a shade plot, but for large numbers of probes this is slow, in which case the mean response for each group can be plotted instead. A spreadsheet containing the grouped data can also be saved using the SPREADSHEET option.

With large numbers of probes, the limit of RAM can be quickly reached, so option PERCENT can be set so that only cluster probes with the largest mean absolute responses are clustered.

By default the plots for the groups are displayed in a trellis arrangement, but you can set option ARRANGEMENT=single to display them separately, in single plots. The DTITLE and GTITLE options can supply titles for the dendrogram and groups plot, respectively, and the WINDOW option specifies the window to use (by default 3). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

Options: PRINT, PLOT, METHOD, DMETHOD, LMETHOD, CRITERION, NGROUPS, GTHRESHOLD, PERCENT, DTITLE, GTITLE, ARRANGEMENT, WINDOW, DEVICE, GRAPHICSFILE, SPREADSHEET.
Parameters: DATA, SLIDES, PROBES, SIMILARITY, GROUPS, AMALGAMATIONS.

Action with RESTRICT
Any restrictions on the DATA variates are removed.

See also
Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAESTIMATE, MAHISTOGRAM, MAPLOT, MASCLUSTER, MASHADE, MAVOLCANO, MA2CLUSTER, MNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MAPLOT

Produces two-dimensional plots of microarray data (D.B. Baird).

Options

SLIDES = *factor* or *text*	Defines the slides when the X and Y variates contain data from more than one slide
SLIST = *variate* or *text*	Subset of slides to plot; default * i.e. all
GROUPS = *factor*	Specifies groups within slides
COLOURS = *text*, *scalar* or *variate*	Colours to use for the plots
SYMBOLS = *scalar* or *variate*	Symbols to use for the plots
REFERENCELINECHOICE = *string token*	Reference line to include (identity, zero, none); default none
TRANSFORMATION = *string token*	Whether to transform data to logarithms base 2 (log2, none); default none
SCALING = *string token*	Whether to use a common scale when not using Trellis plots (common, none); default comm
BANDS = *string token*	Whether to plot approximate confidence bands (confidence, none); default none
SMOOTHEDMEAN = *string token*	Whether to plot spline smooth of mean (yes, no); default no
NROWS = *scalar*	Number of rows on a page in a trellis plot
NCOLUMNS = *scalar*	Number of columns on a page in a trellis plot
TITLE = *text*	Title for the graph
YTITLE = *text*	Title for the y-axis
XTITLE = *text*	Title for the x-axis
ARRANGEMENT = *string token*	Whether to use trellis, single or multiple plots when the X and Y variates contain data from more than one slide (separate, overlaid, trellis); default trel
WINDOW = *scalar*	Window number for the graphs; default 3
KEYWINDOW = *scalar*	Window number for the key; default 0
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameters

Y = *variates* or *pointers*	Y-coordinates
X = *variates* or *pointers*	X-coordinates

Description

MAPLOT produces two-dimensional plots of microarray data or transformed data using log base 2. The x- and y-coordinates are supplied, in variates or pointers, using the X and Y parameters, respectively. If you have data from several slides, you can set DATA either to a list of variates, or to a pointer to the variates (one for each slide), or to a single variate containing the data from all the slides. When multi-slide data are in a single DATA variate, the SLIDES option must supply a factor to identify the slides. If they are in a pointer, the SLIDES option can be omitted, or it can supply a text to identify the slides. By default, a plots are produced for the data from every slide, but you can set option SLIST to a variate or text to define a subset of the slides to plot.

The REFERENCELINECHOICE option allows you to include either an identity reference line or a horizontal reference at zero. The BANDS option includes approximate confidence bands, and the SMOOTHEDMEAN option adds a spline smooth of the mean. By default none of these are

plotted.

The COLOURS option can be set to a text, scalar or variate to define the colour(s) to use for the plots, and the SYMBOLS option can be set to a scalar or variate to define the symbols. The TITLE, YTITLE and XTITLE options can supply titles for the graph, the y-axis and the x-axis, respectively. The WINDOW option specifies the window to use (by default 3), and the KEYWINDOW option can specify a window for a key (by default there is none). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

Options: SLIDES, SLIST, GROUPS, COLOUR, SYMBOLS, REFERENCELINECHOICE, TRANSFORMATION, SCALING, NROWS, NCOLUMNS, TITLE, YTITLE, XTITLE, ARRANGEMENT, WINDOW, KEYWINDOW, DEVICE, GRAPHICSFILE.
Parameters: Y, X.

Action with RESTRICT

MAPLOT takes account of any restrictions on X, Y or SLIDES.

See also

Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAESTIMATE, MAHISTOGRAM, MAPCLUSTER, MASCLUSTER, MASHADE, MAVOLCANO, MA2CLUSTER, MNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MAREGRESSION

Does regressions for single-channel microarray data (P. Brain, R.W. Payne & D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output (model, summary); default * i.e. none
TERMS = *formula*	Defines the regression model over the slides
WEIGHTS = *variate*	Weights for the regression; default 1
OFFSET = *variate*	Offset; default * i.e. none
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalar*	Limit for expansion of model terms; default 3
FULL = *string token*	Whether to assign all possible parameters to factors and interactions (yes, no); default no
POOL = *string token*	Whether to pool the information on each term in the analysis of variance (yes, no); default no
RMETHOD = *string token*	Type of residuals to form (deviance, Pearson, simple); default devi
SPREADSHEET = *string tokens*	What results to save in a book of spreadsheets (aov, residuals, fittedvalues, estimates, se, testimates, prestimates); default * i.e. none

Parameters

Y = *variates* or *pointers*	Y-values for each set of analyses
PROBES = *factors* or *texts*	Defines the probe information for each analysis
SLIDES = *factors* or *texts*	Defines the slide information for each analysis
CHECK= *texts* or *variates*	Slide ID's that can be compared with the labels or levels of the SLIDES factor to ensure that the slide order is correct in each analysis
IDS = *texts*	Saves the probes names that have been generated to label the rows of the output structures from each analysis
RESIDUALS = *matrices*	Saves residuals from each set of analyses
FITTEDVALUES = *matrices*	Saves fitted values from each set of analyses
ESTIMATES = *matrices*	Saves estimates from each set of analyses
SE = *matrices*	Saves s.e.'s of estimates
TESTIMATES = *matrices*	Saves t-statistics of estimates
PRESTIMATES = *matrices*	Saves t-probabilities of estimates
DF = *pointers*	Saves degrees of freedom for the model terms or variates in each analysis of variance
SS = *pointers* or *variates*	Saves sums of squares for the model terms in each analysis of variance
MS = *pointers* or *variates*	Saves mean squares for the model terms in each analysis of variance
RDF = *variates*	Saves degrees of freedom from the "residual" lines in each analysis of variance
RSS = *variates*	Saves sums of squares from the "residual" lines
RMS = *variates*	Saves mean squares from the "residual" lines
TDF = *variates*	Saves degrees of freedom from the "total" lines in each analysis of variance
TSS = *variates*	Saves sums of squares from the "total" lines
TMS = *variates*	Saves mean squares from the "total" lines

VR = *pointers* or *variates*	Saves variance ratios for the model terms in each analysis of variance
PRVR = *pointers* or *variates*	Saves probabilities of the variance ratios

Description

Procedure MAREGRESSION does regression analyses for microarray experiments with single-channel data. The experiment is assumed to consist of several slides, each of which represents a unit of the design. The model for the regressions is specified by the TERMS, WEIGHTS, OFFSET, CONSTANT, FACTORIAL and FULL options, which operate exactly as in ordinary regression (see the MODEL, TERMS and FIT directives). The lengths of the factors and variates in the model should be the same as the number of slides (and MAREGRESSION will give a failure diagnostic if this is not so).

Each slide contains data on a (large) number of probes or genes. MAREGRESSION does a between-slide analysis of the data on each probe. So, it uses the mean value for any probe observations that are replicated within a slide, and prints a warning if the replication of any probe differs from slide to slide. The data from the slides are specified by the Y, PROBES and SLIDES parameters, and can be in either a stacked or an unstacked representation. With stacked data, the observations from all the slides are supplied by the Y parameter in a single variate, the SLIDES factor indicates the slide on which each observation was made, and the PROBES factor specifies the probe. With unstacked data, the Y parameter supplies a pointer with a variate for each slide. The PROBES factor or text specifies the probes (which must be in the same order on every slide). The SLIDES factor can be omitted, or it can supply a text defining a label for each slide. The CHECK parameter can supply a text or variate to be compared with the labels or levels of the SLIDES factor, to verify that the slides have been specified in the correct order.

The RESIDUALS and FITTEDVALUES parameters allow you to save the residuals and fitted values from the regressions. These are defined as matrices, with a row for each probe, and a column for each slide. The RMETHOD option indicates what sort of residual to form, as in the other Genstat regression commands. By default, standardized residuals are formed, but you can set RMETHOD=simple to form simple residuals instead.

The ESTIMATES, SE, TESTESTIMATES and PRESTIMATES parameters save the estimates, standard errors, t-statistics and t-probabilities for the parameters in the regression model. These are defined as matrices, with a row for each probe, and a column for each parameter.

The DF, SS, MS, RDF, RSS, RMS, TDF, TSS, TMS, VR and PRVR parameters store information from the analysis of variance table. (DF, SS, MS, VR and PRVR are from the "regression" line, RDF, RSS and RMS are from the "residual" line, and TDF, TSS and TMS are from the "total" line.) With the default setting no of the POOL option each of these is a pointer containing a variate for each term in the TERMS formula. The variates each have a unit for every probe. Alternatively, if you set POOL=yes, the parameters each have a single variate, with the values pooled over the terms.

Printed output is controlled by the PRINT option, with settings:

model	for a description of the regression model, and
summary	for a summary of the significance levels found over the probes for each parameter in the model.

The SPREADSHEET option allows you to save the various output components in spreadsheets.

Options: PRINT, TERMS, WEIGHTS, OFFSET, CONSTANT, FACTORIAL, FULL, RMETHOD, SPREADSHEET.

Parameters: Y, PROBES, SLIDES, CHECK, IDS, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, TESTIMATES, PRESTIMATES, DF, SS, MS, RDF, RSS, RMS, TDF, TSS, TMS, VR, PRVR.

Method
The analyses are performed by the FIT directive and by matrix calculations.

Action with RESTRICT
If any of the y-variates is restricted, the analysis will involve only the units not excluded by the restriction.

See also
Procedures: AFFYMETRIX, FDRBONFERRONI, FDRMIXTURE, MAANOVA, MABGCORRECT, MAEBAYES, MARMA, MAROBUSTMEANS, MAVDIFFERENCE, MAVOLCANO, QNORMALIZE, RYPARALLEL.
Genstat Reference Manual 1 Summary section on: Microarray data.

MARMA

Calculates Affymetrix expression values (D.B. Baird).

Options

PRINT = *string token*	What to print (estimates, monitoring); default esti
METHOD = *string token*	Method of establishing grid background (rma, rma2, moments, maximumlikelihood); default rma
NORMALIZED = *string token*	Whether slides have been normalized (yes, no); default no

Parameters

DATA = *variates* or *pointers*	Perfect-match data
SLIDES = *factors* or *texts*	Defines the slides
NEWDATA = *variates* or *pointers*	Saves the corrected values; if this is unset, they replace the original values in DATA
ESTIMATES = *variates*	Saves the estimated parameters of the model

Description

MARMA estimates expression values over the perfect match (PM) values for each probe on Affymetrix slides/chips. On Affymetrix chips, each probe has 8-20 pairs of DNA sequences with a central base changed between perfect match and mismatch sequences. With the robust means analysis (RMA) the value for the probe level of expression is taken as an average over the perfect-match spots, after removing any estimated noise effects, ignoring the mismatch spots. The DATA parameter supplies the PM values from the slides, in either a stacked or an unstacked representation. With stacked data, the observations from all the slides are supplied by the DATA parameter in a single variate, and the SLIDES factor indicates the slide on which each observation was made. With unstacked data, the DATA parameter supplies a pointer with a variate for each slide. The SLIDES parameter can be omitted, or can supply a text defining a label for each slide.

The default setting of the METHOD option uses the RMA probe-level model, introduced by Irizarry *et al*. (2003), which uses only PM information and transforms the values based on a kernel density estimate of the PM distribution. The rma2 setting uses an adaptation of the algorithm, which fits the kernel density to a truncated distribution of the perfect-match values, with the truncation point based on an initial kernel density estimate. The moments setting uses the method of moments, and the maximumlikelihood setting uses maximum likelihood to estimate the background noise in the PM distribution.

Options: PRINT, METHOD, NORMALIZED.
Parameters: DATA, SLIDES, NEWDATA, ESTIMATES.

References

Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U. & Speed, T.P. (2003). Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, Number 2, 249-264.

See also

Procedures: AFFYMETRIX, FDRBONFERRONI, FDRMIXTURE, MAANOVA, MABGCORRECT, MAEBAYES, MAREGRESSION, MAROBUSTMEANS, MAVDIFFERENCE, MAVOLCANO, QNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MAROBUSTMEANS

Does a robust means analysis for Affymetrix slides (D.B. Baird).

Options

TRANSFORMATION = *string token*	How to transform the data (log2, none); default none
MAXCYCLE = *scalar*	Maximum number of iterations; default 50
TOLERANCE = *scalar*	Tolerance for convergence; default 0.0001

Parameters

DATA = *variates* or *pointers*	Expression data to be summarized
SLIDES = *factors* or *texts*	Defines the slides
PROBES = *factors*	Defines the probes
IDPROBES = *factors*	Saves the probe IDs
MEDIANS = *variates* or *pointers*	Saves the robust means
SEM = *variates* or *pointers*	Saves approximate standard errors of the robust means

Description

MAROBUSTMEANS calculates the medians of probe expression values across a series of Affymetrix slides. The median expression level for each slide is estimated and removed in a iterative fashion using an algorithm called the median polish algorithm. An approximate standard error of the mean can be calculated which is defined as 1.483 times the median absolute deviation from the probe medians.

The expression intensity values are supplied by the DATA parameter. If these are in a single variate, the SLIDE parameter should supply a factor to index the slides, and the PROBES parameter should index the probes or genes. Alternatively, you can supply a pointer containing a variate for each slide. The SLIDES parameter can be omitted, or it can supply a text to label the slides in the pointer. The PROBES factor refers to a single slide, and all the slides must have a common layout.

Options: TRANSFORMATION, MAXCYCLE, TOLERANCE.
Parameters: DATA, SLIDES, PROBES, IDPROBES, MEDIANS, SEM.

See also

Procedures: AFFYMETRIX, FDRBONFERRONI, FDRMIXTURE, MAANOVA, MABGCORRECT, MAEBAYES, MAREGRESSION, MARMA, MAVDIFFERENCE, MAVOLCANO, QNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MASCLUSTER

Clusters microarray slides (D.B. Baird).

Options

PRINT = *string tokens*	What to print (cluster, pco, correlations, distances); default clus, pco, corr, dist
PLOT = *string tokens*	What to plot (dendrogram, mst); default dend, mst
DMETHOD = *string token*	What distance method to use to form the similarity matrix (correlation, euclidean, cityblock); default corr
PERCENT = *scalar*	Percentage of the probes/genes to use to calculate correlations; default 100
DTITLE = *text*	Title for the dendrogram
MTITLE = *text*	Title for the minimum spanning tree
WINDOW = *scalar*	Window number for the graphs; default 3
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameters

DATA = *variates* or *pointers*	Data values (i.e. log-ratios)
SLIDES = *factors*, *texts* or *variates*	Identifies the slides
PROBES = *factors*, *texts* or *variates*	Identifies the probes or genes
CORRELATION = *symmetric matrices*	Saves the correlation matrix
DISTANCE = *symmetric matrices*	Saves the distance matrix

Description

MASCLUSTER clusters microarray slides (or targets) together on the similarity of their responses over a number of probes or genes. The slides are grouped together so that the pattern of responses over the probes/genes are similar, with the groups as distinct as possible.

The DMETHOD option specifies the distance method to use to form the similarity matrix: either correlation (default), euclidean, or cityblock.

With large numbers of probes or genes, many may be non-informative, only being subject to random variation. So the PERCENT option controls the percentage of the probes to use: if PERCENT is less than the default 100, MASCLUSTER uses only the top PERCENT of probes according to their mean absolut response.

The log-ratios are supplied by the DATA parameter. If these are in a single variate, the SLIDE parameter should supply a factor to index the slides, and the PROBES parameter should index the probes or genes. Alternatively you can supply a pointer containing a variate for each slide. The SLIDES factor is then not required; if it is given it should just have one entry for each slide in the order of the variates in the pointer. The PROBES factor is that for a single slide, and all slides must have a common layout.

The DTITLE and MTITLE options can supply titles for the plots of the dendrogram and minimum spanning tree, respectively, and the WINDOW option specifies the window to use (by default 3). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

Options: PRINT, PLOT, DMETHOD, PERCENT, DTITLE, MTITLE, WINDOW, DEVICE, GRAPHICSFILE.

Parameters: DATA, SLIDES, PROBES, CORRELATION, DISTANCE.

Action with RESTRICT
Any restrictions on the DATA variates are removed.

See also
Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAESTIMATE, MAHISTOGRAM, MAPCLUSTER, MAPLOT, MASHADE, MAVOLCANO, MA2CLUSTER, MNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MASHADE

Produces shade plots to display spatial variation of microarray data (D.B. Baird).

Options

SLIDES = *factor* or *text*	Defines the slides when the DATA variate contains data from more than one slide
SLIST = *variate* or *text*	Subset of slides to plot; default * i.e. all
ROWS = *factor* or *variate*	Row to which each DATA unit belongs
COLUMNS = *factor* or *variate*	Column to which each DATA unit belongs
COLOURS = *text*, *scalar* or *variate*	Colours to use for the plots; default !t(blue,red)
SHADING = *string token*	Shading scale (natural, percentiles); default natu
TITLE = *text*	Title for the graph
YTITLE = *text*	Title for the y-axis
XTITLE = *text*	Title for the x-axis
WINDOW = *scalar*	Window number for the graphs; default 3
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameter

DATA = *variates* or *pointers*	Values for each shade plot

Description

The data values are supplied, in either one or several variates or a pointer to variates, using the DATA parameter. If you have data from several slides, you can set DATA either to a list of variates, or to a pointer to the variates (one for each slide), or to a single variate containing the data from all the slides. When multi-slide data are in a single DATA variate, the SLIDES option must supply a factor to identify the slides. If they are in a pointer, the SLIDES option can be omitted, or it can supply a text to identify the slides. By default, a plot is produced for the data from every slide, but you can set option SLIST to a variate or text to define a subset of the slides to plot. The ROWS and COLUMNS parameters supply a factor or variate, to define the row and column positions within each slide.

The COLOURS option supplies a text, scalar or variate to define the colours to use for the plots. The default of !(blue,red) uses colours ranging from blue to red. The SHADING option chooses whether to allocate the DATA values to shades by percentiles, thus giving similar amounts of each colour on the plots, or by their actual values (the default).

The TITLE, YTITLE and XTITLE options can supply titles for the graph, the y-axis and the x-axis, respectively. The WINDOW option specifies the window to use (by default 3), and the KEYWINDOW option can specify a window for a key (by default there is none). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

Options: SLIDES, SLIST, ROWS, COLUMNS, COLOURS, SHADING, TITLE, YTITLE, XTITLE, WINDOW, DEVICE, GRAPHICSFILE.
Parameter: DATA.

Action with RESTRICT

Restrictions are ignored.

See also

Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAESTIMATE, MAHISTOGRAM, MAPCLUSTER, MAPLOT, MASCLUSTER, MAVOLCANO, MA2CLUSTER, MNORMALIZE.

Genstat Reference Manual 1 Summary section on: Microarray data.

MAVDIFFERENCE

Applies the average difference algorithm to Affymetrix data (D.B. Baird).

Options

PRINT = *string token*	Whether to print monitoring information (monitoring); default *
SDLIMIT = *scalar*	Maximum number of iterations; default 50

Parameters

DATA = *variates* or *pointers*	Data values
GROUPS = *factors*	Groupings of the data values
MEANS = *variates*	Saves the means
SE = *variates*	Saves standard errors

Description

MAVDIFFERENCE uses the average difference algorithm to remove extreme values from Affymetrix data. These are defined as values more than option SDLIMIT standard deviations from the mean.

The data values are specified by the DATA parameter. They can be in a single variate, with any groupings (corresponding to different genes or probes) specified by the GROUPS parameter. Alternatively, they can be in separate variates, one for each group. The MEANS parameter saves the means. The SE parameter saves the estimated standard deviation when there are no groups, or the standard error of the mean when there are groups.

Options: PRINT, SDLIMIT.
Parameters: DATA, GROUPS, MEANS, SE.

Action with RESTRICT

MAVDIFFERENCE takes account of any restrictions on DATA or GROUPS.

See also

Procedures: AFFYMETRIX, FDRBONFERRONI, FDRMIXTURE, MAANOVA, MABGCORRECT, MAEBAYES, MAREGRESSION, MARMA, MAROBUSTMEANS, MAVOLCANO, QNORMALIZE.
Genstat Reference Manual 1 Summary section on: Microarray data.

MAVOLCANO

Produces volcano plots of microarray data (D.B. Baird).

Options

NGROUPS = *scalar*	Number of groupings for a Z variate; default 10
COLOURS = *text*, *scalar* or *variate*	Colours to use for the plots; default !t(blue,red)
SYMBOL = *scalar*	Symbol to use for the points; default 1
TRANSFORMATION = *string token*	Whether to transform data to logarithms base 2 (log10, none); default log10
TITLE = *text*	Title for the graph
YTITLE = *text*	Title for the y-axis
XTITLE = *text*	Title for the x-axis
WINDOW = *scalar*	Window number for the graphs; default 3
KEYWINDOW = *scalar*	Window number for the graphs; default 0
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameters

X = *variates*	X-coordinates
Y = *variates* or *factors*	Y-coordinates
Z = *variates* or *factors*	Z-coordinates

Description

MAVOLCANO produces volcano plots of microarray data. The y-coordinates are specified by the Y parameter. Typically these are probabilities, in which case they are usually transformed to −log10(Y). This transformation is thus applied by default, but you can set option TRANSFORMATION=none to suppress it. Less commonly the y-coordinates may be t-values. The x-coordinates are specified by the X parameter. These are usually measures of differential expression such as log-ratios.

The Z parameter can specify a variate or factor to use to colour the points on the graph. If this is a variate, the values are grouped into the number of percentiles specified by the NGROUPS option. The COLOURS option supplies a text, scalar or variate to define the colours to use for the plots. The default of !t(blue,red) uses colours ranging from blue to red. The SYMBOL option defines the symbol to use for the points (default 1).

The TITLE, YTITLE and XTITLE options can supply titles for the graph, the y-axis and the x-axis, respectively. The WINDOW option specifies the window to use (by default 3), and the KEYWINDOW option can specify a window for a key (by default there is none). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

Options: NGROUPS, COLOURS, SYMBOL, TRANSFORMATION, TITLE, YTITLE, XTITLE, WINDOW, KEYWINDOW, DEVICE, GRAPHICSFILE.
Parameters: X, Y, Z.

Action with RESTRICT

MAVOLCANO takes account of any restrictions on X, Y or Z.

See also
Procedures: DPROBABILITY, FDRBONFERRONI, FDRMIXTURE.
Genstat Reference Manual 1 Summary sections on: Microarray data, Graphics.

MA2CLUSTER

Performs a two-way clustering of microarray data by probes (or genes) and slides (D.B. Baird).

Options

PRINT = *string tokens*	What to print (cluster, groups, summary); default clus
PLOT = *string tokens*	What to plot (dendrogram, shade, meanshade); default dend, shad
METHOD = *string token*	Type of clustering to use (hierarchical, kmeans); default hier
DMETHOD = *string token*	Distance method to use for hierarchical clustering (euclidean, cityblock); default eucl
LMETHOD = *string token*	What type of link to use in hierarchal clustering (singlelink, nearestneighbour, completelink, furthestneighbour, averagelink, mediansort, groupaverage); default aver
CRITERION = *string token*	Criterion to use in forming groups when LMETHOD=kmeans (sums, predictive, within, Mahalanobis); default sums
PNGROUPS = *scalar*	Number of probe groups to form when LMETHOD=kmeans
SNGROUPS = *scalar*	Number of target (slide) groups to form when LMETHOD=kmeans
GTHRESHOLD = *scalar*	Grouping threshold for forming probe groups from the dendrogram; default *
SGTHRESHOLD = *scalar*	Grouping threshold for forming target (slide) groups from the dendrogram; default *
MINOBSERVATIONS = *scalar*	Smallest number of observations before probes are dropped; default *
PERCENT = *scalar*	Percentage of the probes/genes to use; default 100
STANDARDIZE = *string token*	Allows you to centre the values by slide and probe (centre); default * i.e. no centring
COLOURS = *text*, *scalar* or *variate*	Colours to use for shade plot; default !t(blue,red)
DTITLE = *text*	Title for the dendrogram
STITLE = *text*	Title for the shade plot
WINDOW = *scalar*	Window number for the graphs; default 3
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *
SPREADSHEET = *string token*	What results to put in spreadsheets (top%probes); default * i.e. none

Parameters

DATA = *variates* or *pointers*	Data values (i.e. log-ratios)
SLIDES = *factors*, *texts* or *variates*	Identifies the slides
PROBES = *factors*, *texts* or *variates*	Identifies the probes or genes
GMEANS = *matrices*	Saves the tabulation of the data by probe groups and target groups, as a two-way matrix
PGROUPS = *factors*	Saves the group membership for each probe (or gene)
SGROUPS = *factors*	Saves the group membership for each slide (or target)
PAMALGAMATIONS = *matrices*	Saves the probe (or gene) amalgamation data when

	METHOD=hier
SAMALGAMATIONS = *matrices*	Saves the slide (or target) amalgamation data when METHOD=hier

Description

MA2CLUSTER perform a two-way clustering of probes (which may be thought of as representing genes) and slides (or target) effects. The METHOD option specifies whether the clustering is hierarchical, or non-hierarchical using the k-means algorithm. A range of clustering criteria are available for each method (option DMETHOD, LMETHOD and CRITERION). The probes are grouped together so that the responses of each group are similar, with the groups as distinct as possible. For the hierarchical clustering, the allocation to groups is specified by using the PGTHRESHOLD and SGTHRESHOLD option to provide a threshold for the levels of similarity within a group when clustering the probes and slides, respectively. The dendrograms are then cut at these levels, generating an unknown number of groups. For the k-means algorithm, the number of groups must be specified using the PNGROUPS and SNGROUPS options. The group memberships can be saved by the PGROUPS and SGROUPS parameters. You can set option STANDARDIZE=centre to centre the log-ratios by probe and slide before the clustering.

The log-ratios are supplied by the DATA parameter. If these are in a single variate, the SLIDE parameter should supply a factor to index the slides, and the PROBES parameter should index the probes or genes. Alternatively you can supply a pointer containing a variate for each slide. The slides factor is then not required; if it is given it should just have one entry for each slide in the order of the variates in the pointer. The PROBES factor is that for a single slide, and all slides must have a common layout.

The PLOT option allows you to plot a dendrogram for the hierarchical cluster analyses, but for a large number of probes this is less useful as individual probes cannot be read. The responses of each probe across the targets/slides can also be plotted in a shade plot, but for large numbers of probes this is slow, in which case the mean response for each group can be plotted instead. A spreadsheet containing the grouped data can also be saved using the SPREADSHEET option.

With large numbers of probes, the limit of RAM can be quickly reached, so option PERCENT can be set so that only cluster probes with the largest mean absolute responses are clustered.

The DTITLE and STITLE options can supply titles for the dendrogram and shade plot, respectively, and the WINDOW option specifies the window to use (by default 3). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

Options: PRINT, PLOT, METHOD, DMETHOD, LMETHOD, CRITERION, PNGROUPS, SNGROUPS, PGTHRESHOLD, SGTHRESHOLD, PERCENT, COLOURS, DTITLE, STITLE, WINDOW, DEVICE, GRAPHICSFILE, SPREADSHEET.

Parameters: DATA, SLIDES, PROBES, GMEANS, PGROUPS, SGROUPS, PAMALGAMATIONS, SAMALGAMATIONS.

Action with RESTRICT

Any restrictions on the DATA variates are removed.

See also

Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAESTIMATE, MAHISTOGRAM, MAPCLUSTER, MAPLOT, MASCLUSTER, MASHADE, MAVOLCANO, MNORMALIZE.

Genstat Reference Manual 1 Summary section on: Microarray data.

MCNEMAR

Performs McNemar's test for the significance of changes (R.W. Payne & D.A. Murray).

Options

`PRINT` = *string tokens*	Controls printed output (`test`, `table`); default `test`
`METHOD` = *string token*	Type of test required (`twosided`, `greaterthan`, `lessthan`); default `twos`

Parameters

`Y1` = *factors* or *tables*	Factor containing the responses obtained before the treatment (with 1 indicating a positive response) or two-by-two table (classified by factors representing the two occasions of testing) summarizing the responses before and after treatment
`Y2` = *factors*	Factor containing the responses obtained after the treatment (need not be specified if `Y1` is a table)
`STATISTIC` = *scalars*	Saves the test statistic
`PROBABILITY` = *scalars*	Saves the probability value

Description

The McNemar test is useful for analysing studies where subjects are assessed before and after a treatment. The response on each occasion is assumed to be categorized by a factor with two levels. Usually level 1 represents a *negative* response, and level 2 a *positive* response. The test assesses the consistency of the responses on the two occasions. By default the test is assumed to be two-sided (that is, changes in the overall response from level 1 to level 2 or from level 2 to level 1 are equally of interest). However, you can set the `METHOD` option to `greaterthan` for a one-sided test of the null hypothesis that the number of level 2 responses is not increasing (i.e. that the overall response is not becoming more positive), or to `lessthan` for a test of the null hypothesis that the number of level 2 responses is not decreasing.

The data for the test can be supplied as two variates (one for each occasion) using the `Y1` and `Y2` parameters. Positive responses are represented by the value one, and other values are taken to indicate negative responses. (So the variates might be formed from logical tests, for example using the `.EQ.` or `.EQS.` operators.) Alternatively, you can set `Y1` to a two-by-two table classified by a factor representing the assessments before the treatment and another representing the assessments after the treatment.

In its original form, the test leads to a chi-square test (see the Method section). However, this may be inaccurate when there are small numbers of subjects. Consequently Genstat also provides an exact probability (based on the binomial distribution). The value of the statistic can be saved using the `STATISTIC` parameter, and the exact probability can be saved using the `PROBABILITY` parameter.

The `PRINT` option controls printed output with settings:

`test`	to print the test statistic and probabilities, and
`table`	to print the table of responses.

The default is `PRINT=test`.

Options: `PRINT`, `METHOD`.
Parameters: `Y1`, `Y2`, `STATISTIC`, `PROBABILITY`.

Method
The test is constructed by first forming a table giving the numbers of subjects giving each combination of responses over the two occasions. Suppose that the table contains the values A, B, C and D as below:

First occasion	Second occasion	
	negative	positive
negative	C	D
positive	A	B

The test statistic is a chi-square statistic for assessing the equality of A and D, which represent the changes from positive to negative, and negative to positive, respectively. Including the continuity correction of Yates (1934), leads to the calculation

```
Statistic = ((ABS(A - D) - 1)**2) / (A + D)
```

see Siegel (1956), pages 63-67. Under the null hypothesis, this has a chi-square distribution with one degree of freedom. The alternative exact probability calculation assumes that, under the null hypothesis, the numbers A and D are generated by a binomial distribution with A+D samples and probability 0.5.

Action with RESTRICT
If Y1 or Y2 are restricted the test is made on only the units not excluded by the restriction.

References
Siegel S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

Yates, F. (1934). Contingency tables involving small numbers and the chi-square test. *Supplement to the Journal of the Royal Statistical Society*, **1**, 217-235.

See also
Procedures: CATRENDTEST, QCOCHRAN, SMCNEMAR.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

MCOMPARISON

Performs pairwise multiple comparison tests within a table of means (D.M. Smith).

Options

PRINT = *string tokens*	Controls printed output (comparisons, critical, description, lines, letters, plot, mplot, pplot); default lett
METHOD = *string token*	Test to be performed (flsd, bonferroni, sidak); default flsd
DIRECTION = *string token*	How to sort means (ascending, descending); default asce
PROBABILITY = *scalar*	The required significance level; default 0.05
STUDENTIZE = *string token*	Whether to use the alternative LSD test where the Studentized Range statistic is used instead of Student's t (yes, no); default no

Parameters

MEANS = *tables* or *variates*	Means to be compared
SED = *symmetric matrix* or *scalar*	Standard errors of differences of the means
DF = *symmetric matrix* or *scalar*	Degrees of freedom for the standard errors of differences
VMEANS = *pointer* or *variate*	Saves the means in a variate, sorted as requested by the DIRECTION option
DIFFERENCES = *symmetric matrix*	Saves differences between the (sorted) means
LABELS = *text*	Saves labels for the (sorted) means
LETTERS = *text*	Saves letters indicating groups of means that do not differ significantly
SIGNIFICANCE = *symmetric matrix*	Indicators to show significant comparisons between (sorted) means
CIWIDTH = *symmetric matrix*	Saves the width of the confidence interval for the absolute differences between the (sorted) means
TERMNAME = *texts*	Name of the term, to use to annotate the graphs

Description

MCOMPARISON can be used perform all pairwise multiple comparison tests on tables of predicted means, that may be saved for example from PREDICT or VPREDICT. The methodology implemented in the procedure closely follows that described in Chapter 5 of Hsu (1996).

The MEANS parameter supplies a table containing the means to be compared. You also need to provide standard errors of differences and their degrees of freedom, using the SED and DF parameters, respectively. These can be in scalars, if they are identical for every comparison between two means. Otherwise they should be in symmetric matrices.

Printed output is controlled by the PRINT option, with settings:

comparisons	prints the differences between the pair of means, upper and lower confidence limits for the differences, t-statistics and an indication of whether or not they are significant;
critical	gives critical values for the t-statistic for situations where these do not vary amongst the comparisons (i.e. for the Scheffe, Bonferroni and Sidak methods, as well as the Fisher LSD methods provided all the comparisons have the same number of residual degrees of freedom);
description	provides a description including information such as the

	experiment-wise and compartment-wise error rates;
`lines`	gives the means, with lines joining those that do not differ significantly;
`letters`	gives the means, with identical letters (a, b etc.) alongside those that do not differ significantly;
`mplot`	does a mean-mean scatter plot (synonym `plot`);
`pplot`	displays the probabilities in a shade plot.

By default, `PRINT=letters`.

The means are usually sorted into ascending order, but you can set option `DIRECTION=descending` for descending order, or `DIRECTION=*` to leave them in their original order. Note, though, that the lines joining means with non-significant differences may then be broken.

If the standard errors for the differences between the means are unequal, the memberships of the groups defined by the lines or letters may be inconsistent. Suppose, for example, you have ordered means A, B and C. If the s.e.d. for A vs. C is large compared to those for A vs. B and B vs C, you might find that there is no significant difference between A and C, but there are significant differences between A and B, and between B and C. So treatments A and B and treatments B and C would be in different groups. However, treatments A and C (which are further apart) would be in the same group. This contradicts the idea behind multiple comparisons, where you expect that if two means are in the same group, then any mean between them should be in that group too. If `MCOMPARISON` finds inconsistencies like this, it gives a diagnostic and suppresses the printing of lines and letters (but not the other types of output).

The mean-mean scatter plot allows you to assess the confidence region for the difference between each pair of means visually. It has grid lines from both the x- and y-axis at the position of each mean, and a diagonal line at 45 degrees marking y=x. The confidence interval for each pair of means is plotted as a line at an angle of −45 degrees and centred on the intersection above the line y=x of the grid lines for the two means (so the y grid line is for the larger of the two means, and the x grid line is for the smaller mean). The difference between the means is significant if their confidence line does not intersect the line y=x. For more details, see Hsu (1996) pages 151-153.

The shade plot displays the probabilities in a symmetric matrix. The colour of each cell represents the probability for the difference between the means for the treatments in the corresponding row and column.

The type of test to be performed is specified by the `METHOD` option, with settings `FLSD` (Fisher's Unprotected Least Significant Difference), `Bonferroni` and `Sidak`. The `PROBABILITY` option allows the experiment-wise significance level for the intervals from the Bonferroni and Sidak tests to be changed from the default 0.05 (e.g. to 0.01). For the Fisher's test, it changes the pair-wise significance level. The `STUDENTIZE` option can specify that the Fisher's protected or unprotected LSD tests should use the Studentized Range statistic rather than Student's t (for further information see Hsu 1996, page 139).

The `VMEANS` parameter can save the means in a variate, sorted according to the `DIRECTION` option and omitting any that were non-estimable. The `LABELS` parameter can save a text containing labels to identify the means, and the `LETTERS` parameter can save a text with the letters identifying means that do not differ significantly. The `SIGNIFICANCE` parameter can save a symmetric matrix containing ones or zeros according to whether the various comparisons were significant or non-significant. The `DIFFERENCES` parameter can save a symmetric matrix containing the differences between the (sorted) means, and the `CIWIDTH` parameter can save a symmetric matrix containing the widths of the confidence intervals for the differences.

Options: `PRINT`, `METHOD`, `DIRECTION`, `PROBABILITY`, `STUDENTIZE`.
Parameter: `MEANS`, `SED`, `DF`, `VMEANS`, `DIFFERENCES`, `LABELS`, `LETTERS`, `SIGNIFICANCE`,

CIWIDTH, TERMNAME.

Method
The methodology implemented is based on that described in Hsu (1996).

Reference
Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also
Procedures: AMCOMPARISON, PPAIR, AUMCOMPARISON, VMCOMPARISON.

MCORANALYSIS

Does multiple correspondence analysis (A.I. Glaser).

Options

PRINT = *string tokens*	Printed output from the analysis (roots, rowscores, rowinertias, rowchisquare, rowmass, rowquality, colscores, colinertias, colchisquare, colmass, colquality); default * i.e. no output
ROWMETHOD = *string token*	Analysis method for rows i.e. units (indicator); default indi
COLMETHOD = *string token*	Analysis method for columns i.e. factors (adjusted, burt, indicator); default adju
NROOTS = *scalar*	Number of latent roots for printed output; default * requests them all to be printed
%METHOD = *string token*	How to represent proportions or %s in quality statistics (permills, percentages, proportions); default prop
NDIMENSIONS = *scalar*	Number of dimensions for which quality statistics are required; default 2
TOLERANCE = *scalar*	Tolerance criteria for zero eigenvalues; default 10^{-6}

Parameters

DATA = *pointers*	Data to be analysed
ROOTS = *diagonal matrices*	Saves the squared singular values from each analysis
ROWSCORES = *matrices*	Saves the scores for the rows of the data
COLSCORES = *matrices*	Saves the scores for the columns of the data
ROWINERTIAS = *matrices*	Saves the total inertias for the rows of the data
COLINERTIAS = *matrices*	Saves the total inertias for the columns of the data
ROWQUALITY = *matrices*	Saves the quality statistics for rows of the data
COLQUALITY = *matrices*	Saves the quality statistics for columns of the data
SUBINERTIAS = *matrices*	Saves the inertias of the subtables of the Burt matrices
FREQUENCY = *variates*	Frequencies for elements of DATA
SAVE = *pointers*	Saves details of the analysis for use by CABIPLOT

Description

Ordinary correspondence analysis is an ordination technique used to analyse relationships between two categorical variables (see procedure CORANALYSIS). Ordination techniques aim to represent the relationships approximately, in a reduced number of dimensions, to make them easier to study e.g. with graphs. Multiple correspondence analysis provides a similar analysis for more than two variables.

The data consist of a list of factors, which are supplied in a pointer by the DATA parameter. By default, each unit of the factors is assumed to represent a single observation. However, with large data sets, you may want to use the FREQUENCY parameter to supply a variate defining frequencies (or numbers of replications) for each unit. MCORANALYSIS uses the data to form an *indicator* matrix *D*, with a row for each unit and a columns for each level of every factor. Each row of the matrix has the value one in the columns corresponding to the levels of the factors that occurred in that data unit and zero elsewhere. (This is equivalent to the design matrix that is used in analysis of variance or regression.) The factors must not contain any missing values.

The relationships between the rows are assessed by doing an ordinary correspondence analysis on the indicator matrix. This analysis also provides information on the relationships between the

columns (i.e. the factor levels). However, an alternative method for the columns does the correspondence analysis on the Burt matrix $D'D$. A refinement of the use of the Burt matrix discards eigenvalues below a threshold $1/Q$, where Q is the number of DATA factors. This adjusts for the inflation of the eigenvalues that arises from the within-factor diagonal blocks of the Burt matrix; see Greenacre (2007) Chapter 19 for more details. The difference between the results obtained using the indicator and Burt matrices is that the singular values obtained from the Burt matrix will be the squares of those obtained from the indicator matrix. The adjusted method is the default method for the columns, but the other two methods can be requested by using the COLMETHOD option. With very large data sets it may be impractical to do the correspondence analysis on the indicator matrix for rows. So MCORANALYSIS allows this to be suppressed by setting option ROWMETHOD=*.

Printed output is controlled by the PRINT option with settings:

roots	to print the roots (together with the roots expressed as percentages and cumulative percentages),
rowscores	to print the scores for the rows of the indicator matrix,
rowinertias	to print the inertias for the rows of the indicator matrix,
rowmass	to print the row masses,
rowchisquare	to print the row chisquare distances,
rowquality	to print the quality statistics for the rows,
colscores	to print the scores for the columns of the indicator or Burt matrix (as selected by the COLMETHOD option),
colinertias	to print the inertias for the columns,
colmass	to print the column masses,
colchisquare	to print the column chisquare distances,
colquality	to print the quality statistics for the columns, and
subinertias	to print the inertias of the subtables of the Burt matrix.

The NROOTS option controls the printed output of roots, scores and inertias. By default, results are printed for all the roots greater than the limit defined by the TOLERANCE option. However, you can set the NROOTS option to specify a lesser number.

The quality settings produce tables with the following columns:

- the mass of the row (or column), in proportion to the total mass;
- the "quality" of the representation i.e. how much of the inertia of a row (or column) is represented by the dimensions shown;
- the proportion of the total inertia of the row (or column) compared to the total inertia for all rows (or columns);
- principal coordinates of the rows (or columns) in the specified dimension;
- the amount of inertia for each row (or column) in the specified dimension relative to the total amount of inertia given by the value of the quality statistic – hence the sum of a specific row (or column) across the dimensions shown will be equal to the value given by the quality statistic;
- the proportion of inertia explained by a row (or column) in a dimension, compared to the total inertia in that dimension.

The representation of the columns of proportions is controlled by the %METHOD option; these can be printed either as proportions (default), percentages or as permills i.e. tenths of a percent. The NDIMENSIONS option specifies the number of dimensions for which to print quality statistics; default 2.

Results from the analysis can be saved using the parameters ROOTS, ROWSCORES, COLSCORES, ROWINERTIAS, COLINERTIAS, ROWQUALITY and COLQUALITY. The structures specified for these parameters need not be declared in advance. The SAVE parameter can save full details of the analysis for use by the CABIPLOT procedure.

Options: PRINT, ROWMETHOD, COLMETHOD, NROOTS, %METHOD, NDIMENSIONS, TOLERANCE.
Parameters: DATA, ROOTS, ROWSCORES, COLSCORES, ROWINERTIAS, COLINERTIAS, ROWQUALITY, COLQUALITY, SUBINERTIAS, FREQUENCY, SAVE.

Method

MCORANALYSIS first applies correspondence analysis to the indicator matrix. This is essentially the design matrix D for an analysis of variance or regression, fitting a model with just the main effects of the factors, and can be obtained from the TERMS directive as follows:

```
CALC       nv = NVAL(DATA[1])
MODEL      !(1...#nv)
TERMS      [FULL=yes; DESIGN=D] DATA[]
DUPLICATE [REDEFINE=yes] D$[*; -1]; NEWSTRUCTURE=D
```

(The DUPLICATE statement removes the column for the constant, which is not required.) The Burt matrix D′D can be calculated by

```
CALCULATE Burt = T(D) *+ D
```

When METHOD=adjusted, all the eigenvalues (squared singular values) less than or equal to $1/Q$ are set to zero, where Q is the number of variables in the data. To take into account the inflated inertia, each non-zero eigenvalue λ is then multiplied by

$$(Q / (Q - 1) * (\lambda - 1/Q))^2$$

The percentages are calculated by dividing the adjusted eigenvalues by the sum of the pre-adjusted eigenvalues, so they may not always sum to 100%.

References

Greenacre, M.J. (1984). *Theory and Applications of Correspondence Analysis*. Academic Press, London.

Greenacre, M. (2007). *Correspondence Analysis in Practice, second edition*. Chapman & Hall, London.

See also

Procedures: CABIPLOT, CORANALYSIS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

MCROSSPECTRUM

Performs a spectral analysis of a multiple time series (G. Tunnicliffe Wilson & R.P. Littlejohn).

Options

`PRINT` = *string token*	Controls printed output (`description`); default `desc`
`PLOT` = *string tokens*	Variables for which to plot the analysis (`explanatory`, `response`); default `expl`, `resp`
`CORRECT` = *string token*	Whether to mean or trend correct the series (`mean`, `linear`, `quadratic`, `none`); default `mean`
`BANDWIDTH` = *scalar*	Bandwidth for smoothing, must be between 0 and 0.5; if unset, a default is calculated automatically
`MAXLAG` = *scalar*	Maximum lag for the time domain outputs; if unset, a default is calculated automatically
`PROBABILITY` = *scalar*	Probability value for confidence limits; default 0.95
`TAPER` = *scalar*	The proportion of data to be tapered using a cosine bell window; default 0
`YLOG` = *string token*	Whether to plot the univariate spectra with a $\log_{10}$-transformed y-axis (`yes`, `no`); default `no`

Parameters

`Y` = *variates*	Response time series
`X` = *variates or pointers*	Explanatory time series
`ALIGN` = *variate*	Shifts to apply to the explanatory series; default none
`SPECTRUM` = *pointers*	Saves autospectra, co-spectra and quad-spectra
`FREQUENCY` = *variate*	Saves the frequency values at which the spectra are calculated
`VARSPECTRUM` = *pointers*	Saves information about the variation of the spectrum: coefficient of variation, degrees of freedom, and lower and upper multiplicative limits for the univariate spectra
`MULTICOHERENCYSQUARED` = *pointers*	Saves estimates, significance limits, lower and upper confidence limits for the squared multiple coherency between the response and explanatory series
`PARTIALCOHERENCYSQUARED` = *pointers*	Saves estimates, significance limits, lower and upper confidence limits for the squared partial coherency of the response series with each explanatory series
`GAIN` = *pointers*	Saves estimates, lower and upper limits for the estimated gain of response series from each of the explanatory series
`PHASE` = *pointers*	Saves estimates, lower and upper limits for the estimated phase of response series from each of the explanatory series
`NOISESPECTRUM` = *variates*	Saves the estimated spectrum of the noise process
`IMPULSERESPONSE` = *pointers*	Saves the impulse response from −maxlag to +maxlag: estimates and significance limit
`LAGS` = *variates*	Saves the lags for the impulse response
`ACFNOISE` = *variates*	Saves the ACF of the noise process

Description

MCROSSPECTRUM performs a spectral analysis of a multiple time series. The response series is specified by the Y parameter. The explanatory series are specified by the X parameter; the setting can be a single variate if there is only one explanatory series, or a pointer of variates if there are several. All the series should be the same length, *n* say, and this must be greater than 10. There must also be no missing values and no restrictions. The ALIGN parameter can supply a variate, with a value for each explanatory variate, which specifies a shift *s* so that $X(t-s)$ is more closely aligned with $Y(t)$. These are used to improve the accuracy of the analysis but the results still relate to the original (unshifted) series.

The band-width of the smooth is specified by the BANDWIDTH option. If this is unset, a default is calculated automatically. If BANDWIDTH is less than $1/n$, only the sample spectra are returned with no smoothing. The MAXLAG option defines the maximum lag for the time domain outputs. If this is not set, a default is calculated automatically. Also, if the supplied value of MAXLAG is too great in relation to the series length or the bandwidth used, then it is adjusted as necessary. The TAPER option specifies the tapering proportion (default 0), and the PROBABILITY option defines the size of confidence limits and acceptance region for coherencies (default 0.95).

The CORRECT option has settings mean, linear, quadratic and none to control whether a mean, linear or quadratic trend correction is applied to all the series. The default is mean correction.

Printed output can be suppressed by setting the option PRINT=*; by default, PRINT=description, which summarizes the variables used and the option settings. The plots that are produced are controlled by the PLOT option, with settings:

explanatory	produces a graphics page for each explanatory variable containing the spectrum, its partial coherency squared with the response variable, phase, gain and impulse response function, and
response	produces a graphics page with the response and noise spectra, the multiple coherency squared, and the autocorrelation function for the noise process. Where given, green lines denote null significance limits.

By default, both pages are produced.

The YLOG option specified the transformation to be made to the y-axes of the autospectra plots. By default, the plot is on the natural, untransformed scale. Alternatively, you can set YLOG=yes, to plot on the scale of logarithm, base 10.

The SPECTRUM parameter saves a pointer, with 2 suffixes, storing variates of spectra: "diagonals" (e.g. [1][1], [2][2] etc.) store autospectra, "super-diagonals" ([1][2] etc.) store co-spectra, and "sub-diagonals" ([2][1] etc.) store quad-spectra. The frequency values at which the spectra are calculated can be saved, in a variate, by the FREQUENCY parameter. The frequency range is from 0 to 0.5 cycles per sampling interval of the series. This range is divided into a round number of intervals with approximately 10 divisions covering one bandwidth.

The VARSPECTRUM parameter saves a pointer with information about the variation of the spectrum. The first element of the pointer is a variate storing the coefficient of variation of the spectrum. Similarly the second element stores the corresponding degrees of freedom, and the third and fourth elements store lower and upper multiplicative limits for the univariate spectra.

The MULTICOHERENCYSQUARED parameter saves a pointer containing the squared multiple coherency between the response and explanatory series. The first element of the pointer is a variate storing the estimates, the second element stores the significance limits, and the third and fourth elements store the lower and upper confidence limits.

The PARTIALCOHERENCYSQUARED, GAIN, PHASE and IMPULSERESPONSE parameters each save their results in variates within a pointer with two suffixes. The first suffix changes according to the type of result, while the second suffix has an element 1...*m* for each of the *m*

explanatory variates. The PARTIALCOHERENCYSQUARED parameter saves results for the squared partial coherency of response series with the explanatory series; its first suffix has elements 1-4 to store the estimates, the significance limits, and the lower and upper confidence limits. The GAIN and PHASE parameters save the estimated gain and phase of response series from each of the explanatory series; their first suffixes have elements 1...3, storing the estimates, the lower and the upper limits. The IMPULSERESPONSE parameter saves the impulse response, from −maxlag to +maxlag; its first suffix has elements 1 and 2, storing the estimates and the significance limits.

The NOISESPECTRUM and ACFNOISE parameters store the estimated spectrum and ACF of the noise process, in a variate. Finally, the LAGS parameter stores the lags for the impulse response, again in a variate.

Options: PRINT, PLOT, CORRECT, BANDWIDTH, MAXLAG, PROBABILITY, TAPER, YLOG.
Parameters: Y, X, ALIGN, SPECTRUM, FREQUENCY, VARSPECTRUM, MULTICOHERENCYSQUARED, PARTIALCOHERENCYSQUARED, GAIN, PHASE, NOISESPECTRUM, IMPULSERESPONSE, LAGS, ACFNOISE.

Action with RESTRICT
There must not be any restrictions.

See also
Directive: FOURIER.
Procedures: DFOURIER, PERIODTEST, PREWHITEN, REPPERIODOGRAM, SMOOTHSPECTRUM.
Genstat Reference Manual 1 Summary section on: Time series.

MC1PSTATIONARY

Gives the stationary probabilities for a 1st-order Markov chain (R.P. Littlejohn).

Option

PRINT = *string token*	What to print (transitions, pstationary); default psta

Parameters

DATA = *matrices* or *factors*	Specifies the Markov chain as a factor, or matrix of transitions
STATES = *texts*	Labels for the states
PSTATIONARY = *variates*	Saves the stationary probabilities
TRANSITIONS = *matrices*	Saves the transition matrices

Description

MC1PSTATIONARY prints and/or saves the stationary probabilities for a first-order Markov chain. The data are input using the DATA parameter, as either a matrix of transition counts or a factor of states from which the transition matrix is calculated. The probabilities and transition matrix can be saved using the TRANSITIONS and PSTATIONARY parameters, reaspectively.

Option: PRINT.
Parameters: DATA, STATES, PSTATIONARY, TRANSITIONS.

Method

The procedure uses LSVECTORS to obtain the required eigenvector.

Action with RESTRICT

If the DATA parameter is set to a list of factors, these must not be restricted.

See also

Genstat Reference Manual 1 Summary section on: Time series.

MEDIANTETRAD

Gives robust identification of multiple outliers in 2-way tables (J.K.M. Brown).

Options

PRINT = *string tokens*	Printed output required (graph, table); default grap, tabl
GRAPHICS = *string tokens*	Type of graph required (highresolution, lineprinter); default high
SORT = *string tokens*	Sorting of printed output, in order of absolute value of median tetrad (ascending, descending, none); default none

Parameters

TABLE = *tables*	Specifies the two-way table of data
ROWS = *factors*	Saves the factor classifying the table rows
COLUMNS = *factors*	Saves the factor classifying the table columns
DATA = *variates*	Saves the data values in the body of the table
MEDIANTETRADS = *variates*	Saves median tetrads for each cell in the table
RANKS = *variates*	Saves ranks of absolute values of median tetrads
HALFNORMALSCORES = *variates*	Saves half-Normal scores of absolute values of median tetrads
TESTOUTLIERS = *scalars*	Specifies the number of cells, with the highest absolute median tetrads, to be set to their predicted values before re-running the analysis

Description

In a table of data cross-classified by two factors, some cells may be outliers, in that they contain values substantially higher or lower than those expected from the means of the relevant rows and columns. Median tetrad analysis is a robust, single-step method of identifying several outliers in a two-way table (Bradu & Hawkins 1982).

A tetrad is calculated from four cells which form a square in the body of the table. For instance, if the cell in row i and column j has a value c_{ij}, the tetrad involving that cell and the cell in row p and column q is defined as

$$t_{ij;\,pq} = c_{ij} - c_{iq} - c_{pj} + c_{pq}$$

In a clean tetrad, none of the values c_{iq}, c_{pj} or c_{pq} are themselves outliers, so the tetrad is an estimate of the amount by which c_{ij} deviates from its expected value. In a contaminated tetrad, one of more of c_{iq}, c_{pj} or c_{pq} are outliers, so a contaminated tetrad is not a reliable estimate of the deviation of c_{ij} from its expectation.

MEDIANTETRAD calculates the median of all the tetrads involving each cell of the table (such that $i \neq p$ and $j \neq q$, so the four cells in the tetrad form a square). These median tetrads are robust estimates of the deviations for each cell and therefore indicate which cells may contain outliers. The method is robust because the median will be a clean tetrad (and therefore a reliable estimate of the deviation) so long as fewer than half the tetrads involving that cell are contaminated. Furthermore, the robustness of the method allows several outliers to be detected reliably in a single step; other methods of detecting outliers may detect only a single outlier, or may require several steps, one for each outlier.

The options of MEDIANTETRAD control the output. PRINT has two settings. The graph setting produces a plot of half-Normal scores of the median tetrads against the absolute values of the median tetrads. In the half-Normal plot, inliers (values for cells which are not outliers, with low deviations) fall on a straight line passing through the origin, while outliers (with high deviations) fall at the upper end of this line and below the level of the line. A regression line, passing

through the origin, of half-Normal scores against absolute values of median tetrads, is also plotted. The setting `table` prints the factors which classify the table, the data in the body of the table, the median tetrads, the ranks of the absolute values of the median tetrads and the half-Normal scores. The `GRAPHICS` option controls graphical output, as a high-resolution plot (the default setting) or as a line-printer plot. The `SORT` option controls whether the output provided by setting `PRINT=table` is sorted in ascending order (most extreme median tetrad last), descending order, or not at all.

The `TABLE` parameter specifies a table, classified by two factors, in which outliers are to be identified. The table may contain missing values, in which case the corresponding median tetrad is returned as a missing value. The `TABLE` parameter must be set, while the other parameters are optional. The next six parameters save output. `ROWS` and `COLUMNS` save the factors which classify the table, `DATA` saves the numerical body of the table, and `MEDIANTETRADS`, `RANKS` and `HALFNORMALSCORES` save the median tetrads, their ranks and half-Normal scores respectively.

When a table has few rows (or, equivalently, few columns), a large outlier in the cell in row *i* and column *j* may cause other cells in column *j* to appear to be moderately outlying. This is bound to be a problem if the table has only two or three rows, in which case 100% or at least 50%, respectively, of tetrads involving cells in column *j* will be contaminated, so the median tetrads of those cells will be contaminated. The presence of missing values may also cause this problem to occur in larger tables, by reducing the proportion of clean tetrads. The parameter `TESTOUTLIERS` can be used to examine the influence of suspected outliers on the deviations of other cells. When `TESTOUTLIERS` is set to a positive integer (*m*), the analysis is run twice. In the first run, the data used is that supplied in `TABLE`. In the second run, the cells with the highest *m* absolute median tetrads are set to values estimated from the remainder of the data (i.e. those not suspected to be outliers). If these *m* values are indeed the only notable outliers, all the data will now be inliers, so the half-Normal plot of the median tetrads will be a close fit to a straight line passing through the origin. Note that, if `TESTOUTLIERS` is set, the output saved in the variates set by the `DATA`, `MEDIANTETRADS`, `RANKS` and `HALFNORMALSCORES` parameters will be from the second analysis, that of the modified table. If the option `GRAPHICS=highresolution` is set in combination with a non-zero value of `TESTOUTLIERS`, you may need to set the option "Multiple Windows" in the Windows version of Genstat Graphics in order to see the two graphs, before and after adjustment of the suspected outliers.

Options: `PRINT`, `GRAPHICS`, `SORT`.

Parameters: `TABLE`, `ROWS`, `COLUMNS`, `DATA`, `MEDIANTETRADS`, `RANKS`, `HALFNORMALSCORES`, `TESTOUTLIERS`.

Method

All proper tetrads are calculated for each cell and their median is calculated. The median tetrad for a cell with a missing value is set to a missing value. The absolute values of the median tetrads are then ranked and their half-Normal scores calculated, as described in the Procedure Library Manual for `APLOT`. If `TESTOUTLIERS` is set to an integer $m>0$, the cells with the highest *m* outliers are set to missing values, an analysis of variance (anova) is carried out with treatmentstructure `ROWS + COLUMNS` (i.e. no interaction term is fitted), then the *m* cells with suspected outliers are given the appropriate fitted value saved from that anova.

References

Bradu, D. & Hawkins, D.M. (1982). Location of multiple outliers in two-way tables, using tetrads. *Technometrics*, **24**, 103-108.

See also
Directive: TABULATE.
Procedure: DRESIDUALS, RCHECK.

META

Combines estimates from individual trials (R.W. Payne & S. Senn).

Options

PRINT = *string tokens*	Controls output (estimates, overalltest, heterogeneity, confidenceplot, radialplot, monitoring); default esti, over, hete, conf
SELECTION = *string tokens*	Which combined estimates to include in the output (fixed, random); default fixe, rand
RMETHOD = *string token*	How to form the random estimate (maxlikelihood, maxremllikelihood, moments, reml); default reml
XLABEL = *text*	Label for the x-axis of the confidence plot; default 'treatment effect'
SMETHOD = *string token*	How to set the sizes of symbols on the confidence plot (equal, inversese); default inve
CIPROBABILITY = *scalar*	Probability level to use for the confidence intervals; default 0.95
CIMETHOD = *string token*	Method to use for calculating the confidence interval for random estimates formed by maximum likelihood or REML (approximate, profile); default prof
PRMETHOD = *string token*	Type of test to use for the overall probability values (greaterthan, lessthan, twosided); default grea
MAXCYCLE = *scalar*	Maximum number of iterations to use with RMETHOD settings maxlikelihood and maxremllikelihood; default 100
TOLERANCE = *scalar*	Convergence criterion to use with RMETHOD settings maxlikelihood and maxremllikelihood; default 10^{-6}

Parameters

ESTIMATES = *variates*	Supplies the estimates to combine
SEESTIMATES = *variates*	Specifies the standard errors of the estimates
LABELS = *texts*	Labels to use for each variate of ESTIMATES in the output
FIXEDESTIMATE = *scalars*	Saves the combined estimate for each variate of ESTIMATES, treating them as fixed effects
SEFIXEDESTIMATE = *scalars*	Saves the standard error of the combined estimate for each variate of ESTIMATES, treating them as fixed effects
PRFIXEDESTIMATE = *scalars*	Saves the probability of the combined estimate for each variate of ESTIMATES, treating them as fixed effects
RANDOMESTIMATE = *scalars*	Saves the combined estimate for each variate of ESTIMATES, treating them as random effects
SERANDOMESTIMATE = *scalars*	Saves the standard error of the combined estimate for each variate of ESTIMATES, treating them as random effects
PRRANDOMESTIMATE = *scalars*	Saves the probability of the combined estimate for each variate of ESTIMATES, treating them as random effects
QSTATISTIC = *scalars*	Saves the statistic Q for the test of heterogeneity across trials
QDF = *scalars*	Saves the degrees of freedom of the statistic Q

RVARIANCE = *scalars*	Saves the random effect variance
LOWER = *variates*	Saves lower values of the confidence interval
UPPER = *variates*	Saves upper values of the confidence interval

Description

META produces a combined estimate of a parameter that has been estimated in several separate trials, using the methods described in Chapter 4 of Whitehead (2002).

The estimates to be combined in the meta analyses must be supplied, in a variate, using the ESTIMATES parameter. Their standard errors must be supplied similarly, using the SEESTIMATES parameter. The LABELS parameter can supply a text with a label to be used for each estimate in the output; if this is not supplied, the default is to use the integers 1, 2 and so on.

Printed output is controlled by the PRINT option, with settings:

estimates	table with the estimates from the individual trials and the combined estimates, with standard errors and confidence intervals;
overalltest	overall tests using the combined estimates;
heterogeneity	test for heterogeneity of the estimates across trials (Whitehead 2002, Section 4.2.3);
confidenceplot	plot of the individual and combined estimates, and their confidence intervals;
radialplot	plot of the standardized estimates against their precision i.e. the reciprocal of the standard error (also known as a Galbraith plot; see Whitehead 2002 Section 7.3.2);
monitoring	monitoring information from the estimation with RMETHOD settings maxlikelihood, maxremllikelihood and reml (see below).

By default PRINT=esti,over,hete,conf.

The SELECTION option controls which combined estimates are presented in the output:

fixed	presents combined estimates formed assuming that the ESTIMATES are fixed (see Whitehead 2002, Section 4.2);
random	presents combined estimates formed assuming that the ESTIMATES are random.

By default SELECTION=fixe,rand.

The method to use to form the combined estimates formed assuming that the ESTIMATES are random, is specified by the RMETHOD option:

maxlikelihood	estimates the variance component of the random effects using maximum likelihood (Hardy & Thompson 1996, also see Whitehead 2002, Section 4.3.8);
maxremllikelihood	estimates the variance component of the random effects by maximizing the REML likelihood (Whitehead 2002, Section 4.3.8);
moments	estimates the variance component of the random effects using the method of moments (DerSimonian & Laird 1986, also see Whitehead 2002, Section 4.3.3);
reml	estimates the variance component of the random effects using the REML directive (Whitehead 2002, Section 4.3.8).

The maxremllikelihood setting is based on the same criterion as the reml setting, but it programs the maximization explicitly, in a FOR loop. It thus provides an alternative to use if the REML directive experiences convergence problems. By default RMETHOD=reml.

The CIMETHOD option specifies how to calculate the confidence interval for a random estimate

formed by maximum likelihood or REML. The default is to use profile likelihood (c.f. Hardy & Thompson 1996), but you can set CIMETHOD=approximate to use a Normal approximation instead.

The XLABEL option can supply a label for the x-axis of the confidence plots; the default is 'treatment effect'. By default the sizes of the symbols used to plot the estimates on the confidence plots are inversely proportional to their standard errors, but you can set option SMETHOD=equal to use equal sizes. The CIPROBABILITY option specifies the probability level to use for the confidence intervals; (default 0.95 i.e. 95%). The PRMETHOD option specifies the type of test to use for the overall probability values: greaterthan, lessthan or twosided; the default is greaterthan.

The MAXCYCLE option specifies the maximum number of iterations to use with RMETHOD settings maxlikelihood and maxremllikelihood (default 100). The TOLERANCE option specifies the convergence criterion (default 10^{-6}).

The combined estimate formed assuming that the ESTIMATES are fixed can be saved, in a scalar, using the FIXEDESTIMATE parameter. Its standard error and probability can be saved, each in a scalar, using SEFIXEDESTIMATE and PRFIXEDESTIMATE parameters. Similarly, the combined estimate formed assuming that the ESTIMATES are random can be saved using the RANDOMESTIMATE parameter, and the SERANDOMESTIMATE and PRRANDOMESTIMATE parameters can save its standard error and probability. The QSTATISTIC and QDF parameters can save the statistic *Q* for the test of heterogeneity across trials and its number of degrees of freedom, again in scalars. The RVARIANCE parameter can save a scalar containing the random effect variance. Finally, the LOWER and UPPER parameters can save variates containing the lower and upper values of the confidence interval.

Options: PRINT, SELECTION, RMETHOD, XLABEL, CIPROBABILITY, CIMETHOD, PRMETHOD, MAXCYCLE, TOLERANCE.

Parameters: ESTIMATES, SEESTIMATES, LABELS, FIXEDESTIMATE, SEFIXEDESTIMATE, PRFIXEDESTIMATE, RANDOMESTIMATE, SERANDOMESTIMATE, PRRANDOMESTIMATE, QSTATISTIC, QDF, RVARIANCE, LOWER, UPPER.

Method

META uses the algorithms described in Chapter 4 of Whitehead (2002).

Action with RESTRICT

ESTIMATES, SEESTIMATES or LABELS can be restricted to form combined estimates using only a subset of those in ESTIMATES.

References

DerSimonian, R. & Laird, N. (1986). Meta-analysis in clinical trials. *Controlled Clinical Trials*, **7**, 177-188.

Hardy, R.J. & Thompson, S.G. (1996). A likelihood approach to meta-analysis with random effects. *Statistics in Medicine*, **15**, 619-629.

Whitehead, A. (2002). *Meta-Analysis of Controlled Clinical Trials*. Wiley, Chichester.

See also

Directives: REML, VRESIDUAL.

Procedures: VRMETAMODEL, VMETA.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

MICHAELISMENTEN

Fits the Michaelis-Menten equation for substrate concentration versus time data (M.C. Hannah).

Options

PRINT = *string tokens*	What to print (model, deviance, summary, estimates, correlations, fittedvalues, monitoring); default mode, summ, esti
PLOT = *string tokens*	What to plot (concentration, rate); default conc
WINDOW = *scalar*	Window in which to plot the graphs; default 1
TITLE = *text*	Title for the graphs; default 'Michaelis-Menten process'
TTIMES = *text*	Title for the times axis; if this is unset, the identifier of the TIMES variate is used
TCONCENTRATIONS = *text*	Title for the concentrations axis; if this is unset, the identifier of the CONCENTRATIONS variate is used if available, otherwise 'Concentration'
TRATES = *text*	Title for the rates axis; if this is unset, the identifier of the RATES variate is used if available, otherwise 'Rate'
WEIGHTS = *variate*	Weights for the observations, to use in the fit, if required; default * i.e. all observations with weight one

Parameters

TIMES = *variates*	Times at which substrate concentration data were measured
CONCENTRATIONS = *variates*	Substrate concentration data
STEPLENGTHS = *variates*	Variate with four values defining initial step lengths for the parameters S_0, V_{max}, K_m and K_1 (in that order)
INITIAL = *variates*	Variate containing initial values for the parameters, similarly to STEPLENGTHS
RESIDUALS = *variates*	Saves the residuals from each fit
FITTEDVALUES = *variates*	Saves the fitted concentration values
ESTIMATES = *variates*	Saves the parameter estimates
SE = *variates*	Saves the standard errors of the estimates
VCOVARIANCE = *symmetric matrix*	Saves the variance-covariance matrix of the estimates
OBSRATES = *variates*	Saves reaction rates, calculated from the observed concentrations
FITRATE = *variates*	Saves fitted reaction rates

Description

The Michaelis-Menten equation, for biochemical reaction rate v, versus substrate concentration S

$$v(t) = dS(t) / dt = V_{max} S(t) / (K_m + S(t))$$

can be fitted in Genstat using

```
FITCURVE [CURVE=ldl; CONSTANT=omit]
```

with v as the response variate, and $1/S$ as the explanatory variate. However, in practice, data are available only for substrate concentration S at time t, and not for the reaction rate v. Instead of attempting to derive rate data, it is better statistically to fit $S(t)$ to the directly observed concentration data. The solution to the above differential equation, $S(t)$, has a characteristic hockey-stick shape where the response decreases linearly initially, and then curves to become

horizontal as it approaches the x-axis. However, no closed form expression for $S(t)$ exists. The procedure thus uses Golicnik's (2010) method to fit the model.

So, the procedure fits the curve $S(t)$ to observed concentration versus time data, obtaining parameter estimates for V_{max} and K_m. It can also estimate the initial concentration S_0, and an additive constant K_1 representing the concentration of non-reactive substrate (i.e. a lower asymptote). This generalized Michaelis-Menten curve is given by

$$v(t) = dS(t) / dt = V_{max} (S(t) - K_1) / (K_m + S(t) - K_1)$$

The substrate concentration data and the corresponding time values must be supplied, in variates, using the CONCENTRATIONS and TIMES parameters. Weights can be supplied using the WEIGHTS option.

You can supply initial values for the parameters, in a variate, using the INITIAL parameter. The variate should have four values, corresponding to the parameters S_0, V_{max}, K_m and K_1 (in that order). If INITIAL is unset, or if any of the values in the variate is missing, the procedure finds its own starting values for those not supplied. The STEPLENGTHS parameter can supply step lengths, again in a variate. You can fix a parameters at a specific value by specifying that value as the initial value, and defining a step length of zero. When doing this, it is usually simplest to fill the positions of the other, non-fixed, parameters with missing values, in both the INITIAL and STEPLENGTHS variates.

Printed output is controlled by the PRINT option. The settings all operate as in the FITNONLINEAR directive (which is used to fit the model). The default is to print a description of the model, the analysis summary and the estimated parameters.

The PLOT option controls the graphs that are plotted, with settings

concentration	to plot the curve fitted to the concentrations, and
rate	to plot the estimated reaction rates against the concentrations, and against time.

By default, PLOT=concentration.

The WINDOW option specifies the window to use for the graphs (default 1). The TITLE option can specify an overall title, and the TTIMES, TCONCENTRATIONS and TRATES options can specify titles for the axes for times, concentrations and rates, respectively.

You can save the fitted concentrations using the FITTEDVALUES parameter, and the residuals from the fit using the RESIDUALS parameter. The parameter estimates, their standard errors and variance-covariance matrix can be saved using the ESTIMATES, SE and VCOVARIANCE parameters. You can also save "observed" reaction rates (calculated from the observed concentrations) with the OBSRATES parameter, and fitted reaction rated with the FITRATES parameter.

You can use the post-regression directives, RCHECK, RKEEP etc., in the usual way to display or save additional output. You can also use an associated procedure, MMPREDICT, to predict $S(t)$ and $v(t)$ for a new time vector, given the parameter values estimated by MICHAELISMENTEN.

Options: PRINT, PLOT, WINDOW, TITLE, TTIMES, TCONCENTRATIONS, TRATES, WEIGHTS.
Parameters: TIMES, CONCENTRATIONS, STEPLENGTHS, INITIAL, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, VCOVARIANCE, OBSRATES, FITRATES.

Method

The procedure uses Golicnik's (2010) method to fit the model.

Action with RESTRICT

The data variates must not be restricted.

Reference

Golicnik, M. 2010. Explicit reformulations of time-dependent solution for a Michaelis-Menten

enzyme reaction model. *Analytical Biochemistry*, **406**, 94-96.

See also
Directives: FITCURVE, FITNONLINEAR.
Procedure: MMPREDICT.
Genstat Reference Manual 1 Summary section on: Regression analysis.

MINFIELDWIDTH

Calculates minimum field widths for printing data structures (R.W. Payne).

Option

IPRINT = *string tokens*	What identifier and/or text to print for the structure (identifier, extra); default is to take the IPRINT setting of each STRUCTURE

Parameters

STRUCTURE = *identifiers*	Data structures to be printed
FIELDWIDTH = *scalars*	Saves the minimum field widths
DECIMALS = *scalars*	Number of decimal places to be used for numerical data structures; if unset, a default is obtained using the DECIMALS procedure
SKIP = *scalars*	Number of spaces to leave before each value of the structure; default 1
FREPRESENTATION = *string tokens*	How to represent factor values (labels, levels, ordinals); default is to use labels if available, otherwise levels

Description

MINFIELDWIDTH can be used to calculate the minimum field width that would be required to print a data structure in an even column down the page using the PRINT directive. The data structures are specified by the STRUCTURE parameter, and can be any of those supported by Genstat. The calculated field width is saved, in a scalar, by the FIELDWIDTH parameter.

The IPRINT option indicates how the values of each STRUCTURE are to be labelled, so that the field widths will be wide enough for the column headings as well as the data values. The identifier setting uses the identifier of the STRUCTURE, while the extra setting used the information that can be specified by the EXTRA parameter when data structures are defined by directives like VARIATE, FACTOR and TEXT. You can set IPRINT=* to indicate that the values are not to be labelled by either of these. Alternatively, if IPRINT is not specified, the default is taken from the IPRINT attribute of the STRUCTURE (which can be set by the IPRINT option of VARIATE, FACTOR, TEXT etc). This is the same default that is used by PRINT if its own IPRINT option is not specified.

With numerical structures, like variates or matrices, the DECIMALS parameter specifies the number of decimal places that are to be used. If you set DECIMALS to a scalar containing a missing value, the DECIMALS procedure is used by MINFIELDWIDTH to determine a default number of decimal places, and this is stored in the scalar so that you can use it later. The DECIMALS procedure is also used to obtain a default if the DECIMALS parameter is not set.

The SKIP parameter specifies how many spaces are to be left before each element of each STRUCTURE; default 1.

The FREPRESENTATION parameter controls the printing of the factor values. The default is to print labels if there are any; if there are none, it is assumed that levels will be printed. The ordinals setting represents the values by the integers 1 upwards.

Option: IPRINT.
Parameters: STRUCTURE, FIELDWIDTH, DECIMALS, SKIP, FREPRESENTATION.

Action with RESTRICT

Any restrictions are ignored.

See also
Directive: PRINT.
Procedure: DECIMALS.
Genstat Reference Manual 1 Summary section on: Input and output.

MINIMIZE

Finds the minimum of a function calculated by a procedure (R.W. Payne).

Options

PRINT = *string tokens*	What output to produce (minimum, monitoring); default mini
FUNCTIONVALUE = *scalar*	Saves the minimum function value
DATA = any type	Data to be used with procedure _MINFUNCTION
MAXCYCLE = *scalar*	Maximum number of iterations; default 2000
NSTARTS = *scalar*	Maximum number of restarts; default 4
STEPADJUSTMENT = *scalar*	Adjustment to step lengths at each restart; default 0.1
EXIT = *scalar*	Indicates whether there has been convergence (zero) or non-convergence (non-zero)
TOLERANCE = *scalar*	Convergence criterion; default 0.0001
METHOD = *string token*	Algorithm for fitting nonlinear model (GaussNewton, NewtonRaphson, FletcherPowell); default Newt

Parameters

PARAMETER = *scalars*	Parameters to be estimated
LOWER = *scalars*	Lower bound for each parameter
UPPER = *scalars*	Upper bound for each parameter
STEPLENGTH = *scalars*	Step length for each parameter
INITIAL = *scalars*	Initial value for each parameter

Description

MINIMIZE searches for the minimum of a function that is calculated by a procedure _MINFUNCTION, which operates similarly to the RESAMPLE procedure that is called by procedures BOOTSTRAP and JACKKNIFE. This means that you can use any Genstat command to obtain the function value (e.g. ANOVA, FIT, SVD and so on). Any data structures that are needed by _MINFUNCTION to calculate the value of the function should be listed by the DATA option. Details are given in the Methods Section.

The parameters to be estimated in the minimization are listed by the PARAMETER parameter of MINIMIZE. Step lengths and initial values must be supplied using the STEPLENGTH and INITIAL parameters. You can also specify lower bounds with the LOWER parameter, and upper bounds with the UPPER parameter.

The PRINT option controls printed output with the settings:

minimum	to print the minimum function value and parameter values, and
monitoring	to print to monitor information showing the progress of the minimization.

By default, PRINT=minimum.

The MAXCYCLE option sets a limit on the number of function evaluations that are made by _MINFUNCTION (default 5000). The NSTARTS option controls how many times the optimization is restarted during the optimization, and the STEPADJUSTMENT option controls how the step lengths are adjusted at each restart; for more information, see *Method*.

If the optimization is successful, the scalars specified by the PARAMETER parameter will contain the estimated values of the parameters. The FUNCTIONVALUE option can save the minimum value.

The optimization search is performed by calling the FITNONLINEAR directive successively (see Method). You can also save a scalar, using the EXIT option, to indicate whether the minimization was successful. A zero value indicates success. Non-zero values indicate the

various types of failure codes as defined by the RKEEP directive. RKEEP can be also used to save other information from the optimization.

The TOLERANCE option corresponds to the TOLERANCE option of the RCYCLE directive, specifying the tolerance for convergence (default 0.0001). Similarly, the METHOD option selects the optimization method (default NewtonRaphson).

Options: PRINT, FUNCTIONVALUE, DATA, MAXCYCLE, NSTARTS, STEPADJUSTMENT, EXIT, TOLERANCE, METHOD.
Parameters: PARAMETER, LOWER, UPPER, STEPLENGTH, INITIAL.

Method

The procedure _MINFUNCTION, that calculates the function has two options. DATA supplies a pointer containing the data structures specified by the DATA option of MINIMIZE (so, DATA[1] is the first of these structures, DATA[2] is the second, and so on). FUNCTIONVALUE is a scalar, which should be set to the function value. There is one parameter, called PARAMETER. The PROCEDURE statement that defines _MINFUNCTION should set option PARAMETER=pointer. The parameters of the function can then be referred to as PARAMETER[1], PARAMETER[2], and so on (and these will be in the same order as in the PARAMETER parameter of MINIMIZE).

MINIMIZE calls the FITNONLINEAR directive successively to perform the optimzation search. The function evaluations so far are placed into a variate y, and terminated with a missing value. The expressions Func[1...3] defined below access the values successively, placing each one in turn into the scalar Target. This is identified as the value to minimize by supplying it as the setting of the FUNCTIONVALUE option of the MODEL directive. The missing final value for Target in y causes each use of FITNONLINEAR to terminate. The next parameter values for which FITNONLINEAR wants a function value are in the scalars x[]. _MINFUNCTION is called to obtain the value, this is then placed into y, and the process continues.

```
EXPRESSION Func[1...3]; VALUE=!e(i=i+1),\
  !e(ELEMENTS(x[];i)=PARAMETER[]),\
  !e(Target=ELEMENTS(y;i))
```

The use of FITNONLINEAR slows down as the number of function evaluations increases. So MINIMIZE allows the optimization to be restarted after MAXCYCLE/NSTARTS evaluations (where MAXCYCLE and NSTARTS are two of the options of MINIMIZE). At each restart, the step lengths are adjusted by multipying by a value specified by the STEPADJUSTMENT option.

For more information about function optimization using FITNONLINEAR, see the *Guide to Genstat, Part 2 Statistics*, Section 3.8.4.

Action with RESTRICT

The effects of restrictions on the data variables will depend on how the calculation is defined within the _MINFUNCTION procedure.

See also

Directive: FITNONLINEAR.
Procedures: DEMC, FPARETOSET, MIN1DIMENSION, SIMPLEX.
Genstat Reference Manual 1 Summary section on: Regression analysis.

MIN1DIMENSION

Finds the minimum of a function in one dimension (R.W. Payne).

Options

PRINT = *string tokens*	What output to produce (minimum, monitoring, plot); default mini
CALCULATION = *expression structures*	Expressions to calculate the target function
FUNCTIONVALUE = *scalars*	Identifier of the scalar, calculated by CALCULATION, whose value is to be minimized
DATA = any type	Data to be used with procedure _MIN1DFUNCTION
CRITERION = *string token*	Criterion for convergence (function, parameters); default func
MAXCYCLE = *scalars*	Maximum number of iterations; default 250
EXIT = *scalars*	Indicates whether there has been convergence (0) or non-convergence (1)
TOLERANCE = *scalars*	Convergence criterion; default 10^{-6} or variate

Parameters

PARAMETER = *scalars*	Parameters to be estimated
LOWER = *scalars*	Lower bound for each parameter
UPPER = *scalars*	Upper bound for each parameter
STEPLENGTH = *scalars*	Step length for each parameter
INITIAL = *scalars*	Initial value for each parameter

Description

MIN1DIMENSION searches for the minimum of a function in one dimension. The parameters to be estimated by the minimization are listed by the PARAMETER parameter of MIN1DIMENSION. Step lengths and initial values must be supplied using the STEPLENGTH and INITIAL parameters. When there are several parameters, these also define the dimension in the parameter space over which the function is minimized. Within each step of the minimization, the same multiple is used for the step length of every parameter. So the dimension is defined as the set of parameter values that can be calculated as

```
PARAMETER[1...p] = INITIAL[1...p] + Move * STEPLENGTH[1...p]
```

where Move is a scalar, and p is the number of parameters. You can also specify lower bounds with the LOWER parameter, and upper bounds with the UPPER parameter.

The function can be defined by specifying a list of Genstat calculation structures with the CALCULATION option, similarly to the way in which functions for optimization are specified for the FITNONLINEAR directive (see the *Guide to the Genstat Command Language, Part 2 Statistics*, Section 3.8). For example, you could find the minimum of the function 5*X-25*LOG(X) as follows.

```
EXPRESSION    Calc; VALUE=!e(Fx = 5 * X - 25 * LOG(X))
MIN1DIMENSION [PRINT=minimum,monitor,plot; EXIT=exit;\
              CALCULATION=Calc; FUNCTIONVALUE=Fx]\
              X; STEPLENGTH=1; INITIAL=1; LOWER=0.001
```

Alternatively, more complicated functions can be specified by defining a procedure _MIN1DFUNCTION, which operates similarly to the RESAMPLE procedure that is called by procedures BOOTSTRAP and JACKKNIFE. This is more complicated to specify, but it has the advantage that you can use any Genstat command to obtain the function value (e.g. ANOVA, FIT, SVD and so on). The DATA option is then used to list any data structures that are needed by _MIN1DFUNCTION to calculate the value of the function. Details are given in the Methods

Section.

The PRINT option controls printed output with the settings:

minimum	to print the minimum function value and parameter values,
monitoring	to print to monitor information showing the progress of the minimization, and
plot	to plot the function values around the initial values.

By default, PRINT=minimum.

The scalars specified by the PARAMETER parameter save the estimated values of the parameters, and the FUNCTIONVALUE scalar saves the minimum value. You can also save a scalar, using the EXIT option, which is set to 0 if the minimization was successful or to 1 if it did not converge.

The MAXCYCLE option sets a limit on the number of iterations; by default this is 250. The TOLERANCE option specifies the tolerance for convergence (default 10^{-6}), and the CRITERION option specifies what is tested. When CRITERION=function, convergence is achieved when the current function evaluations differ by less than the (scalar) value supplied by TOLERANCE. Alternatively, when CRITERION=parameters, the parameter values at the current evaluations must differ by less than TOLERANCE, which can then be set to either a scalar (to use the same tolerance with every parameter) or a variate (for different tolerances).

Options: PRINT, CALCULATION, FUNCTIONVALUE, DATA, CRITERION, MAXCYCLE, EXIT, TOLERANCE.

Parameters: PARAMETER, LOWER, UPPER, STEPLENGTH, INITIAL.

Method

MIN1DIMENSION performs a series of iterations in which three points are moved in the one dimension to locate the minimum. The idea is that the two outer points should bracket the minimum, while the inner point locates it.

The procedure _MIN1DFUNCTION, which you can use to calculate the function instead of the CALCULATION and FUNCTIONVALUE options, has two options. DATA supplies a pointer containing the data structures specified by the DATA option of MIN1DIMENSION (so, DATA[1] is the first of these structures, DATA[2] is the second, and so on). FUNCTIONVALUE is a scalar, which should be set to the function value. There is one parameter, called PARAMETER. The PROCEDURE statement that defines _MIN1DFUNCTION should set option PARAMETER=pointer. The parameters of the function can then be referred to as PARAMETER[1], PARAMETER[2], and so on (and these will be in the same order as in the PARAMETER parameter of MIN1DIMENSION). The definition below has the same effect as the expression

```
EXPRESSION Calc; VALUE=!e(Fx = 5 * X - 25 * LOG(X))
```

shown in the description.

```
PROCEDURE [PARAMETER=pointer] '_MIN1DFUNCTION'
" calculates the function for MIN1DIMENSION "
OPTION NAME=\
  'DATA', "(I: pointer) data to calculate the function"\
  'FUNCTIONVALUE'; "(O: scalar) returns the function value"\
  MODE=p; TYPE='pointer','scalar'
PARAMETER NAME=\
  'PARAMETER'; "(I: scalar) parameter values"\
  MODE=p; TYPE='scalar'; SET=yes; DECLARED=yes; PRESENT=yes
CALCULATE FUNCTIONVALUE = 5*PARAMETER[1] - 25*LOG(PARAMETER[1])
ENDPROCEDURE
```

The parameter X can then be estimated by the statement

```
MIN1DIMENSION [PRINT=minimum,monitor,plot; EXIT=exit]\
              X; STEPLENGTH=1; INITIAL=1; LOWER=0.001
```

Action with RESTRICT
The effects of restrictions on the data variables will depend on how the calculation is defined (by the CALCULATION option or within the _MIN1DFUNCTION procedure).

See also
Directive: FITNONLINEAR.
Procedures: DEMC, FPARETOSET, MINIMIZE, SIMPLEX.
Genstat Reference Manual 1 Summary section on: Regression analysis.

MMPREDICT

Predicts the Michaelis-Menten curve for a particular set of parameter values (M.C. Hannah).

Options

PLOT = *string tokens*	What to plot (concentration, rate); default conc
WINDOW = *scalar*	Window in which to plot the graphs; default 1
TITLE = *text*	Title for the graphs; default 'Michaelis-Menten process'
TTIMES = *text*	Title for the times axis; if this is unset, the identifier of the TIMES variate is used
TCONCENTRATIONS = *text*	Title for the concentrations axis; if this is unset, the identifier of the CONCENTRATIONS variate is used if available, otherwise 'Concentration'
TRATES = *text*	Title for the rates axis; if this is unset, the identifier of the RATES variate is used if available, otherwise 'Rate'

Parameters

PARAMETERS = *variates*	Variate with four values specifying the values of the parameters S_0, V_{max}, K_m and K to use to form the predictions
TIMES = *variates*	Times at which to make predictions
CONCENTRATIONS = *variates*	Saves the predicted substrate concentrations
RATES = *variates*	Saves the predicted reaction rates

Description

A generalized Michaelis-Menten equation, for biochemical reaction rate $v(t)$, versus substrate concentration $S(t)$ at time t may be written as

$$v(t) = dS(t) / dt = V_{max} (S(t) - K_1) / (K_m + S(t) - K_1)$$

This can be fitted to concentration and time data in Genstat using the MICHAELISMENTEN procedure.

If we have values for the parameters, including an initial concentration S_0, we might like to predict $S(t)$ and/or its derivative $v(t)$ at various times. This seems simple until it is realized that there is no closed-form expression for $S(t)$. Thus this procedure uses the method of Golicnik (2010) to calculate $S(t)$ and $v(t)$. The required times must be specified in a variate, by the TIMES parameter. Values for S_0, V_{max}, K_m and K must be supplied in a variate (in that order), by the PARAMETERS parameter.

The PLOT option controls the graphs that are plotted, with settings

concentration	to plot the curve fitted to the concentrations, and
rate	to plot the estimated reaction rates against the concentrations, and against time.

By default, PLOT=concentration.

The WINDOW option specifies the window to use for the graphs (default 1). The TITLE option can specify an overall title, and the TCONCENTRATIONS, TRATES and TTIMES options can specify titles for the axes for concentration, rate and time, respectively.

The values predicted for $S(t)$ and $v(t)$ can be saved, in variates, by the CONCENTRATIONS and RATES parameters.

Options: PLOT, WINDOW, TITLE, TTIMES, TCONCENTRATIONS, TRATES.
Parameters: PARAMETERS, TIMES, CONCENTRATIONS, RATES.

Reference

Golicnik, M. 2010. Explicit reformulations of time-dependent solution for a Michaelis-Menten enzyme reaction model. *Analytical Biochemistry*, **406**, 94-96.

See also

Procedure: `MICHAELISMENTEN`.

Genstat Reference Manual 1 Summary section on: Regression analysis.

MNORMALIZE

Normalizes two-colour microarray data (D.B. Baird).

Options

PRINT = *string tokens*	What to print (summary, slidesummary, monitoring); default summ, slid, moni
PLOT = *string tokens*	What plots to produce (pineffects, roweffects, columneffects, intensityeffects, rowxcoleffects, ma, standardizedma, spatialresiduals); default * i.e. none
METHOD = *string token*	What type of model components to fit (spline, loess); default spli
MODELTERMS = *string tokens*	What model components to fit (pins, rows, columns, intensity, pinxintensity, ar1, rowxcolumn, pinxrow, pinxcolumn); default pins, rows, colu, inte
DFINTENSITY = *scalar*	Degrees of freedom for intensity cubic spline; default 24
DFROWXCOLUMN = *scalar*	Degrees of freedom for row × col thinplate spline; default 49
POORFLAGS = *text* or *variate*	Levels of FLAGS that are poor quality spots
BADFLAGS = *text* or *variate*	Levels of FLAGS that are bad spots
ARRANGEMENT = *string token*	Whether to use trellis or single plots (single, trellis); default trel
WINDOW = *scalar*	Window number for the graphs; default 3
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameters

LOGRATIOS = *variates* or *pointers*	Log-ratios
INTENSITIES = *variates* or *pointers*	Spot intensities
SLIDES = *factors* or *texts*	Slides
PINS = *factors*	Pins
SROWS = *factors*	Rows across whole slide
SCOLUMNS = *factors*	Columns across whole slide
PROWS = *factors*	Rows within pins
PCOLUMNS = *factors*	Columns within pins
FLAGS = *factors* or *pointers*	Quality flags
CLOGRATIOS = *variates* or *pointers*	Save corrected log-ratios
SLOGRATIOS = *variates* or *pointers*	Save standardized log-ratios
SDSMOOTH = *variates* or *pointers*	Save smoothed deviations
PINEFFECTS = *tables*	Save estimated pin effects
ROWEFFECTS = *tables*	Save estimated row effects
COLEFFECTS = *tables*	Save estimated column effects
INTEFFECTS = *variates* or *pointers*	Save estimated intensity effects
CLRED = *variates* or *pointers*	Save corrected log2 red values
CLGREEN = *variates* or *pointers*	Save corrected log2 green values
VAREXPLAINED = *variates*	Save the variance explained by slide

Description

With large microarrays it is essential to identify sources of variation and correct for them, to allow for robust use of this technology. Through normalization procedures, such variations can be identified and removed to obtain data for follow-on research. The analysis of the microarrays is thus a two-step process: a within-slide analysis aimed at normalization and, if required, standardization; then a between-slide analysis to estimate the differences between targets (or treatments) and evaluate their consistency.

Various techniques have been suggested for normalization, including linear regression, ratio statistics, local smoothing and analysis of variance. The approach in MNORMALIZE is to model the variation associated with spatial and structural components and remove this as noise. Examples of spatial components are the grid layout on the slide (rows × columns), and of structural components are the pins, print order and differential dye responses to binding and scanning. The model can be specified to fit the type of variation found in the particular series of slides. The usual statistical modelling approach is taken where all possible sources of noise are jointly fitted in one model, and the need for each term is assessed using the statistical significance of the reduction in the remaining unexplained variation. Model terms can be added or removed as required. The fitted model then indicates where useful modification of protocols and equipment would help minimize variation in future experiments.

The type of model to use is selected using the METHOD option, with settings:

spline	a mixed model including cubic smoothing splines, fitted with the REML directive; or
loess	regression with the LOESS smoothing function, fitted with the FIT directive.

The terms to include in the models are selected by the MODELTERMS option, with settings:

pins	an effect for each pin on the slide;
rows	an effect for each row on the slide;
columns	an effect for each column on the slide;
intensity	a cubic smoothing spline or Loess curve for spot intensity, with degrees of freedom defined by the DFINTENSITY option (default 24);
pinxintensity	a different linear effects of intensity for each pin;
ar1	autoregressive model with order 1, separately in row and column directions (REML only);
rowxcolumn	a thin-plate spline (REML only) which fits a smooth surface with row and column interaction, with degrees of freedom defined by the DFROWXCOLUMN option (default 49);
pinxrow	pin-by-row interaction; and
pinxcolumn	pin-by-column interaction.

The log-ratios and spot intensities are supplied by the LOGRATIOS and INTENSITIES parameters. If these are single variates, the SLIDES parameter should supply a factor to index the slides. Alternatively you can supply pointers containing a variate for each slide for these, and the SLIDES parameter may be omitted; alternatively it can supply a text giving a label for each slide.

The slide layout is specified by the parameters PINS, SROWS, SCOLUMNS, PROWS and PCOLUMNS. PINS provides a factor to index the pins. SROWS and SCOLUMNS provide factors to index the rows and columns within the whole slide. PROWS and PCOLUMNS provides factors to index the rows and columns within the pins. If LOGRATIOS is a pointer, the slide layout factors refer to a single slide, and all slides must have a common layout.

The FLAGS parameter supplies a factor giving a quality flag for each spot, which must match the type and length of the LOGRATIOS parameter. The POORFLAGS and BADFLAGS options can then each supply a text or variate, defining levels of FLAGS that indicate poor or bad quality

spots. The poor spots are still used for model fitting, but are excluded from the output variates. The bad quality spots are excluded from any analysis.

The CLOGRATIOS parameter can supply a variate or pointer, to save the corrected log-ratios. Similarly, the SLOGRATIOS parameter can save the standardized log-ratios, and SDSMOOTH can save the smoothed deviations. The PINEFFECTS, ROWEFFECTS and COLEFFECTS parameters can save tables containing estimated pin, row and column effects, respectively. The INTEFFECTS parameter can save the estimated intensity effects. The CLRED and CLGREEN parameters can save the corrected $\log_2$ red and green values, respectively. If they have already been defined, the output structures specified by CLOGRATIOS, SLOGRATIOS, SDSMOOTH, INTEFFECTS, CLRED and CLGREEN must have the same type as the LOGRATIOS parameter (i.e. variates if LOGRATIOS is a variate, and pointers if LOGRATIOS is a pointer). Finally, the VAREXPLAINED parameter can save a variate with the variance explained by the fitted model on each slide.

The PRINT option controls printed output, and the PLOT option controls what graphs are produced. By default the plots for the slides are displayed in a trellis arrangement, but you can set option ARRANGEMENT=single to display them separately, in single plots. The WINDOW option specifies the window to use for the graphs (by default 3). You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option then supplies a template for the file names.

Options: PRINT, PLOT, METHOD, MODELTERMS, DFINTENSITY, DFROWXCOLUMN, POORFLAGS, BADFLAGS, ARRANGEMENT, WINDOW, DEVICE, GRAPHICSFILE.

Parameters: LOGRATIOS, INTENSITIES, SLIDES, PINS, SROWS, SCOLUMNS, PROWS, PCOLUMNS, FLAGS, CLOGRATIOS, SLOGRATIOS, SDSMOOTH, PINEFFECTS, ROWEFFECTS, COLEFFECTS, INTEFFECTS, CLRED, CLGREEN, VAREXPLAINED.

Action with RESTRICT

Any restrictions on LOGRATIOS, INTENSITIES, SLIDES, PINS, SROWS, SCOLUMNS, PROWS, PCOLUMNS or FLAGS are removed (and a warning is given).

See also

Procedures: DMADENSITY, FDRBONFERRONI, FDRMIXTURE, MACALCULATE, MAESTIMATE, MAHISTOGRAM, MAPCLUSTER, MAPLOT, MASCLUSTER, MASHADE, MAVOLCANO, MA2CLUSTER.

Genstat Reference Manual 1 Summary section on: Microarray data.

MOVINGAVERAGE

Calculates and plots the moving average of a time series (R.P. Littlejohn, G. Tunnicliffe Wilson & D.B. Baird).

Options

PRINT = *string token*	What to print (parameters); default * i.e. nothing
NSAMPLES = *scalar*	Number of samples used to calculate each moving average
METHOD = *string token*	How to calculate the averages (past, centred, exponential, filter, holtwinters) default past
ORDER = *scalars*	Order for polynomial smoothing (0, 1, 2, 3, 4); default 0 i.e. ordinary moving-averages calculated from means
TRIM = *string token*	Whether to trim transients with METHOD settings past or centre when ORDER=0 (yes, no); default no
PLOT = *string token*	What to plot (components, movingaverages, predictions); default * i.e. nothing
ALPHA = *scalar*	Allows the smoothing parameter for the contribution of the last value in the series to the moving average to be specified for the exponential or Holt-Winters methods
BETA = *scalar*	Allows the smoothing parameter for the trend to be specified for the Holt-Winters method
GAMMA = *scalar*	Allows the smoothing parameter for the seasonal component to be specified for the Holt-Winters method
MULTIPLICATIVE = *string token*	Controls whether the seasonal component is multiplicative in the Holt-Winters method (yes, no); default no
NPREDICTIONS = *scalar*	Number of predicted values to form for the Holt-Winters method; default is twice the number of levels of the SEASONAL factor, or 2 if SEASONAL is not set

Parameters

SERIES = *variates*	Time series whose moving averages are required
MASERIES = *pointers*	Saves the moving averages for the defined ORDER settings
TITLE = *texts*	Title for the graph
SEASONAL = *factors*	Factor for seasonal adjustment
SAVE = *pointers*	Saves results from the Holt-Winters method or from seasonal adjustment

Description

MOVINGAVERAGE calculates and plots an unweighted or exponentially weighted moving average of a time series, or uses TFILTER with two-sided ARIMA smoothing of transients. This allows you to smooth out short-term volatility and assess longer-term trends or cycles in the data.

The method of averaging is specified by the METHOD option, with settings:

past	takes an unweighted average of past values (default);
centred	takes an average centred on the current value with the first and last values receiving weights of 0.5 when NSAMPLES is even;
exponential	takes an exponentially weighted average of past values;
filter	uses TFILTER to smooth the data, using a specially constructed ARIMA model; and

`holtwinters` uses the Holt-Winters method; see Holt (1957) and Winters (1960).

The time series is specified, in a variate, using the `SERIES` parameter. The moving averages can be saved using the `MASERIES` parameter. They are saved in a pointer with a suffix for each setting of the `ORDER` option.

The `SEASONAL` factor can specify a factor to perform a seasonal adjustment of the moving average. The residuals (the observed values minus the moving average) are calculated and averaged for each level of the factor. These averages for each level are then subtracted from the corresponding units of the moving average, so that the mean residual for each level is now zero.

The `NSAMPLES` option specifies the number of data points that are used to calculate the moving average. When `METHOD=exponential` the weighting parameter can be specified by the `ALPHA` option. If this is unset, the default is calculated as

$\alpha = 2 / (\texttt{NSAMPLES} + 1)$.

In the `filter` method, most of the weight is spread over an `NSAMPLES` range centred upon each point. Outside this range, the weights go slightly negative before dying away.

With `METHOD` settings `past` or `centre`, the `ORDER` option can be used to request polynomial smoothing. The default is `ORDER`=0, which gives ordinary moving averages calculated from the means of the defined range of values. Alternatively, you can set `ORDER` to values in the range 1-4 to calculate the averages by fitting polynomials of those orders to the data values in the defined range. `NSAMPLES` should then be an integer greater than the requested `ORDER`. With these `METHOD` settings, the transients at either end of the series are trimmed by default, but they can be evaluated by setting option `TRIM=yes`.

The Holt-Winters method uses a weighted average of past estimates of the level and trend in estimating the current value. The smoothing parameters can be specified by the `ALPHA` and `BETA` options. These control the balance between past and current contributions to the estimates of the level and trend respectively. They must take a value between 0 and 1, and the closer they are to zero, the less the current value contributes to the estimate, giving greater smoothing of the series. When the `SEASONAL` parameter is set, the Holt-Winters moving average uses a weighted estimate of the seasonal effects. The smoothing parameter that controls the balance between past and current contributions to the seasonal effects can be specified by the `GAMMA` option. If the parameters are not specified, their values are estimated by minimizing the prediction sums of squares. You can set option `PRINT=parameters` to print the values of the parameters. By default, the seasonal component is used in an additive model

estimate = level + trend + season

but you can set option `MULTIPLICATIVE=yes` to use a multiplicative model

estimate = (level + trend) × season.

When `METHOD=holtwinters`, the `NPREDICTIONS` option specifies the number of predicted points to form at the end of the original series. The default is twice the number of levels of the `SEASONAL` factor, or two if `SEASONAL` is not set.

The graphs that are produced by `MOVINGAVERAGE` are controlled by the `PLOT` option, with settings:

`components` to plot the separate components (trend, level and season) of the estimate from a Holt-Winters or a seasonal model,

`movingaverages` to plot the moving averages, together with the original series, and

`predictions` to plot the predicted values with 95% confidence limits at the end of the series for a Holt-Winters model.

By default nothing is plotted. The `TITLE` parameter can supply a title for the graphs. The default is to construct a title automatically from the name of the series variate and the type of moving average.

The `SAVE` parameter allows you to save a pointer with results from the Holt-Winters method

or from seasonal adjustment. For Holt-Winters, the pointer has elements `'Level'`, `'Trend'`, `'Season'` (if `SEASONAL` is set), `'Parameters'` and, if `NPREDICTIONS` is greater than zero, `'Lower'`, and `'Upper'`. With other seasonal adjustment models, it has elements for `'Trend'`, `'Season'`. However, if polynomial smoothing is being used, the pointer has two levels of suffix, with these at the second level, and the polynomial components as the first level.

Options: `PRINT`, `NSAMPLES`, `METHOD`, `ORDER`, `TRIM`, `PLOT`, `ALPHA`, `BETA`, `GAMMA`, `MULTIPLICATIVE`, `NPREDICTIONS`.
Parameters: `SERIES`, `MASERIES`, `TITLE`, `SEASONAL`, `SAVE`.

Method
The procedure uses `MODEL` and `FIT` to do polynomial smoothing. The `filter` setting uses an ARIMA model with forward and backward filtering, to implement a Wiener signal extraction procedure in a similar manner to the example of the `TFILTER` directive illustrated in Example 7.6.1b of the *Guide to the Genstat Command Language, Part 2 Statistics*.

The filter is designed to estimate a regularly sampled continuous time integrated random walk (the signal), to which random noise has been added. At the sampled points the signal follows an ARIMA(0,2,1) process, and the observations follow an ARIMA(0,2,2) process. The parameters of these processes depend on the signal to noise ratio, which is imputed from the specified value of `NSAMPLES`. An important aspect of the implementation is that the assumed stochastic properties of the signal allow one to reduce the end effect (transients) by implicit forecasting and backforecasting, to get a fully optimal filter for the finite set of observations. The result is, in fact, exactly equivalent to fitting a cubic smoothing spline with knots at each observation point. Example 7.6.1b in the Guide implements a similar filter, but for a signal that follows the more simple random walk process, and the result is a two-sided exponential smoother that is equivalent to a linear smoothing spline. To be effective, the assumed model does not have to be correct. The procedure just removes high frequency variations with relatively small distortion of the lower frequency variations, the cut off between high and low frequencies being determined by the setting of `NSAMPLES`.

For the Holt-Winters model, the prediction sum of squares is minimized by `FITNONLINEAR` using functions in `CurveFuncs.dll`, if this is available in the version of Genstat that is being used. Otherwise, or if `FITNONLINEAR` does not find a solution, it is minimized by the `MINIMIZE` procedure. The calculations are then done by procedures `_HWNFUNCTION`, `_HWAFUNCTION` and `_HWMFUNCTION`, that are subsidiary procedures of `MOVINGAVERAGE`

Action with `RESTRICT`
Restrictions are not permitted.

References
Holt, C.C. (1957). Forecasting trends and seasonals by exponentially weighted moving averages. *ONR Research Memorandum*, **52**.

Winters, P.R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, **6**, 324–342.

See also
Genstat Reference Manual 1 Summary section on: Time series.

MPOLISH

Performs a median polish of two-way data (D.B. Baird).

Options

MAXCYCLE = *scalar*	Maximum number of iterations; default 50
TOLERANCE = *scalar*	Tolerance for convergence; default 0.0001

Parameters

DATA = *variates* or *pointers* or *matrices* or *tables*	Two-way data to be polished
ROWS = *factors*	Row definitions for a DATA variate
COLUMNS = *factors*	Column definitions for a DATA variate
ROWEFFECTS = *variate*	Row effects removed from polished results
COLEFFECTS = *variate*	Column effects removed from polished results
POLISH = *variates* or *pointers* or *matrices* or *tables*	Polished result in same format as DATA
CENTRE = *scalars*	Estimate of overall centre point

Description

MPOLISH performs a median polish of two-way data, supplied by the DATA parameter. This can be a two-dimensional table, a matrix or a pointer of variates. Alternatively, it can be a single variate. The rows and columns are then defined by factors supplied by the ROWS and COLUMNS parameters, or by just the COLUMNS parameter with the data valueS assumed to be sorted into row order within each column.

The MAXCYCLE option sets a limit on the number of iterations. The TOLERANCE option specifies the convergence criterion: convergence occurs when

```
ABS(1-SUM(ABS(OldPolish)) / SUM(ABS(NewPolish))) < TOLERANCE
```

The polished data can be saved by the POLISH parameter, row effects by the ROWEFFECTS parameter, column effects by the COLEFFECTS parameter, and the overall centre point by the CENTRE parameter.

Options: MAXCYCLE, TOLERANCE.
Parameters: DATA, ROWS, COLUMNS, ROWEFFECTS, COLEFFECTS, POLISH, CENTRE.

See also

Procedures: MPOLISH, ROBSSPM, TUKEYBIWEIGHT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

MPOWER

Forms integer powers of a square matrix (P.W. Lane).

No options

Parameters

MATRIX = *matrices*, *symmetric matrices* or *diagonal matrices*	Matrix from which to form the power
POWER = *scalars*	Power to which each matrix is to be raised
RESULT = *identifiers*	Structure to store the result

Description

MPOWER forms powers of a square matrix, using as few matrix operations as possible in order to save time and decrease rounding errors. The square matrix is specified using the MATRIX parameter, and can be either an ordinary matrix structure (with an equal number of rows and columns), a symmetric matrix or a diagonal matrix. The required power, which must be a positive integer, is specified using the POWER parameter. The RESULT parameter supplies the identifier of the structure to save the results; this will be declared automatically to be of the same type as the input structure.

Options: none. Parameters: MATRIX, POWER, RESULT.

Method

For general matrices, successive powers of two of the matrix are formed by matrix products, and the result formed by taking the product of those that are needed to achieve the specified power. Diagonal matrices are dealt with using simple exponentiation of the diagonal values. Symmetric matrices are spectrally decomposed, and the result formed as a product of the matrix containing the latent vectors (V) with the simple power of the diagonal matrix containing the latent roots (R):

```
RESULT = V *+ R**POWER *+ TRANSPOSE(V).
```

See also

Function: MPOWER .

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

MSEKERNEL2D

Estimates the mean square error for a kernel smoothing (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

Y = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
X = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NSTEP = *scalars*	How many values of the kernel width to use; no default – this parameter must be set
HMAX = *scalars*	Maximum values for the kernel width; no default – this parameter must be set
HVALUES = *variates*	Variates to receive the values of the kernel width
MSE = *variates*	Variates to receive the estimated mean square error for each value of the kernel width

Description

This procedure calculates an estimate of the mean square error for a kernel smoothing given a particular kernel width. The method used is that of Berman & Diggle (1989). The data required by the procedure are the coordinates of a spatial point pattern (specified using the parameters X and Y), the coordinates of a polygon within which smoothing is to be performed (specified using the parameters XPOLYGON and YPOLYGON), the number of values of the kernel width at which to estimate the mean square error (specified using the parameter NSTEP), and the maximum kernel width to use (specified using the parameter HMAX). The output of the procedure is a variate containing a sequence of NSTEP equally-spaced values of the kernel width parameter from HMAX/NSTEP up to HMAX, and a corresponding vector containing the mean square error for each kernel width. The values of the kernel width and the corresponding mean square error estimates can be saved using the parameters HVALUES and MSE.

Printed output is controlled using the PRINT option. The default setting of summary prints the values of the kernel width and the corresponding mean square error estimates under the headings HVALUES and MSE.

The output of the procedure may be used to select the optimum kernel width to use with the procedure PTKERNEL2D. Note that the estimated mean square errors returned by the procedure are, in fact, scaled estimates. The scaling simplifies the calculations but it can produce negative estimates of mean square errors. The scaling is, however, independent of the kernel width, so that the true mean square error has its minimum at the same kernel width as the scaled version.

Option: PRINT.
Parameters: Y, X, YPOLYGON, XPOLYGON, NSTEP, HMAX, HVALUES, MSE.

Method
A procedure PTCHECKXY is called to check that X and Y have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The procedure then calculates a sequence of NSTEP equally-spaced values for the kernel width, starting at HMAX/NSTEP and finishing at HMAX. It then calls a procedure PTPASS to call a Fortran program to calculate the estimated mean square error associated with each value of the kernel width.

Action with RESTRICT
If X and Y are restricted, only the subset of values specified by the restriction will be included in the calculations. XPOLYGON and YPOLYGON may also be restricted, as long as the same restrictions apply to both parameters.

Reference
Berman, M. & Diggle, P.J. (1989). Estimating weighted integrals of the second-order intensity of a spatial point process. *Journal of the Royal Statistical Society, Series B*, **51**, 81-92.

See also
Procedures: KERNELDENSITY, PTKERNEL2D, PTK3D.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

MTABULATE

Forms tables classified by multiple-response factors (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (counts, totals, nobservations, means, minima, maxima, variances, quantiles, sds, skewness, kurtosis, semeans, seskewness, sekurtosis); default * i.e. none
CLASSIFICATION = *factors*	Non multiple-response factors classifying the tables
MRESPONSE = *pointers*	Pointers to factors defining the multiple-responses for the tables
MRFACTOR = *identifiers*	Identifier of factors to index the sets of multiple responses in the tables
COUNTS = *table*	Saves a table counting the number of units with each factor combination; default *
MARGINS = *string token*	Whether the tables should be given margins (yes, no); default no
WEIGHTS = *variate*	Weights to be used in the tabulations; default * indicates that all units have weight 1
PERCENTQUANTILES = *scalar* or *variate*	Percentages for which quantiles are required; default 50 i.e. median

Parameters

DATA = *variates*	Data values to be tabulated
TOTALS = *tables*	Tables to contain totals
NOBSERVATIONS = *tables*	Tables containing the numbers of non-missing values in each cell
MEANS = *tables*	Tables of means
MINIMA = *tables*	Tables of minimum values in each cell
MAXIMA = *tables*	Tables of maximum values in each cell
VARIANCES = *tables*	Tables of cell variances
QUANTILES = *tables* or *pointers*	Table to contain quantiles at a single PERCENTQUANTILE, or pointer of pointers to tables for several PERCENTQUANTILES
SDS = *tables*	Tables of standard deviations
SKEWNESS = *tables*	Tables of skewness coefficients
KURTOSIS = *tables*	Tables of kurtosis coefficients
SEMEANS = *tables*	Tables of standard errors of means
SESKEWNESS = *tables*	Tables of skewness coefficients
SEKURTOSIS = *tables*	Tables of kurtosis coefficients

Description

Multiple responses occur in surveys as the result of open-ended questions like "Which cities have you visited?". In Genstat, these can be formed by the FMFACTORS procedure and are represented by a pointer containing a factor for each possible response code. The factors have levels 0 and 1, and corresponding labels 'absent' and 'present'. If the original response codes were textual, the various strings are used as labels of the pointer; while if they were numerical, the numbers are used as the pointer suffixes.

The multiple responses for the tables are specified by the MRESPONSE option, while any

ordinary factors are specified by the CLASSIFICATION option. The MARGINS option indicates whether or not the tables are to contain margins. For the multiple responses, these represent summaries not over the responses but over the respondents (who may each have given several responses). MTABULATE needs to generate an ordinary factor to classify the dimension of the tables corresponding to each set of multiple responses. You can supply identifiers for these factors (thus allowing them to be accessed outside the procedure), using the MRFACTOR option.

The other options and parameters are similar to those of the TABULATE directive. The COUNTS option can save a table containing the frequencies of the various responses. The DATA parameter provides information about the respondents who made the multiple responses. (So, for example, you could set DATA to the incomes of the respondents and then tabulate the average incomes of the people who have visited each of the cities.) The other parameters allow you to save the various types of numerical summary: totals, numbers of non-missing values, means, minima, maxima, variances, quantiles, standard deviations, skewness and kurtosis coefficients and (within-cell) standard errors of means, skewness and kurtosis.

The PERCENTQUANTILES option specifies which quantiles you want. By default just the median (the 50% quantile) is produced. However, you can set PERCENTQUANTILES to a scalar to request another percentage point, or to a variate to request several. The QUANTILE parameter will then return a pointer with length equal to the required number of quantiles, instead of a single table.

The PRINT option allows you to print the tables (as well as, or instead of, saving them). By default nothing is printed.

Options: PRINT, CLASSIFICATION, MRESPONSE, MRFACTOR, COUNTS, MARGINS, WEIGHTS, PERCENTQUANTILES, SDS, SKEWNESS, KURTOSIS, SEMEANS, SESKEWNESS, SEKURTOSIS.
Parameters: DATA, TOTALS, NOBSERVATIONS, MEANS, MINIMA, MAXIMA, VARIANCES, QUANTILES.

Method

MTABULATE uses TABULATE to form tables for each multiple response or combination of multiple responses, and then EQUATE to put them all into a single table.

Action with RESTRICT

MTABULATE takes account of any restrictions on the classification or multiple-response factors or the DATA or WEIGHT variates.

See also

Directives: TABULATE.
Procedures: FMFACTORS, FFREERESPONSEFACTOR, SVTABULATE.
Genstat Reference Manual 1 Summary section on: Survey analysis.

MULTMISSING

Estimates missing values for units in a multivariate data set (H.R. Simpson & R.P. White).

Option

MAXCYCLE = *scalar*	Defines the maximum allowed number of iterations; default 10

Parameters

DATA = *pointers*	Each pointer contains a set of variates whose missing values are to be estimated; these will be overwritten by the estimates unless the OUT parameter is specified
OUT = *pointers*	Each pointer contains a set of variates to hold the results

Description

MULTMISSING estimates missing values for units in a multivariate data set, using an iterative regression technique. The input for the procedure is a set of variates contained in a pointer specified by the DATA parameter. The output can be saved in a different set of variates by supplying a similar pointer with the parameter OUT; if this is absent, the output values will overwrite the values of the variates given by DATA. The maximum number of iterations is set by the option MAXCYCLE, with a default of 10. If MAXCYCLE is set to zero, missing values will be replaced by variate means calculated from the units that have no values missing for any of the variates.

Option: MAXCYCLE.
Parameters: DATA, OUT.

Method

Initial estimates of the missing values in each variate are formed from the variate means using the values for units that have no missing values for any variate. Estimates of the missing values for each variate are then recalculated as the fitted values from the multiple regression of that variate on all the other variates. When all the missing values have been estimated the variate means are recalculated. If any of the means differs from the previous mean by more than a tolerance (the initial standard error divided by 1000) the process is repeated, subject to a maximum number of repetitions defined by the MAXCYCLE option.

The default maximum number of iterations (10) is usually sufficient when there are few missing values, say two or three. If there are many more, 20 or so, it may be necessary to increase the maximum number of iterations to around 30.

The method is similar to that of Orchard & Woodbury (1972), but does not adjust for bias in the variance-covariance matrix as suggested by Beale & Little (1975).

Action with RESTRICT

All the variates must be unrestricted, or they must all be restricted to the same set of units; otherwise a fault will occur in a CALCULATE statement within MULTMISSING.

References

Beale, E.M.L. & Little, R.J.A. (1975). Missing values in multivariate analysis. *Journal of the Royal Statistical Society, Series B*, **37**, 129-145.

Orchard, T. & Woodbury, M.A. (1972). A missing information principle: theory and applications. In: *Proceedings of the 6th Berkeley Symposium in Mathematical Statistics and Probability, Vol I*, 697-715.

See also
Directive: INTERPOLATE.
Procedures: ANTMVESTIMATE, SVHOTDECK, QMVREPLACE.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Calculations and manipulation.

MVAOD

Does an analysis of distance of multivariate data (R.W. Payne & R.P. White).

Options

`PRINT` = *string tokens*	Controls printed output (`aodtable`, `permutationtest`); default `aodt`
`TERMS` = *formula*	Model terms to fit in the analysis; must be specified
`FACTORIAL` = *scalar*	Limit on the number of factors or variates in a term for it to be included in the analysis; default 3
`NTIMES` = *scalar*	Number of permutations to use in the permutation test; default 999
`SEED` = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically

Parameters

`DATA` = *symmetric matrices*	Supplies the squared distances between the data points
`SSD` = *variates*	Saves the sums of squared distances
`DF` = *variates*	Saves the numbers of degrees of freedom
`PRPERMUTATION` = *variates*	Saves probabilities from the permutation test
`DISTANCES` = *pointers*	Contains a symmetric matrix of distances for each model term

Description

This procedure implements the analysis of multivariate distance devised by Gower & Krzanowski (1999). This is useful when you have units whose positions in multi-dimensional space may be explained by a linear statistical model. It provides a breakdown of the sums of squared distances between the units, similar to that provided for sums of squares in an analysis of variance. So, the total squared distance between the units is partitioned into the components that can be explained by each of the terms in the model. These cannot be tested directly as in an analysis of variance, as it is unclear what probability distributions would be appropriate. Instead the importance of the terms can be assessed by doing a permutation test, in which the several permutations of the units are made, and the significances of the sums of squared distances from the observed data are calculated by seeing where they lie in the distribution of values obtained from all the analyses (the original analysis and those of the permuted data sets).

The squared distances between the units must be supplied in a symmetric matrix, using the `DATA` parameter. In some situations, these may be actual distances. Alternatively, the units may often be described by a collection of attributes ranging from continuous measurements to categorical variables, like the presence or absence of a particular feature. In these circumstances, the `FSIMILARITY` directive can be used combine these attributes to give a symmetric matrix that represents the similarity between each pair of units. This can then be converted into a squared distance matrix, for example, by subtracting the similarities from one. (So `MVAOD` can be regarded as providing an alternative to multivariate analysis of variance, for units whose attributes are not all continuous variables.)

The model to fit in the analysis is specified by the `TERMS` option. The `FACTORIAL` option sets a limit on the number of factors of variates that the terms can contain; any terms with more factors of variates are deleted from the analysis.

Printed output is controlled by the `PRINT` option, with settings:

`aodtable`	for an analysis-of-distance table, giving the sums of squared distances and numbers of degrees of freedom for each model term; and

`permutationtest` adds a column to the analysis-of-distance table containing probabilities from the permutation test.

The `NTIMES` option specifies the number of permutations to perform; the default is 999. The `SEED` option specifies the seed to use to generate the random numbers that are used to select the permutations; the default of zero continues the sequence of random numbers from a previous generation or, if none have yet been used in this Genstat job, it initializes the seed automatically. `MVAOD` checks whether `NTIMES` is greater than the number of possible permutations available for the data set. If so, it does an exact test instead, which uses each possible permutation once.

The `SSD`, `DF` and `PRPERMUTATION` parameters allow you to save the sums of squared distances, degrees of freedom and permutation probabilities. These are each saved in a variate, with each unit labelled by the name of the model term concerned. There are also two final units in each variate to save the corresponding information for residual and the total.

The `DISTANCES` parameter can save a pointer containing a symmetric matrix for each model term. Each matrix has a row for each combination of levels of the factors in the corresponding term, and its values are the distances between the factor combinations in the multi-dimensional space defined by the possible effects of the term. So, to investigate the relationships between the effects of the term, you could convert the `DISTANCES` to similarities, and then use them as input for a principal coordinates analysis (see `PCO` for details).

Options: `PRINT`, `TERMS`, `FACTORIAL`, `NTIMES`, `SEED`.
Parameters: `DATA`, `SSD`, `DF`, `PRPERMUTATION`, `DISTANCES`.

Method

The method of analysis is described by Gower & Krzanowski (1999) and Krzanowski (2002), who show that the sum of squares of distances for each term *i* is given by

```
TRACE( Proj[i] *+ DATA *+ Proj[i]) / 2
```

where `Proj[i]` is a projection matrix for the term. If the model contains only factors, `MVAOD` uses `ANOVA` to check whether the model is orthogonal and, if so, it calculates the projection matrices using the method described by Payne & Tobias (1992). For a non-orthogonal model, `MVAOD` adjusts the design matrix `X[i]` of each term *i* for the earlier terms by using its columns as y-variates in a regression analysis, fitting all the earlier terms, and then reforming the design matrix by replacing each column with the residuals from the corresponding regression. The projection matrix is then

```
X[i] *+ Ginverse(T(X[i] *+ X[i]) *+ T(X[i])
```

References

Gower, J.C. & Krzanowski, W.J. (1999) Analysis of distance for structured multivariate data and extensions to multivariate analysis of variance. *Applied Statistics*, **48**, 505-519.

Krzanowski, W.J. (2002) Multifactorial analysis of distance in studies of ecological community structure. *Journal of Agricultural, Biological and Ecological Statistics*, **7**, 222-232.

Payne, R.W. & Tobias, R.D. (1992). General balance, combination of information and the analysis of covariance. *Scandinavian Journal of Statistics*, **19**, 3-23.

See also

Directive: `PCO`.
Procedures: `MANOVA`, `RMULTIVARIATE`.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

MVARIOGRAM

Fits models to an experimental variogram (S.A. Harding, D.A. Murray & R. Webster).

Options

PRINT = *string tokens*	Controls printed output from the fit (model, summary, estimates, correlations, fittedvalues, monitoring); default mode, summ, esti
MODELTYPE = *string token*	Defines which model to fit (power, boundedlinear, circular, spherical, doublespherical, pentaspherical, exponential, besselk1, gaussian, affinepower, linear, cubic, stable, cardinalsine, matern); default powe
WEIGHTING = *string token*	Method to be used for weighting (counts, cbyvar, equal); default coun
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
SMOOTHNESS = *scalar*	Value of power parameter for the stable model, or ν parameter for the Matern model; default * i.e. estimate
ISOTROPY = *string token*	Defines whether to fit an isotropic or geometrical anisotropic model (isotropic, geometrical); default isot
WINDOW = *scalar*	Window in which to plot a graph; default 0 i.e. no graph
TITLE = *text*	Title for the graph
XUPPER = *scalar*	Upper limit for the *x*-axis in the graph
PENDATA = *scalar*	Pen to be used to plot the data; default 1
PENMODEL = *scalar*	Pen to be used to plot the model; default 2

Parameters

VARIOGRAM = *variates* or *matrices*	Experimental variogram to which the model is to be fitted, as a variate if in only one direction or as a matrix if there are several
COUNTS = *variates* or *matrices*	Counts for the points in each variogram (not required if WEIGHTING=equal)
DISTANCE = *variates* or *matrices*	Mean lag distances for the points in each variogram
DIRECTION = *variates*	Directions in which each variogram was computed
INITIAL = *scalars* or *variates*	Scalar defining initial distance parameter for an isotropic model, or variate with two values for a double-spherical isotropic model, or a variate with three values for a geometrical anisotropic model
ESTIMATES = *variates*	Estimated parameter values
FITTEDVALUES = *variates*	Fitted values
EXIT = *scalars*	Exit status from the nonlinear fitting
SAVE = *pointers*	Saves the model name and estimates in a pointer that can be used in KRIGE

Description

Procedure MVARIOGRAM uses the directives FIT, FITCURVE and FITNONLINEAR to fit various models to the experimental variogram. Models must be authorized in the sense that they cannot give rise to negative variances when data are combined. Technically they are conditionally negative semi-definite (CNSD); see Webster & Oliver (1990, 2001), Journel & Huijbregts (1978) or Chiles & Delfiner (1999) for an explanation.

The MODELTYPE option specifies the model that is to be fitted. There are bounded isotropic models with finite ranges. These all take the value $c + c_0$ for $h \geq a$, and the following values for $h < a$

boundedlinear	$c_0 + ch/a$
circular	$c_0 + c\ \{1 - (2/\pi)\arccos(h/a) + (2h/(\pi a))\sqrt{(1-h^2/a^2)}\}$
spherical	$c_0 + c\ \{1.5h/a - 0.5(h/a)^3\}$
doublespherical	$c_0 + c_1\ \{1.5h/a_1 - 0.5(h/a_1)^3\} + c_2\ \{1.5h/a_2 - 0.5(h/a_2)^3\}$ for $h \leq a_1$
	$c_0 + c_1 + c_2\ \{1.5h/a_2 - 0.5(h/a_2)^3\}$ for $a_1 < h < a_2$
	where $c = c_1 + c_2$
pentaspherical	$c_0 + c\ \{1.875h/a - 1.25(h/a)^3 + 0.375(h/a)^5\}$
cubic	$c_0 + c\ \{7(h/a)^2 - 8.75(h/a)^3 + 3.5(h/a)^5 - 0.75(h/a)^7\}$

There are also bounded asymptotic models

besselk1	$c_0 + c\ \{1 - h/a\ K_1(h/a)\}$ (Whittle's elementary correlation, Whittle 1954)
exponential	$c_0 + c\ \{1 - \exp(-h/a)\}$
gaussian	$c_0 + c\ \{1 - \exp(-h^2/a^2)\}$
stable	$c_0 + c\ \{1 - \exp(-(h/a)^b))\}$
matern	$c_0 + c\ \{1 - 1 / (2^{(\nu-1)}\ \Gamma(\nu))\ (h/a)^\nu\ K_\nu(h/a)\}$

unbounded models

power	$c_0 + g\ h^\alpha$ (power function with exponent α strictly between 0 and 2)
linear	$c_0 + c\ h$ which is a special case of the power function with exponent 1

and hole effect models

cardinalsine	$c_0 + c \times (1 - a/h \times \sin(h/a))$.

Geometrically anisotropic models, i.e. ones that might be made isotropic by a simple linear transformation of the spatial coordinates, can be fitted by setting option ISOTROPY=geometrical. The following transformation is used:

$$\Omega(\theta) = \sqrt{\{\ a^2\cos^2(\theta-\varphi) + b^2\sin^2(\theta-\varphi)\ \}}$$

where θ represents the direction (specified by the DIRECTION parameter) converted from degrees to radians. So, for example, a geometrical anisotropic power model would be

$$c_0 + (\ \Omega(\theta)\ h\ )^{\text{power}}$$

(Note: this particular model can also be defined by setting MODELTYPE=affinepower; the ISOTROPY option is then ignored.)

In all these models, the intercept term (or *nugget variance*) c_0 can be omitted by setting the CONSTANT option to omit; the default is estimate.

For the stable model the SMOOTHNESS option controls the power parameter for the model, and for the matern model it specifies the ν parameter. By default, the parameter is estimated. However, you can supply a value, to fix the parameter for the model fitting.

The data for the procedure can be taken directly from the FVARIOGRAM directive, with parameters DISTANCES, VARIOGRAMS and COUNTS corresponding to those with the same names in FVARIOGRAM. The data will be in variates if the variogram was calculated in only one direction. If it is in several, they can either be in matrices (as generated by FVARIOGRAM) or in variates. For ISOTROPY=geometrical, directions must be supplied, using the DIRECTIONS

parameter. These should be in a variate with one value for each column if the other data are in matrices; alternatively, they should be in a variate of the same length as the other variates.

The WEIGHTING option controls the weights that are used when fitting the model. The default setting counts uses the values supplied by the COUNTS parameter, cbyvar uses the COUNTS divided by the values in VARIOGRAM, and equal uses equal weights (of one).

By default, MVARIOGRAM generates rough starting values for the parameters. If the solution does not converge there are two likely reasons. The model may be unsuited for the particular experimental variogram. For example, a bounded model is specified when the variogram is clearly unbounded, or vice versa. You should choose only models that have approximately the right shape. Alternatively, the starting values may be too far from a sensible solution. You should then supply initial values using the INITIAL parameter. For a double-spherical isotropic model, INITIAL must be set to a variate with two values representing the two distance parameters. For the other isotropic models it should be set to a scalar defining the initial distance parameter. Finally, for a geometrical anisotropic model, it should be set to a variate with three values, defining the initial values for φ, the maximum distance parameter and the minimum distance parameter.

Printed output is controlled by the PRINT option, and includes all the usual settings as in FIT, FITCURVE or FITNONLINEAR. You can also produce a high-resolution graph of the data and the fitted model, by setting the WINDOW option to the number of a suitable window. By default WINDOW is zero, and no graph is produced. The TITLE option can supply a title for the plot. Option XUPPER can define an upper value for the *x*-axis (i.e. distance), and PENDATA and PENMODEL can supply the numbers of the pens to be used to plot the experimental variogram and the fitted model respectively (by default 1 and 2). Alternatively, you can use the ESTIMATES parameter to save the parameter estimates, and plot the variogram and model later with the DVARIOGRAM procedure.

The SAVE parameter saves the parameter estimates and associated information required by the KRIGE directive. The FITTEDVALUES parameter saves the fitted values, and the EXIT parameter saves the exit "status code" from FIT, FITCURVE or FITNONLINEAR. A zero value indicates success (see the *Guide to the Genstat Command Language*, Part 2, Section 3.7.4).

Options: PRINT, MODELTYPE, WEIGHTING, CONSTANT, SMOOTHNESS, ISOTROPY, WINDOW, TITLE, XUPPER, PENDATA, PENMODEL.

Parameters: VARIOGRAM, COUNTS, DISTANCE, DIRECTION, INITIAL, ESTIMATES, FITTEDVALUES, EXIT, SAVE.

Method

The model is fitted using directives FIT, FITCURVE or FITNONLINEAR as appropriate.

Action with RESTRICT

If the data variates are restricted, only the units not excluded by the restriction will be used.

References

Chiles, J-P. & Delfiner, P. (1999). *Geostatistics: Modeling Spatial Uncertainty*. Wiley, Chichester.

Journel, A.G. & Huijbregts, C.J. (1978). *Mining Geostatistics*. Academic Press, London.

Webster, R. & Oliver, M.A. (1990). *Statistical Methods in Soil and Land Resource Survey*. Oxford University Press.

Webster, R. & Oliver, M.A. (2001). *Geostatistics for Environmental Scientists*. Wiley, Chichester.

Whittle, P. (1954). On stationary processes in the plane. *Biometrika*, **41**, 434-449.

See also
Directives: FVARIOGRAM, MCOVARIOGRAM, KRIGE.
Procedure: DVARIOGRAM.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

MVFILL

Replaces missing values in a vector with the previous non-missing value in that vector (J.T.N.M. Thissen).

No options

Parameter

VECTORS = *vectors*	Variates, texts or factors whose missing values are replaced by the previous non-missing value of that vector

Description

A data matrix often has the values of a grouping variable in one column and values of quantitative measurements in other columns. However, one common method of data entry is to specify the value of the grouping variable only in the first row of each group. Procedure MVFILL replaces the missing values in a vector (variate, text or factor) with the previous non-missing value in that vector. If the first values of the VECTOR parameter are missing there is then no previous non-missing value, and these values remain missing.

Options: none.
Parameter: VECTORS.

Method

The procedure uses the data manipulation functions SHIFT, DIFFERENCE and NEXPAND.

Action with RESTRICT

Restrictions are ignored. This means that missing values are replaced in the same way as for unrestricted vectors. However, after the procedure the vectors are restricted in the same way as before.

See also

Directive: INTERPOLATE.
Procedures: MULTMISSING, SVHOTDECK.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

NCONVERT

Converts integers between base 10 and other bases (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (number); default numb
METHOD = *string token*	Whether to convert NUMBER to DIGITS or vice versa (tobase, frombase); default toba
BASE = *scalars*	Base to which to convert number; default 2

Parameters

NUMBER = *scalars*	Number in base 10
DIGITS = *pointers*	Digits of the NUMBER in the base specified by the BASE option
SIGN = *scalars*	Sign of the NUMBER

Description

NCONVERT can be used to convert an integer between the standard base 10 and another base, specified by the BASE option (default 2 i.e. binary). The number in base 10 is specified by the NUMBER parameter. In the other base it is represented in a pointer, specified by the DIGITS parameter, containing an integer for each of the digits required to represent it in that base. The SIGN parameter contains a scalar with the value +1 or −1 according to whether the number is positive or negative.

For example, the number 29 in base 10 would be represented in base 2 by a pointer containing five scalars with the values 1, 1, 1, 0 and 1. This results from the fact that

$$29 = 16 + 8 + 4 + 1$$

So there are 5 digits corresponding to the multipliers of 2^4, 2^3, 2^2, 2^1 and 2^0.

The PRINT option has a single setting, number, which prints the number in the two bases. By default this is printed, but you can suppress that by setting PRINT=*. The METHOD option controls the direction of the conversion: the default, tobase, converts from base 10 to the other base; conversion in the other direction is requested by the alternative setting, frombase.

Options: PRINT, METHOD, BASE.
Parameters: NUMBER, DIGITS, SIGN.

Method

The conversion is done by standard arithmetic using, for example, the MODULO function.

See also

Procedures: BPCONVERT, PRIMEPOWER.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

NCSPLINE

Calculates natural cubic spline basis functions for use e.g. in REML (S.J. Welham).

Options

INKNOTS = *variate*	Defines a set of knots to use to construct the spline
METHOD = *string token*	Whether to produce a basis suitable for use with independent or correlated random effects; (independent, correlated); default inde
ORTHOGONALIZETO = *variate*	Variate to use to get an orthogonalized basis; default * i.e. orthogonalization with respect to KNOTS

Parameters

X = *variates*	Values for which the basis functions are calculated
BASIS = *pointers*	Non-linear part of spline basis for use as design matrix for random effects in REML analysis
DBASIS = *pointers*	First derivative of BASIS functions
D2BASIS = *pointers*	Second derivative of BASIS functions
INVCOVARIANCE = *symmetric matrices*	Inverse covariance matrix for use with correlated spline random effects
SECONDDIFFERENCES = *matrices*	Scaled second divided difference matrix associated with KNOTS
KNOTS = *variates*	Knots used in construction of basis
DISTANCES = *variates*	Inter-knot distances used in construction of basis
SCALE = *scalars*	Saves the appropriate value for scaling design matrix

Description

This procedure generates the non-linear part of a basis for natural cubic splines with specified knots, evaluated at the variate X. The primary purpose of the procedure is to generate bases that can be used to specify and fit cubic smoothing splines within the mixed model framework (see Method section below). The explanatory covariate values at which the basis functions are to be calculated are specified in a variate using parameter X.

The INKNOTS option can be used to specify the knot points to use to construct the natural cubic spline basis. The set of distinct values found in this variate are then used as the knots. This option can be used to produce a low-rank approximation to the smoothing spline (but see Methods section below) which uses smaller matrices and hence is faster to fit. If INKNOTS is unset, the distinct values of X are used. The knots that are used can be saved using the parameter KNOTS. The METHOD option specifies whether the design matrix of basis functions is to be generated for independent random effects, or for correlated random effects. The ORTHOGONALIZETO option is used to specify a variate to be used in orthogonalization. The set of basis functions produced will then have mean zero and be orthogonal to the specified variate. Setting of this option to the data variate X is recommended, as then the fitted values corresponding to the fixed model will represent the whole of the linear trend in the fitted spline.

The basis that is generated can be saved in a pointer using the parameter BASIS. As the scale of this matrix is highly dependent on the inter-knot distances, it is recommended that scaling is used to keep the spline variance component within a reasonable range. If B is a matrix containing the basis values (as columns), then the recommended scaling provided is such that

```
TRACE(B *+ T(B)) = NROWS(B),
```

i.e. so that the average contribution of the spline random term to the variance of a unit is equal to the spline variance component. The SCALE parameter can be used to save the scalar value

```
c = SQRT(TRACE(B*+T(B))/NROWS(B))
```

so that the required scaling can be applied via the calculation

```
CALCULATE B[]=B[]/c
```

The first and second derivatives of the basis functions can be saved using parameters DBASIS and D2BASIS. Parameter INVCOVARIANCE can be used to save the inverse variance matrix for random effects when METHOD=correlated is used. (This corresponds to matrix *R* defined by Green & Silverman 1994.) Parameters SECONDDIFFERENCES and DISTANCES can save the divided second difference matrix and inter-knot distances used in construction of the basis. (These correspond to the matrix *Q* and vector h of Green & Silverman 1994.)

Options: INKNOTS, METHOD, ORTHOGONALIZETO.
Parameters: X, BASIS, DBASIS, D2BASIS, INVCOVARIANCE, SECONDDIFFERENCES, KNOTS, DISTANCES, SCALE.

Method

Within the mixed model framework, the natural cubic spline g(x) with *r* knots evaluated at variate x can be written as

$$g(x) = X\,\tau + B\,\delta$$

where X=[1 x] is a design matrix containing 2 linear basis functions, and B is a design matrix containing $r-2$ non-linear basis functions.

The cubic smoothing spline can be fitted as a natural cubic spline with knots at the distinct covariate values via a linear mixed model

$$y = X\,\tau + B\,\delta + \varepsilon$$

with

$$\mathrm{var}(\delta) = \lambda\,R^{-1},$$

where R is a banded symmetric matrix with $r-2$ rows defined by Green & Silverman (1994) and λ is a function of the smoothing parameter, which can also be estimated using REML. The matrix Z is the basis saved in a pointer using the BASIS parameter when METHOD=correlated is set. The mixed model cubic spline can be fitted using the following commands for data Y with covariate X:

```
NCSPLINE    [INKNOTS=X; METHOD=correlated; ORTHOGONAL=X]\
            X=X; BASIS=B; INVCOVARIANCE=R; SCALE=scB
CALCULATE   B[]=B[]/scB
VCOMPONENTS [FIXED=X] RANDOM=B
VSTRUCTURE  [TERM=B] FACTOR=B; MODEL=FIXED; INVERSE=R
REML        Y
```

It is more straightforward to transform the basis B to

$$Z = B\,R^{-0.5}$$

and fit the model

$$y = X\,\tau + Z\,u + \varepsilon$$

with

$$\mathrm{var}(u) = \lambda\,I,$$

i.e. a model with independent random effects. The transformed basis is obtained by using the default setting METHOD=independent, and the basis is saved by the using the BASIS parameter. The mixed model spline can then simply be fitted using the commands:

```
NCSPLINE    [INKNOTS=X; ORTHOG=X] X=X; BASIS=Z; SCALE=scZ
CALCULATE   Z[]=Z[]/scZ
VCOMPONENTS [FIXED=X] RANDOM=Z
REML        Y
```

See Verbyla *et al*. (1999) for further details of mixed model cubic smoothing splines.

This procedure can also be used to generate bases using a set of knots that differs from the set of distinct covariate values. This can be used as an approximation to reduce the computing load

associated with cubic splines with many knots, but the fitted spline no longer has the optimality properties associated with cubic smoothing splines. The goodness of the approximation depends on the number and position of knots chosen. Ruppert *et al*. (2003) recommend using

$$r = \min(n/4, 35)$$

and placing the r knots at the $i/(r+1)$ quantiles of the covariate for $i=1...r$.

Action with `RESTRICT`

The variates contained in `BASIS`, `DBASIS` and `D2BASIS` pointers are restricted in the same way as the `X` parameter. Values in the units excluded by the restriction are set to missing. If there is any restriction on the `KNOTS` option, knot values are calculated only from the included subset.

References

Green, P.J. & Silverman, B.W. (1994). *Non-parametric Regression and Generalised Linear Models*. London: Chapman & Hall.

Ruppert, D., Wand, M.P. & Carroll, R.J. (2003). *Semi-parametric Regression*. Cambridge: Cambridge University Press.

Verbyla, A.P., Cullis, B.R., Kenward, M.G. & Welham, S.J. (1999). The analysis of designed experiments and longitudinal data using smoothing splines (with discussion). *Applied Statistics*, 48, 269-311.

See also

Directive: `VCOMPONENTS`.

Procedures: `SPLINE`, `LSPLINE`, `PENSPLINE`, `PSPLINE`, `RADIALSPLINE`, `TENSORSPLINE`.

Function: `SSPLINE`.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

NLAR1

Fits curves with an AR1 or a power-distance correlation model (R.W. Payne).

Options

PRINT = *string tokens*	What to print (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, cparameter, cmonitoring, cplot); default mode, summ, esti, cpar
CURVE = *string token*	Which standard curve to fit (exponential, dexponential, cexponential, lexponential, logistic, glogistic, gompertz, ldl, qdl, qdq, fourier, dfourier, gaussian, dgaussian); default expo
SENSE = *string token*	Sense of a standard curve (right, left); default righ
ORIGIN = *scalars*	Constrained origin for a standard curve; default * i.e. not constrained
NONLINEAR = *string token*	How to treat nonlinear parameters between groups in standard curves (common, separate); default comm
CALCULATION = *expression structures*	Define a nonlinear model involving explanatory variates and nonlinear parameters; default * implies that a standard curve is fitted
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalars*	Limit for expansion of model terms; default 3
POOL = *string token*	Whether to pool ss in accumulated summary between all terms fitted in a linear model (yes, no); default no
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default *
FPROBABILITY = *string token*	Printing of probabilities for variance and deviance ratios (yes, no); default no
SELECTION = *string tokens*	Statistics to be displayed in the summary of analysis produced by PRINT=summary (%variance, %ss, adjustedr2, r2, seobservations, dispersion, %cv, %meandeviance, %deviance, aic, bic, sic); default %var, seob
SELINEAR = *string token*	Whether to calculate s.e.s for linear parameters when nonlinear parameters are also estimated (yes, no); default no
WEIGHTS = *variate*	Prior weights for the units
CPARAMETER = *scalars*	Correlation parameter
CPOSITIONS = *variate*	Correlation positions
CGROUPS = *factor*	Groupings of correlation positions
MAXCYCLE = *scalars*	Maximum number of iterations; default 100
TOLERANCE = *scalars*	Convergence criterion; default 10^{-5}

Parameter

TERMS = *formula*	Terms to be fitted

Description

NLAR1 allows you to fit curves and nonlinear models to data, such as repeated measurements, where the residuals may follow an AR1 or a power-distance correlation model. The CPOSITIONS option specifies the coordinates of the observations in the direction (e.g. time) along which the correlation model operates. You can also use the CGROUPS option to specify a factor to define groups of observations for the model – the correlation model is then defined only over the observations that belong to the same groups. If you are fitting a standard curve, CPOSITIONS will take the x-variate for the curve as its default, and the group factor (if specified e.g. to define parallel curves) as the default for CGROUPS. NLAR1 also allows the data units to have unequal weights, which can be supplied in a variate using the WEIGHTS option.

The parameter *phi* of the AR1 or power-distance model is estimated within NLAR1, and is assumed to be the same for every group. (Note that the model will be AR1 if the observations are each one unit apart within each group – the power-distance model is the natural extension of the AR1 model to unequally-spaced data; see Method.) You can save the estimated value of *phi*, in a scalar, using the CPARAMETER option.

Otherwise, NLAR1 is used much like FITCURVE or FIT (which are used inside NLAR1 to fit the model). NLAR1 must be preceded by a MODEL statement. You must also give an RCYCLE statement first if you want to fit a user-defined nonlinear model (using FIT), rather than a standard curve (using FITCURVE). The MODEL statement must have the WEIGHT option set to a symmetrix matrix, which need not have any values defined. NLAR1 will set the values according to the distances (CPOSITIONS), groups (CGROUPS) and estimated parameter *phi*. These values remain set after NLAR1. So you can display or save further output using RCHECK, RDISPLAY, RGRAPH or RKEEP, in the usual way. You could also, for example, use NLAR1 to fit a full set of regression terms, and then use DROP to investigate smaller models while still using the *phi* estimate from the full model. NLAR1 has a TERMS parameter to specify the terms to be fitted, like the parameter of FIT and FITCURVE. It also has options CURVE, SENSE, ORIGIN, NONLINEAR, CALCULATION, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, SELECTION and SELINEAR which operate like those of FITCURVE and FIT. If the CALCULATION option is unset, then options CURVE, SENSE, ORIGIN, NONLINEAR define which standard curve to fit (using FITCURVE). Alternatively, if CALCULATION is set, those options are ignored, and the expressions specified by CALCULATION define a nonlinear model to be fitted (by FIT).

The PRINT option is also similar, except that it has three additional settings:

cparameter	prints the estimated value of the correlation *phi*, together with a test for *phi*=0,
cmonitoring	provides monitoring information for the estimation of *phi*,
cplot	plots the likelihood for *phi*.

Note, the likelihood values omit some constant terms that depend only on the regression terms. The default is PRINT=model,summary,estimates,cparameter.

The other options control the estimation. The MAXCYCLE option defines the maximum number of iterations (default 100) used to estimate *phi*, and the TOLERANCE option specifies the convergence criterion i.e. the accurary to which *phi* is to be estimated (default 10^{-5}).

Options: PRINT, CURVE, SENSE, ORIGIN, NONLINEAR, CALCULATION, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, SELECTION, SELINEAR, WEIGHTS, CMETHOD, CPARAMETER, CPOSITIONS, CGROUPS, MAXCYCLE, TOLERANCE.

Parameter: TERMS.

Method

To estimate *phi* NLAR1 uses procedure MIN1DIMENSION, which calls a procedure _MIN1DFUNCTION, which is loaded automatically with NLAR1. _MIN1DFUNCTION uses the FITCURVE or FIT directives to fit the regression model for a particular value of *phi*, and then evaluates the likelihood. If standard curves are fitted using FITCURVE for groups of observations, these groups must be independent. Otherwise FITCURVE will give a fault diagnostic. (Thus the default setting for the CGROUPS option with standard curves is the group factor, if one has been specified in the TERMS formula.)

The total degrees of freedom for the regression are decreased by one, to take account of the estimation of the correlation parameter *phi*, by setting a variable in the regression save structure (rsave[1][3]$[47]) to one.

Action with RESTRICT

Restrictions are not allowed.

See also

Directives: FITCURVE, FITNONLINEAR, VSTRUCTURE.

Procedure: RAR1.

Genstat Reference Manual 1 Summary sections on: Repeated measurements, Regression analysis.

NLCONTRASTS

Fits nonlinear contrasts to quantitative factors in ANOVA (R.C. Butler).

Options

PRINT = *string tokens*	Printed output required (aovtable, information, covariates, effects, residuals, contrasts, means, %cv, missingvalues); default aovt, info, cova, mean, miss
CURVE = *string token*	Curve (as in FITCURVE) to use for nonlinear regression (exponential, dexponential, cexponential, lexponential, logistic, glogistic, gompertz, ldl, qdl, qdq); default expo
FPROBABILITY = *string token*	Printing of probabilities for variance ratios (yes, no); default no
PSE = *string token*	Standard errors to print with means tables (differences, means); default diff
WEIGHT = *variate*	Variate of weights for each unit; default * (no weights)

Parameters

Y = *variates*	Data to be analysed
XFACTOR = *factors*	Factor with quantitative levels for which contrasts are to be found
XLEVELS = *variates*	Variate of values to use for the levels of XFACTOR; if unset, the factor levels themselves are used
GROUPFACTOR = *factors*	Factor whose interaction with XFACTOR is to be assessed
CONTRASTS = *pointers*	Structures to hold the estimates of the fitted contrasts: CONTRASTS[1] is a pointer with two values, labelled 'Curve' (parameter estimates for a single fitted curve) and 'Deviations' (the differences between this curve and the means for XFACTOR); CONTRASTS[2] has three values, labelled 'Common NonLin' (parameter estimates for curves fitted with common nonlinear parameters for all levels of GROUPFACTOR), 'Separate Curves' (parameter estimates for curves fitted with all parameters varying with the levels of GROUPFACTOR) and 'Deviations' (differences between the treatment means and the Separate Curves); the order of the parameters is as in the output of the procedure, the variates of estimated contrasts are labelled by the parameter names as used in the printed output, while the 'Deviations' are both tables, labelled by the relevant factors
SECONTRASTS = *pointers*	Structures to save the standard errors for the contrast estimates, including 'deviations'; the pointer has the same form as the CONTRASTS pointer
DFCONTRASTS = *pointers*	Structures to save the degrees of freedom for the contrast estimates; the pointer has the same form as the CONTRASTS pointer, except that the variates and tables are replaced by scalars

Description

The ANOVA directive allows linear contrasts to be fitted and incorporated into the analysis-of-variance table. NLCONTRASTS extends this to enable nonlinear contrasts to be fitted to the effects of a quantitative factor and its interaction with another factor. The analysis should include both main effects and the interaction between the factors. The procedure will work for any block structure providing each treatment term is estimated entirely within one stratum. The result is similar to ANOVA with a polynomial contrast, but with slightly different partitions of the treatment sums of squares. The main effect is partitioned into the sum of squares for the "Curve" and the remainder or "Deviations". The interaction sum of squares is partitioned into the sum of squares due to curves with "Common Nonlinear" parameters for the levels of the non-quantitative factor, and the extra sum of squares due to having "Separate Curves" for each level of that factor, and the remaining sum of squares which again represents "Deviations".

The BLOCKSTRUCTURE and TREATMENTSTRUCTURE directives must be used in the normal way before the procedure is called, and any COVARIATES should also be defined first. The structure of the analysis-of-variance table is then accessed from inside the procedure. The Y parameter defines the variate to be analysed, and the form of nonlinear contrast is defined using the CURVE option of the procedure. The same choices of curves are available as for FITCURVE. There are four other options, PRINT, FPROBABILITY, PSE, and WEIGHT, which are exactly as for ANOVA. The XFACTOR parameter defines the factor to which the contrasts are to be fitted, and the XLEVELS parameter may be used to define *x* values for the regressions if the levels already defined for the factor are unsuitable. The GROUPFACTOR parameter defines the factor whose interaction with XFACTOR is to be assessed. The final three parameters CONTRASTS, SECONTRASTS and DFCONTRASTS can be used to save the parameter estimates for the contrasts, their standard errors and degrees of freedom respectively.

Options: PRINT, CURVE, FPROBABILITY, PSE, WEIGHT.

Parameters: Y, XFACTOR, XLEVELS, GROUPFACTOR, CONTRASTS, SECONTRASTS, DFCONTRASTS.

Method

ANOVA is used to obtain the basic analysis-of-variance table and the sums of squares for the treatment terms. FITCURVE is then used with the treatment means to fit three sets of curves: a single curve, curves with common nonlinear parameters, and entirely separate curves. The deviances and degrees of freedom obtained from these are used in conjunction with the treatment sums of squares to calculate the contrast sums of squares and degrees of freedom. Further details are given by Butler & Brain (1992). New lines for the analysis-of-variance table are then constructed using PRINT and EDIT, and these lines are then inserted into the table (saved in a text with ADISPLAY) using EDIT. The standard errors for the parameter estimates and deviances are based on the Residual Mean Square for the appropriate stratum. Standard errors for deviations are calculated using the method in the *Guide to the Genstat Command Language*, Part 2, Section 4.5.

Action with RESTRICT

If the Y variate is restricted, the procedure will use only the units not excluded by the restriction.

Reference

Butler, R.C. & Brain, P. (1993). Nonlinear Contrasts in ANOVA. *Genstat Newsletter*, **29**, 20-27.

See also
Directives: ANOVA, FITCURVE, FITNONLINEAR.
Procedures: NLAR1, RPARALLEL.
Genstat Reference Manual 1 Summary sections on: Analysis of variance, Regression analysis.

NORMTEST

Performs tests of univariate and/or multivariate Normality (M.S. Ridout).

Option

`PRINT` = *string tokens*	Allows the required printed output to be selected: test statistics, tables of critical values and the flagging of significant values with stars (`marginal`, `bivariateangle`, `radius`, `critical`, `stars`); default `marg`, `biva`, `radi`

Parameter

`DATA` = *variates* or *pointers*	Variates whose univariate Normality is to be tested or pointers, each to a set of variates whose Normality and/or multivariate Normality are to be tested

Description

This procedure offers three types of test of Normality.

Marginal (univariate) tests – assess the Normality of each variate in turn. The variates are standardized to have mean=0, variance=1 and then transformed with the `NORMAL` function. The test is based on the idea that, assuming Normality, these transformed values should look like a sample from a uniform distribution on (0,1).

Bivariate angle tests – assess the bivariate Normality of each pair of variates in turn. The variates are standardized so that they are uncorrelated and have mean=0 and variance=1. The test is based on the following idea: if x and y are the standardized values, then the angle between the x-axis and the line joining (0,0) to (x,y) should, assuming Normality, be uniformly distributed on $(0,2\pi)$.

Radius test – provides a single overall test of multivariate Normality. The variates are again standardized to have mean=0 and so that their covariance matrix is the identity matrix. The test uses the fact that if $z_1, z_2, ..., z_n$ are the standardized values then $z_1^2 + z_2^2 + ... + z_n^2$ should, under multivariate Normality, be approximately distributed as chi-square on n degrees of freedom.

For each type of test, the test statistics are empirical distribution function (EDF) statistics – i.e. they compare the empirical distribution function of the sample with the theoretical distribution expected under the null hypothesis. Three EDF statistics are provided for each type of test – the Anderson-Darling statistic, the Cramer-von Mises statistic and the Watson statistic. The idea is to provide good power against a wide range of alternatives. The test statistics are adjusted so that their null distribution is independent of the sample size; critical values can be printed by the procedure (option `PRINT=critical`).

The `DATA` parameter is used to indicate the variate(s) whose Normality is to be assessed. If a single variate is supplied, its Normality is tested using the marginal test. Alternatively, `DATA` can supply a pointer to a set of variates to be tested for multivariate Normality.

The `PRINT` option can be used to select the type of test using the settings `marginal`, `bivariateangle` and `radius`. The setting `critical` allows tables of critical values to be printed, and `stars` requests that significant values of the test statistics be flagged with stars. Settings `bivariateangle` and `radius` are relevant only when testing for multivariate Normality. By default `PRINT=marginal,bivariateangle,radius`

Option: `PRINT`.
Parameter: `DATA`.

Method
The calculations are clearly set out in Aitchison (1986; Section 7.3). Bivariate angle and radius tests are described by Andrews, Gnanadesikan & Warner (1973). Stephens (1974) describes the EDF statistics used and gives tables of critical values and information on their comparative power.

Action with `RESTRICT`
If a variate to which the `DATA` parameter is set is restricted, the tests will be calculated using only the units included by the restriction. Similarly, the variates in a `DATA` pointer can be restricted, but then must all be restricted in the same way. The procedure does not work properly with missing values. If missing values are present, `RESTRICT` should be used (before calling the procedure) to exclude all units for which any of the variates has a missing value.

References
Aitchison J.A. (1986). *The Statistical Analysis of Compositional Data*. London: Chapman & Hall.

Andrews D.F., Gnanadesikan R. & Warner J.L. (1973). Methods for assessing multivariate normality. In: *Multivariate Analysis III* (ed. P.R. Krishnaiah) 95-116. New York: Academic Press.

Stephens M.A. (1974). EDF statistics for goodness of fit and some comparisons. *Journal of the American Statistical Association*, **69**, 730-737.

See also
Directive: `DISTRIBUTION`.
Procedures: `EDFTEST`, `WSTATISTIC`.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

NOTICE

Provides news and other information about Genstat (R.W. Payne).

Option

PRINT = *string tokens*	Indicates what information is required (news, release, errors, instructions); default rele

No parameters

Description

NOTICE allows you to access news and information about Genstat. The PRINT option specifies what information is required, using the following settings:

news	Genstat news,
release	information about changes in the current release,
errors	how to report errors, and
instructions	instructions for authors of library procedures.

Option: PRINT. Parameters: none.

See also

Directive: HELP.
Procedure: LIBHELP.

OPLS

Performs orthogonal partial least squares regression (V. M. Cave).

Options

PRINT = *string tokens*	Printed output required (data, xloadings, yloadings, ploadings, scores, leverages, xerrors, yerrors, scree, xpercent, ypercent, predictions, groups, estimates, fittedvalues, summary); default esti, xper, yper, scor, xloa, yloa, ploa, summ
PCPRINT = *string tokens*	Controls printed output from principal components analysis of orthogonal X matrix (loadings, roots, scores, tests); default root
PLOT = *string token*	What graphs to plot (pcplot); default * (i.e. none)
NORTHOGONALROOTS = *scalar*	Number of orthogonal components to extract; default 1
NROOTS = *scalar*	Number of predictive (i.e. PLS) components to extract; default 1
STANDARDIZE = *string tokens*	Whether to standardize the Y, X and filtered X variables to unit variance and zero mean (Y, X, filteredX); default * (i.e. no standardizing)
NGROUPS = *scalar*	Number of cross-validation groups used by PLS; default 1 (i.e. no cross-validation performed)
SEED = *scalar* or *factor*	A scalar indicating the seed value used for dividing the data randomly into NGROUPS groups for cross-validation by PLS, or a factor indicating a specific set of groupings to use for cross-validation by PLS; default 0
LABELS = *text*	Sample labels for X and Y to use in output; default uses the integers 1...*n* where *n* is the length of the variates in X and Y
PLABELS = *text*	Labels for XPREDICTIONS; default uses P1, P2 etc.
PCMETHOD = *string tokens*	Method used by PCP to perform principal components analysis on the orthogonal X matrix (ssp, correlation, vcovariance, variancecovariance); default * (i.e. principal components analysis not performed)
WINDOW = *scalar*	Window to use for graph (available only when NORTHOGONALROOTS = 1); default 3

Parameters

Y = *pointers*	Pointer to variates containing the dependent variable(s) for each analysis
X = *pointers*	Pointer to variates containing the independent variables for each analysis
YLOADINGS = *pointers*	Pointer to variates containing the Y component loadings, for the predictive (i.e. PLS) dimensions, extracted from the filtered X matrix
XLOADINGS = *pointers*	Pointer to variates containing the component loading weights for the predictive dimensions, extracted from the filtered X matrix
PLOADINGS = *pointers*	Pointer to variates containing the bilinear model loadings for the predictive dimensions, extracted from

	the filtered X matrix
YSCORES = *pointers*	Pointer to variates containing the Y component scores, for each predictive dimension extracted from the filtered X matrix
XSCORES =*pointers*	Pointer to variates containing the component scores for each predictive dimension, extracted from the filtered X matrix
B = *diagonal matrices*	Saves the regression coefficients of YSCORES on XSCORES, for the predictive dimensions, extracted from the filtered X matrix
YPREDICTIONS = *pointers*	Pointer to variates used to store predicted y-values for samples in the prediction set
XPREDICTIONS = *pointers*	Pointer to variates containing data for the independent variables in the prediction set
ESTIMATES = *matrices*	An n_X+1 by n_Y matrix (where n_X and n_Y are the number of variates contained in X and Y, respectively) to store the PLS regression coefficients
FITTEDVALUES = *pointers*	Pointer to variates used to store the fitted values for the Y variates
LEVERAGES = *variates*	Variate to store the leverage that each sample has on the PLS model
PRESS = *variates*	Variate used to store the Predictive Residual Error Sum of Squares for each dimension in the PLS model, available only if cross-validation has been selected
RSS = *variates*	Variate to save residual sums of squares
YRESIDUALS = *pointers*	Pointer to variates containing the residuals from the Y block after NROOTS predictive dimensions have been extracted, uncorrected for any scaling applied using STANDARDIZE
XRESIDUALS = *pointers*	Pointer to variates containing the residuals from the X block after NROOTS predictive dimensions have been extracted, uncorrected for any scaling applied using STANDARDIZE
PCSCORES = *matrices*	Matrix to save principal component scores
PCSAVE = *pointers*	Pointer to save structures from the principal component analysis (by PCP) of the orthogonal X matrix
SAVE = *pointers*	Pointer to save structures from the orthogonal projection

Description

OPLS performs orthogonal partial least squares (O-PLS) regression.

Variation in X that is orthogonal (i.e. uncorrelated) to Y may disturb PLS modelling, complicating the model interpretation. O-PLS combines PLS with a pre-processing step that filters out systematic variation in X, orthogonal to Y, that disturbs the PLS model. To improve model interpretation, the variation explained by each regular PLS component is partitioned into two parts:

1) variation linearly related to Y (i.e. predictive) and
2) variation orthogonal to Y.

The resulting O-PLS model takes the form:

$$X = TP^{\mathrm{T}} + T_{\mathrm{ortho}}P_{\mathrm{ortho}}{}^{\mathrm{T}} + E$$
$$Y = TC^{\mathrm{T}} + F$$

where $T = XW$ and $T_{\mathrm{ortho}} = XW_{\mathrm{ortho}}$. The predictive variation in X is modelled by the matrices T, W and P, whose columns contain the predictive component scores, loading weights and loadings,

respectively. The orthogonal variation is modelled by analogous matrices T_{ortho}, W_{ortho} and P_{ortho}, whose columns contain the orthogonal component scores, loading weights and loadings, respectively. The columns of matrix *C* contain *Y*-loadings, and *E* and *F* are the residual matrices.

The number of predictive components used to model the predictive variation is specified by the NROOTS option; default 1. The number of orthogonal components used to model the orthogonal variation is specified by the NORTHOGONALROOTS option; default 1. The OPLS procedure also enables the orthogonal variation to be further explored, through principal components analysis.

In practice, the OPLS procedure removes *Y*-orthogonal variation from *X* to form a filtered *X* matrix (*Xfiltered*). A PLS model is then fitted to *Xfiltered*, using the PLS procedure.

The dependent and independent variates are supplied using the Y and X parameters, respectively, as pointers containing a variate for each dimension. The Y and X variates must not contain missing values. A pointer of variates containing new *X* data, for which predictions are desired, can be specified by the XPREDICTIONS parameter. Sample labels for X and XPREDICTIONS can be provided by using the LABELS and PLABELS options, respectively.

The STANDARDIZE option controls whether the *Y*, *X* and the filtered *X* variables are standardized to mean zero and unit variance prior to analysis. The *Y* variables are standardized prior to orthogonal projection and PLS analysis, the *X* variables are standardized prior to orthogonal projection, and the filtered *X* variables are standardized prior to modelling by PLS. By default, none of these are standardized. Note, however, that all variables are automatically centred prior to the PLS analysis, even if no standardization is requested.

The SAVE parameter can supply a pointer to store structures from orthogonal projection. The labels of the pointer, and their corresponding information, are as follows:

w_ortho	orthogonal component loading weights,
t_ortho	orthogonal component scores,
p_ortho	orthogonal loadings,
X_filtered	filtered *X* matrix, with the orthogonal variation removed,
X_ortho	matrix containing the orthogonal variation,
Xpred_filtered	filtered prediction *X* matrix, with the orthogonal variation removed,
Xpred_ortho	matrix containing the orthogonal variation of the prediction *X* matrix.

The NGROUPS and SEED options control cross-validation by the PLS procedure. The parameters YLOADINGS, XLOADINGS, PLOADINGS, YSCORES, XSCORES, B, YPREDICTIONS, ESTIMATES, FITTEDVALUES, LEVERAGES, PRESS, RSS, YRESIDUALS and XRESIDUALS allow output from the PLS procedure to be saved (i.e. from modelling the predictive variation).

Printed output is controlled by the PRINT option. Almost all of the settings are the same as those of the PLS procedure, and are used in exactly the same way. However, there is an additional setting, summary, which summarizes the percentage of variation in X explained by each orthogonal and predictive (i.e. PLS) component.

You can set the PCMETHOD option to request a principal component analysis to decompose the matrix of orthogonal variation (see X_ortho above), and to specify the method to use. Its settings are the same as those of the METHOD option of the PCP directive. Printed output is controlled by the PCPRINT option, which operates exactly as the PRINT option of the PCP directive. The PCSAVE parameter can supply a pointer to store details from the analysis. You can set option PLOT = pcplot to produce a score plot; by default, no plot is produced. When NORTHOGONALROOTS = 1, the WINDOW option can be used to control the window to used for the plot; default 3.

Options: PRINT, PCPRINT, PLOT, NORTHOGONALROOTS, NROOTS, STANDARDIZE, NGROUPS, SEED, LABELS, PLABELS, PCMETHOD, WINDOW.

Parameters: Y, X, YLOADINGS, XLOADINGS, PLOADINGS, YSCORES, XSCORES, B, YPREDICTIONS, XPREDICTIONS, ESTIMATES, FITTEDVALUES, LEVERAGES, PRESS, RSS, YRESIDUALS, XRESIDUALS, PCSCORES, PCSAVE, SAVE.

Method

OPLS uses the methodology of Trygg & Wold (2002), applying the algorithm described in Biagioni *et al.* (2011), to remove variation from *X* that is not correlated to *Y*. OPLS then calls the PLS procedure to fit a PLS model to the filtered (i.e. pre-treated) matrix with the orthogonal variation removed.

To perform the principal components analysis on the matrix of orthogonal variation, OPLS uses the PCP directive, taking the setting for its METHOD option from the PCMETHOD option, and the setting for its NROOTS option from the NORTHOGONALROOTS option. When there is only one root, the score plot, which can be requested by setting option PLOT = pcplot, is produced by the DOTHISTOGRAM procedure. When there are several roots, it is produced by the DMSCATTER procedure. If the XPREDICTIONS parameter is set, principal component scores for the samples in the prediction set are estimated as described by Trygg & Wold (2002), and plotted in red.

Action with RESTRICT

OPLS will work with restricted variates, fitting an O-PLS model to the subset of objects formed by the restriction. The subset can be defined by restricting any of the X or Y variates. However, if more than one variate is restricted, they must be restricted in the same way. Note that the unrestricted length of all of the data variates must be the same, and the number of samples in the restricted subset must be at least three. Any restrictions on a text supplied for the LABELS option, or on a factor for the SEED option, are ignored.

When restricted data are supplied, and LABELS are also given, the appropriate subset of labels will appear in the output; if LABELS are not defined, then default labels reflecting the position in the restricted data are used.

No restrictions are allowed on the variates supplied by the XPREDICTIONS parameter, or on the text supplied by the PLABELS option.

References

Biagioni, D.J., Astling, D.P., Graf, P. & Davis, M.F. (2011). Orthogonal projects to latent structures solutions properties for chemometrics and systems biology. *Journal of Chemometrics*, **25**, 514-525.

Trygg, J. & Wold, S. (2002). Orthogonal projections to latent structures (O-PLS). *Journal of Chemometrics*, **16**, 119-128.

See also

Directives: PCP, SVD.

Procedure: PLS.

Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Regression analysis.

ORTHPOLYNOMIAL

Calculates orthogonal polynomials (P.W. Lane).

Options

MAXDEGREE = *scalar*	Maximum degree of polynomial to be calculated; default is the number of identifiers in the pointer specified by the POLYNOMIAL parameter
WEIGHTS = *variate*	Weights to be used in orthogonalization; default * gives an equal weight to each unit

Parameters

X = *variates*	Values from which to calculate the polynomials; no default – this parameter must be set
POLYNOMIAL = *pointers*	Identifiers of variates to store results; no default – this parameter must be set

Description

Polynomials of low degree can be fitted by ordinary linear regression, estimating effects of terms X, X**2, X**3, and so on for a variate X. However, it is sometimes preferable to arrange that successive polynomial terms are orthogonal to each other; certainly, there are likely to be numerical problems with polynomials of degree five or more, if they are not orthogonal. ORTHPOLYNOMIAL calculates orthogonal polynomials up to a specified maximum degree from a given variate. The orthogonalization can be weighted by specifying a variate of weights.

Options: MAXDEGREE, WEIGHTS. Parameters: X, POLYNOMIAL.

Method

Successive formation of polynomials, starting with $p_1 = x - \text{mean}(x)$, ensuring orthogonality of p_i with $p_1 \ldots p_{i-1}$; that is:

$$\sum (\textit{weight} \times p_i \times p_j) = 0$$

Action with RESTRICT

A variate in the X parameter can be restricted: the restriction is transferred to the calculated polynomials, and to the weight variate if specified.

See also

Procedure: VORTHPOLYNOMIAL.
Functions: POL, POLND, REG.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

PAIRTEST

Performs t-tests for pairwise differences (P.W. Goedhart).

Options

PRINT = *string tokens*	What to print (differences, sed, tvalues, tprobabilities); default diff, sed, tval
DF = *scalar* or *symmetric matrix*	Degrees of freedom for calculation of TPROBABILITIES from TVALUES; default 10000, approximates to the normal distribution
SORT = *string token*	Whether ESTIMATES (and other output) are sorted in ascending order (yes, no); default no

Parameters

ESTIMATES = *variates*	Estimates to be compared
VCOVARIANCE = *symmetric matrices*	Symmetric matrix containing the variance-covariance matrix of the estimates
LABELS = *texts*	Text vector naming the elements of ESTIMATES; if unset, the numbers 1, 2... are used as labels
DIFFERENCES = *symmetric matrices*	To save the pairwise differences (ESTIMATES on the diagonal)
SED = *symmetric matrices*	To save the standard errors of the pairwise differences (missing values on the diagonal)
TVALUES = *symmetric matrices*	To save the t-values (missing values on the diagonal)
TPROBABILITIES = *symmetric matrices*	To save the t-probabilities (missing values on the diagonal)

Description

PAIRTEST can be used to test all pairwise differences in every situation in which a vector of estimates and a corresponding variance-covariance matrix are available. PAIRTEST is particularly useful for tests of all pairwise differences of slopes after fitting a model with an interaction between a factor and a variate. In most other situations procedure RPAIR will be more suitable.

All pairwise differences of entries in ESTIMATES with variance-covariance matrix VCOVARIANCE are calculated and tested. The results of these tests can be saved in symmetric matrices DIFFERENCES, SED, TVALUES and TPROBABILITIES. The matrices are labeled by text vector LABELS or, if LABELS is unset, by the numbers 1, 2, 3...

PRINT controls the output of PAIRTEST. The t-probabilities are based on DF degrees of freedom; by default, if DF has not been set, Normal probablitities are calculated. Option SORT controls whether the estimates on the diagonal of DIFFERENCES are sorted in ascending order. The other output is sorted accordingly.

Options: PRINT, DF, SORT.

Parameters: ESTIMATES, VCOVARIANCE, LABELS, DIFFERENCES, SED, TVALUES, TPROBABILITIES.

Method

The calculations are all relatively straightforward.

Action with RESTRICT
The variate ESTIMATES and the text LABELS can be restricted; the analysis is restricted according to restrictions on ESTIMATES. The lengths of the unrestricted vectors ESTIMATES and LABELS must be identical.

See also
Procedures: ALLDIFFERENCES, AMCOMPARISON, AUMCOMPARISON, PPAIR, RPAIR.
Genstat Reference Manual 1 Summary section on: Regression analysis.

PARTIALCORRELATIONS

Calculates partial correlations for a list of variates (S. Langton).

Options

PRINT = *string token*	Output required (correlations); default corr
CORRELATIONS = *symmetric matrix*	Saves the partial correlations
WEIGHTS = *variate*	Supplies weights for the units; default * i.e. all 1

Parameters

DATA = *variates*	Set of variates whose partial correlations are to be calculated

Description

PARTIALCORRELATIONS calculates a symmetric matrix of partial correlations from a set of variates. The matrix contains the correlation between each pair of variates after adjusting for all the other variates in the set. The variates are listed using the DATA parameter, and the matrix can be saved using the CORRELATIONS option. A variate of weights can be supplied using the WEIGHTS option. The PRINT option controls the printing of the partial correlations. The default setting correlations ensures that they are printed, but you can set PRINT=* to suppress printing.

Options: PRINT, CORRELATIONS, WEIGHTS.
Parameter: DATA.

Method

The partial correlations are calculated from the correlations C by the calculation

```
-CORRMAT( INVERSE(C) )
```

Action with RESTRICT

Any units that are restricted within the DATA variates (or which have missing values) are excluded from the analysis.

See also

Directive: CORRELATE.
Procedures: DCORRELATION, FCORRELATION, PRCORRELATION, SCORRELATION.

PCOPROCRUSTES

Performs a multiple Procrustes analysis (P.G.N. Digby).

Options

PROTATE = *string tokens*	Printed output required from each Procrustes rotation (rotations, coordinates, residuals, sums); default * i.e. no output
PPCO = *string tokens*	Printed output required from the PCO analysis (roots, scores, centroid); default root, score, cent
SCALING = *string token*	Whether isotropic scaling should be used for the Procrustes rotations (no, yes); default no
STANDARDIZE = *string tokens*	Whether to centre the configurations and/or normalize them to unit sums-of-squares for the Procrustes rotations (centre, normalize); default cent, norm

Parameters

DATA = *pointers*	Each pointer points to a set of matrices holding the original input configurations
LRV = *LRVs*	Stores the latent vectors (i.e. coordinates), roots and trace from the PCO analysis
CENTROID = *diagonal matrices*	Stores the squared distances of the points representing the input configurations from their overall centroid from the PCO analysis
DISTANCES = *symmetric matrices*	Stores the residual sums-of-squares from the Procrustes rotations

Description

An $N \times V$ matrix represents a configuration of points, for each of N units, in V dimensions. Given a set of M such matrices, a multiple Procrustes analysis compares them in pairs, keeping the residual sums-of-squares, and performs a principal coordinate analysis of the residual sums-of-squares to obtain an ordination representing the individual configurations. The rows of the matrices must represent the same set of units, in the same order; however there is no need for them to have the same number of columns (although generally they will do). An example of the use of multiple Procrustes analysis is given by Digby & Kempton (1987, pages 121-3).

The configurations of points are specified using the DATA parameter. This supplies a pointer containing a matrix with the data for each configuration. The PROTATE option controls the output from the individual Procrustes rotations, and the PPCO option controls that from the principal coordinate analysis. There are $M \times (M-1)/2$ Procrustes rotations so, by default, PROTATE=* to suppress any output. The SCALING and STANDARDIZE options control the way in which the Procrustes rotations are carried out, using the SCALING and STANDARDIZE options of ROTATE. However, the combination of SCALING=yes and STANDARDIZE=centre should not be used, because then the results will be dependent on the order of the input matrices.

The LRV and CENTROID parameters can be used to save results from the principal coordinates analysis, and the DISTANCES parameter can be used to save the symmetric matrix of the residual sums-of-squares from the Procrustes analyses.

Options: PROTATE, PPCO, SCALING, STANDARDIZE.
Parameters: DATA, LRV, CENTROID, DISTANCES.

Method
The pairwise Procrustes rotations are performed using the ROTATE directive, and the residual sums-of-squares are stored in a symmetric matrix of order *M*. This matrix is then used as input to a principal coordinate analysis, performed using the PCO directive on a suitably transformed copy of the matrix.

Reference
Digby, P.G.N. & Kempton, R.A (1987). *Multivariate Analysis of Ecological Communities*. Chapman & Hall, London.

See also
Directives: ROTATE. FACROTATE.
Procedure: GENPROCRUSTES.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

PDESIGN

Prints or stores treatment combinations tabulated by the block factors (R.W. Payne).

Options

PRINT = *string token*	Controls the printing of the design (design); default desi
BLOCKSTRUCTURE = *formula*	Defines the block factors for the design; the default is to take those specified by the BLOCKSTRUCTURE directive
TREATMENTSTRUCTURE = *formula*	Defines the treatment factors for each design; the default is to take those specified by the TREATMENTSTRUCTURE directive
TABLES = *pointer*	Contains tables to store the tabulated factor values for printing outside the procedure in some other format
FREPRESENTATION = *string token*	How to represent the factor values (labels, levels); default leve

No parameters

Description

PDESIGN allows the treatment combinations allocated to each plot in a design to be displayed as tables, classified by the block factors.

By default, the combinations are represented using the levels of the treatment factors. If any factor also has labels these are printed alongside the levels, as a key, after the tables. The levels are printed in formats that are determined automatically in a way that avoids wasted space or unnecessary decimal places. Alternatively, if you set option FREPRESENTATION=labels, the labels are displayed in the table, instead of the levels.

The block factors are obtained from the block structure of the design, which can be specified explicitly using the BLOCKSTRUCTURE option; otherwise PDESIGN will use any structure that has already been defined by a BLOCKSTRUCTURE statement earlier in the job. Similarly, the treatment factors are obtained either from the TREATMENTSTRUCTURE option of the procedure, or from an earlier TREATMENTSTRUCTURE statement.

If the display produced by the procedure is unsuitable, printing can be suppressed by setting option PRINT=* (by default PRINT=design), and the tables of treatment levels can be saved for printing outside the procedure by setting the TABLES option to a pointer. This will be returned with an element for each treatment factor, pointing to a table classified by the block factors and storing the tabulated levels of the treatment.

Options: PRINT, BLOCKSTRUCTURE, TREATMENTSTRUCTURE, TABLES, FREPRESENTATION.
Parameters: none.

Method

The FCLASSIFICATION directive is used to form lists of factors from the block or treatment formulae and, if the block factors do not supply a unique combination of levels for every unit of the design, procedure AFUNITS is used to form a factor to index the units with each combination. Each treatment factor is then copied into a variate and TABULATE is used to put the values into a table classified by the block factors. Numbers of decimal places for printing the factor levels are determined using the DECIMALS procedure. When FREPRESENTATION=labels, the TLABELS parameter of PRINT is used to display the labels within the table.

Action with RESTRICT

If any of the factors is restricted, only the part of the design not excluded by the restriction will

be displayed.

See also
Procedures: ADSPREADSHEET, DDESIGN.
Genstat Reference Manual 1 Summary section on: Design of experiments.

PDUPLICATE

Duplicates a pointer, with all its components (R.W. Payne).

No options

Parameters

OLDPOINTER = *pointers*	Pointers to duplicate
NEWPOINTER = *pointers*	Duplicated pointers

Description

PDUPLICATE is useful when you want to duplicate the complete tree of data structures to which a pointer points. So, it duplicates not only the pointer itself, but all the structures to which it points. Also, if any of these structures is itself a pointer, the structures to which that too points will be duplicated.

The pointer to be duplicated is specified by the OLDPOINTER parameter, and the duplicated pointer is saved by the NEWPOINTER parameter.

Options: none.
Parameters: OLDPOINTER, NEWPOINTER.

Method

PDUPLICATE calls itself recursively to duplicate any pointers inside OLDPOINTER. The individual data structures are duplicated by the DUPLICATE directive.

See also

Directives: DUPLICATE, POINTER.
Genstat Reference Manual 1 Summary section on: Data structures.

PEAKFINDER

Finds the locations of peaks in an observed series (D.B. Baird).

Options

PRINT = *string token*	Controls printed output (peaks); default peak
CURVE = *string token*	Shape of curve to fit to peaks (normal, exponential); default norm
PLOT = *string tokens*	What to plot (peaks, trace); default peak
METHOD = *string token*	The method for finding the peaks (additive, local); default addi
BANDWIDTH = *scalar*	Width of window to use when fitting peaks locally, or the number of low points at the edge of each zone when fitting peaks additively; default takes the number of points divided by ten, or six if this is greater
MINPEAK = *scalar*	Minimum height of a peak; no default (must be set)
MINGAP = *scalar*	Minimum number of points between two peaks when METHOD=additive; default 5
MINFALL = *scalar*	Minimum fall around a peak before a new peak will be found when METHOD=additive; default MINPEAK/10
MINCOHERENCY = *scalar*	Minimum coherency (i.e. proportion of variation explained) for a peak to be selected when METHOD=local; default 0.1
MAXSIGMA = *scalar*	The maximum value of sigma for peaks when METHOD=local; default 4*BANDWIDTH
MAXRESIDUAL = *scalar*	Limit on the absolute size of any residual for the adding of peaks to stop when METHOD=additive; default MINPEAK/3
WINDOW = *scalar*	Window number for the plots; default 3
SCREEN = *string token*	Whether to clear the screen before plotting or continue plotting on the old screen (clear, keep); default clea

Parameters

Y = *variates*	Series to search for peaks
X = *variates*	X-coordinates for the series; default !(1...n) where n is the number of Y values
YPEAKS = *variates*	Saves the y-values of the peaks
XPEAKS = *variates*	Saves the positions of the peaks
FITTEDYPEAKS = *variates*	Saves the heights of the peaks predicted by the fitted models
SIGMA = *variates*	Saves the sigma values of the fitted Normal or exponential models, which provide a measure of the widths of the peaks
COHERENCY = *variates*	Saves the coherency (i.e. the proportion of variation accounted for) of the model fitted to identify each peak model
TITLE = *texts*	Titles for the plots

Description

PEAKFINDER looks for peaks in a series of observations supplied, in a variate, by the Y parameter. The X parameter can supply a variate specifying x-values for the series; if this is not set, these are assumed to be the integers 1...*n*, where *n* is the number of values in Y.

The peaks are found by fitting curves to the y-values, as specified by setting of the CURVE option:

normal	fits a Normal curves EXP(-0.5*((X-p)/sigma)**2)
exponential	fits exponential curves EXP(-ABS(X-p)/sigma)

where p is the location of the peak, and sigma is a measure of its width.

The METHOD option controls how the peaks are fitted. With the default setting, additive, PEAKFINDER looks to see whether the series can be divided into separate zones. The criterion is that these must be separated by $2\times b$ y-values of size less than $m/2$, where the value b is defined by the BANDWIDTH option, and m is defined by the MINPEAK option. MINPEAK must be set, while BANDWIDTH has a default of $n/10$, or 6 if n is less than 60. Then, in each zone, PEAKFINDER starts by fitting a single curve. If the maximum absolute residual from that fit is greater than the value specified by the MAXRESIDUAL option, it adds another curve. (So the model for Y in that zone is now the sum of two curves.) If the maximum absolute residual from the model is still greater than MAXRESIDUAL, it adds another curve. This continues until either the residuals are all less than MAXRESIDUAL, or the model contains ten curves. The success of the procedure depends on the value of MAXRESIDUAL. The default value is MINPEAK divided by three. Smaller values allow more complicated patterns of peaks to be identified, but may slow the procedure down and cause convergence problems. Two very close peaks can be generated with this method, when the shape of the peak does not follow that specified by the CURVE option. A second additive component at a close location but with a different value of sigma may then be added to provide a better fit to the shape of the peak. The MINGAP and MINFALL options attempt to control this behaviour, and ensure that only a single peak is given. The MINGAP option sets a lower limit on the number of points between any two peaks (default 5), and the MINFALL option sets a lower limit on the fall in y-values between peaks (default MINPEAK/10).

With the alternative setting, METHOD=local, PEAKFINDER fits the specified curve locally around each x-value in turn. The size of the local window for the fit is defined by the BANDWIDTH option, and can be sensitive to the value that is chosen. So this may need to be varied to tune the peak finding process. Ideally it should be equal to the width of the anticipated peaks. The MAXSIGMA option sets an upper limit on the value of sigma for a curve if the corresponding peak is to be accepted (default 4*BANDWIDTH), and the MINCOHERENCY option sets a limit on its coherency i.e. the proportion of variation of Y that the curve accounts for (default 0.1). Increasing MINCOHERENCY requires the peaks to conform more closely to the chosen shape, while increasing MAXSIGMA allows broader and flatter peaks to be found. This method will find only one peak in any area, unless there is valley or flat area of size at least BANDWIDTH between the peaks.

You can set option PRINT=peaks to print the peak locations, the corresponding y-values, their fitted heights, sigma values and coherency. These can also be saved using the XPEAKS, YPEAKS, FITTEDYPEAKS, SIGMA and COHERENCY parameters.

The PLOT option controls the graphs that are displayed, with settings:

peaks	to plot the fitted peaks, with a horizontal blue line showing the minimum peak height,
trace	to plot the components of the fitted model.

The WINDOW option specifies the window to use for the plots (default 3). The SCREEN option controls whether or not to clear the screen first (default clear). Note, however, that SCREEN is not used with PLOT=trace. You can supply a title for the plots using the TITLE parameter.

Options: PRINT, CURVE, PLOT, METHOD, BANDWIDTH, MINPEAK, MINGAP, MINFALL, MINCOHERENCY, MAXSIGMA, MAXRESIDUAL, WINDOW, SCREEN.

Parameters: Y, X, YPEAKS, XPEAKS, FITTEDYPEAKS, SIGMA, COHERENCY, TITLE.

Action with `RESTRICT`
Any restrictions on the `Y` variate are ignored.

See also
Procedures: `ALIGNCURVE`, `BASELINE`.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

PENSPLINE

Calculates design matrices to fit a penalized spline as a linear mixed model (S.J. Welham).

Options

KMETHOD = *string token*	Method for constructing the set of knots (equal, quantile, given); default equa
NSEGMENTS = *scalar*	Specifies the number of segments between boundaries; default * obtains a value automatically
INKNOTS = *variate*	Provides the set of knots when KMETHOD=given
DEGREE = *scalar*	Degree of polynomial used to form the underlying spline basis functions; default 1
LOWER = *scalar*	Specifies the lower boundary when KMETHOD=equal; default takes the minimum value in X
UPPER = *scalar*	Specifies the upper boundary when KMETHOD=equal; default takes the maximum value in X
ORTHOGONALIZETO = *variate*	Variate to use to get an orthogonalized basis; default * i.e. orthogonalization with respect to X
SCALING = *scalar*	Scaling of the XRANDOM terms (automatic, none); default auto

Parameters

X = *variates*	The explanatory variate for which the spline values are required
XFIXED = *matrices*	Saves the design matrix to define the fixed terms (excluding the constant) for fitting the penalized spline
XRANDOM = *matrices*	Saves the design matrix to define the random terms for fitting the penalized spline
KNOTS = *variates*	Saves the internal knots and boundaries used to form the basis for the spline
PX = *variates*	Specifies x-values at which predictions are required
PFIXED = *matrices*	Saves the design matrix for the fixed terms (excluding the constant) for the spline at the prediction points
PRANDOM = *matrices*	Saves the design matrix for the random terms for the spline at the prediction points

Description

This procedure generates the fixed and random terms required to fit a penalized spline (Ruppert, Wand & Carroll 2003) as a linear mixed model, using REML estimation of the smoothing parameter. The explanatory variate values at which the spline is to be calculated are specified, in a variate, by the X parameter.

The KMETHOD option specifies how to choose the set of knots for the penalized spline, using settings:

equal	splits the range of X into segments of equal length (default),
quantiles	defines the set of knots in terms of equally-spaced quantiles of X,
given	indicates that the knots will be supplied, in a variate, by the INKNOTS option.

The number of segments or quantiles for the equal and quantile settings is specified by the NSEGMENTS option. If this is unset, the number is determined automatically as

min([*p*/4], 35) + 1

(Ruppert 2002) where p is the number of unique values of the variate X and [r] denotes the integer part of the number r. The lower and upper boundaries for equal segments are specified by the LOWER and UPPER options, respectively, taking the minimum and maximum values of X as their defaults. The set of knots used to form the spline basis can be saved using the KNOTS parameter.

The DEGREE option specifies the degree of polynomial that is used to form the underlying spline basis functions. The default, DEGREE=1, gives a linear penalized spline.

The ORTHOGONALIZETO option specifies a variate to use in orthogonalization. The set of random spline terms will then be orthogonal to the fixed terms when evaluated at the specified values. For most data sets, it is recommended to set ORTHOGONALIZETO to the variate X (the default). The random terms will then be orthogonal to the fixed terms, and fitted values corresponding to the fixed model will represent the whole of the polynomial trend in the fitted spline. For very large data sets, this calculation can be onerous and can be approximated by making the two bases orthogonal at the knots. No orthogonalization is carried out if ORTHOGONALIZETO is set to a scalar value (e.g. ORTHOGONALIZETO=0).

The penalized spline terms are saved as two matrices. The terms required to be fitted as fixed terms can be saved using the XFIXED parameter. This matrix does not include the constant term as this is added by default as part of a mixed model. The terms to be fitted as random can be saved using the XRANDOM parameter.

The random terms can be scaled so that, for a random spline matrix Z,

```
TRACE(Z *+ T(Z)) = NROWS(Z)
```

This ensures that the average contribution of Z to the variance of an observation is equal to one, and hence the overall contribution from the term is equal to the spline variance component. This removes possible computational instabilities, and improves interpretability of the spline variance component. This scaling is imposed by default, but can be avoided by setting option SCALING=none.

The penalized spline terms required for prediction can be saved using the PXFIXED and PXRANDOM parameters. The PX parameter defines the set of x-values at which the predictions are to be made.

Options: KMETHOD, NSEGMENTS, INKNOTS, DEGREE, LOWER, UPPER, ORTHOGONALIZETO, SCALING.
Parameters: X, XFIXED, XRANDOM, KNOTS, PX, PXFIXED, PXRANDOM.

Method

The penalized spline of degree k and r knots, evaluated on variate X, minimizes the penalized sum of squares

$$(y - X\tau - Z u)' R^{-1} (y - X\tau - Z u) + \lambda u' u$$

where

- X is a design matrix containing k basis functions $x^{\{0...k\}}$, with associated unknown parameters τ;
- Z is a design matrix containing r truncated power basis (TPF) functions with associated unknown effects u.

The TPF function of degree k at knot t_j takes the form $(X-t_j)_+^k$, where

$x_+ = x$ for $x > 0$
$= 0$ otherwise.

This penalized sum of squares is reformulated as the estimating equations from a mixed model of the form

$$y = X\tau + Z u + e$$

where

- u is a set of r independently and identically distributed Normal random effects with variance

$\sigma_s^2 I$

e is a vector of residual errors with variance $\sigma^2 R$.

Fitting this mixed model with known λ set equal to σ^2/σ_s^2 produces estimates that minimize the penalized sum of squares. In addition, we can estimate the smoothing parameter using `REML` via the variance component σ_s^2. This can be generalized straightforwardly to mixed models with additional fixed and random terms.

The implementation in this procedure allows the random design matrix to be orthogonalized with respect to the fixed design matrix at a given variate. For orthogonalization with respect to the variate `X`, this is achieved by using random design matrix

$Z^* = (\mathrm{I} - X(X'X)^{-1}X')\mathrm{Z}$

The entirety of the polynomial trend is then captured by the fixed model. Orthogonalization with respect to a variate `t` is calculated as

$Z^* = Z - X(T'T)^{-1}T'\ P(\mathtt{t})$

where T is a matrix holding $\mathtt{t}^{\{0\ldots k\}}$ and $P(\mathtt{t})$ is the appropriate TPF basis evaluated at `t`.

When the random matrix is scaled so that trace($Z^*Z^{*\prime}$) is equal to the number of row of Z^*, the average contribution of the spline term to the variance of each unit ($\sigma_s^2 \times$ diag($Z^*Z^{*\prime}$)) is equal to σ_s^2. This makes the spline variance component value directly comparable with the residual variance.

Note that the constant function is not included in the fixed design matrix generated by `PENSPLINE`, as this term is added automatically to the linear mixed model by the default option setting, `CONSTANT=estimate`, in the `VCOMPONENTS` statement.

The design matrices for use in prediction are calculated by evaluating the same set of basis functions at the predict points.

Action with `RESTRICT`

Input structures must not be restricted.

References

Ruppert, D. (2002). Selecting the number of knots for penalised splines. *Computational & Graphical Statistics*, **11**, 735-757.

Ruppert, D., Wand, M.P. & Carroll, R.J. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.

See also

Directive: `VCOMPONENTS`.

Procedures: `SPLINE`, `LSPLINE`, `NCSPLINE`, `PSPLINE`, `RADIALSPLINE`, `TENSORSPLINE`.

Function: `SSPLINE`.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

PERCENT

Expresses the body of a table as percentages of one of its margins (R.W. Payne).

Options

CLASSIFICATION = *factors*	Factors classifying the margin over which the percentages are to be calculated; if this is not set, the percentages are over the final margin (grand mean or grand total etc.)
METHOD = *string token*	Method to use to calculate the margin if not already present (totals, means, minima, maxima, variances, medians); default tota
HUNDRED = *string token*	Whether to put 100% values into the margin instead of the original values (no, yes); default no

Parameters

OLDTABLE = *tables*	Tables containing the original values
NEWTABLE = *tables*	Tables to store the percentage values; if any of these is unset, the new values replace those in the original table

Description

PERCENT allows you to express the body of a table as percentages of the values in one of its margins. The table is specified using the OLDTABLE parameter. A table to store the new values can be specified using the NEWTABLE parameter, otherwise these replace the values of the original table. The margin is indicated by listing the factors that define it using the CLASSIFICATION option; the default is the final margin (the grand total, or grand mean etc). If the original table has no margins, option METHOD defines how these are to be calculated; the default is to form margins of totals. The values originally in the margin will be left unchanged. If you would prefer these to be replaced by values of 100%, you should set option HUNDRED=yes.

Options: CLASSIFICATION, METHOD, HUNDRED. Parameters: OLDTABLE, NEWTABLE.

Method

If the OLDTABLE has no margins and contains no missing values, these are formed by the MARGIN directive. Alternatively, if there are missing values, margins other than variances can be formed using TABULATE. CALCULATE is then used to put the required margin into a table classified just by the factors that define the margin. The original table is divided by the marginal table and multiplied by 100 to give the required percentages. If option HUNDRED=no, the same operations are done on a dummy table that originally contains random numbers; for this table, values of 100 should occur only in the margin. Thus by using a logical test in which the values of the dummy table are compared with 100, the marginal values of the original table can be put back into the margin of the final table. The random numbers are generated using a specially written procedure URANDOM in case the Genstat random number generator is already in use in the program that called PERCENT.

See also

Directives: COMBINE, TABLE, TABULATE, MARGIN.
Procedures: MTABULATE, SVSTRATIFIED, SVTABULATE, TABMODE, TABSORT, T%CONTROL.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

PERIODTEST

Gives periodogram-based tests for white noise in time series (R.P. Littlejohn).

Option

LENGTH = *scalar* or *variate*	Scalar specifying that the first N units of the series are to be used, or a variate specifying the first and last units of the series to be used

Parameters

SERIES = *variates*	Specify the time series to be analysed
PERIODOGRAM = *variates*	Save periodograms of the time series

Description

PERIODTEST gives periodogram-based tests for departure from white noise in a set of time series. The series are supplied in a list of variates, using the SERIES parameter. The LENGTH option can specify that only part of each series is to be used, using either a scalar N to indicate that the first N values are to be used, or a variate of length two, holding the values of the first and last units of the required subseries. This may be used to eliminate missing values, which are otherwise not permitted.

The mean-adjusted periodogram is calculated for each series using FOURIER, and can be saved using the PERIODOGRAM parameter. The maximum periodogram ordinate test, Fisher's g-test and the Kolmogorov-Smirnov test on the cumulative periodogram are calculated using the standard formulae (Priestley 1981).

The output for each series consists of the value of the maximum periodogram ordinate (after scaling by the length of the analysed series), the frequency at which this maximum occurs (expressed as the unit number in the PERIODOGRAM variate, i.e. if the maximum occurs at $\omega = 2\pi j/N$, then j is given), and the probability of exceeding this maximum; the ratio of the maximum to the total of the periodogram ordinates (Fisher's g), and the probability of exceeding this; and the Kolmogorov-Smirnov D statistic based on the maximum deviation of the cumulative periodogram from the line $y=x$.

Option: LENGTH. Parameters: SERIES, PERIODOGRAM.

Method

The series are mean-corrected, but not trend corrected, before transformation.

Action with RESTRICT

The SERIES may not be restricted; restriction of the input series to a contiguous set of units may be achieved by use of the LENGTH parameter.

Reference

Priestley, M.B. (1981). *Spectral Analysis and Time Series*. Academic Press, London.

See also

Directive: FOURIER.
Procedures: DFOURIER, MCROSSPECTRUM, REPPERIODOGRAM, SMOOTHSPECTRUM.
Genstat Reference Manual 1 Summary section on: Time series.

PERMUTE

Forms all possible permutations of the integers 1...*n* (J.W. McNicol & R.W. Payne).

Option

SORT = *string token*	Whether or not to sort the permutations (no, yes); default no

Parameters

NVALUES = *scalars*	Specifies the final number, *n*, in the sequence of integers 1...*n* to be permuted
PERMUTATIONS = *pointers*	Pointer to a set of variates of length NVALUES storing the permutations

Description

PERMUTE forms all the permutations of the integers 1 up to the value *n* specified by the NVALUES parameter. The permutations are saved, as a set of variates each of length NVALUES, in a pointer supplied by the PERMUTATIONS parameter. By default, the permutations will occur in an arbitrary order, but option SORT can be set to yes to sort them into the standard (lexicographic) order.

Option: SORT.
Parameters: NVALUES, PERMUTATIONS.

Method

The procedure uses the standard Genstat manipulation directives, CALCULATE, EQUATE etc.

See also

Directive: SETALLOCATIONS.
Procedures: APERMTEST, CHIPERMTEST, RPERMTEST.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

PFACLEVELS

Prints levels and labels of factors (R.W. Payne).

No options

Parameter

FACTOR = *factors*	Factors whose levels and labels are to be printed

Description

PFACLEVELS prints the levels, and the labels (if present), of factors. This can be useful, for example, to check that they have been defined correctly, before either reading or printing a large data set. If you have already defined the values of the factors, the output also includes the replication of each level.

You specify the factors, whose levels and labels are to be printed, by the FACTOR parameter.

Options: none.
Parameter: FACTOR.

See also

Directive: FACTOR.
Genstat Reference Manual 1 Summary section on: Input and output.

PLINK

Prints a link to a graphics file into an HTML file (D.A. Murray).

Options

CHANNEL = *scalar*	Output channel number of file; default current output channel
EXCLUDEPATH = *string token*	Whether to remove path information when printing the link (yes, no); default no

Parameter

FILENAME = *texts*	Name of the graphics file to be linked within the html file

Description

PLINK can be used to insert a link within an html file to display a graphics file such as a Portable Network Graphics (PNG) or JPEG file. The link is written within the output file using the html <img /> tag. For example, the html code for a graph called CapeWagtail.jpg would be written as:

```
<img src="CapeWagtail.jpg" border="0"><BR>
```

The name of the graphics file to appear in the link is supplied using the FILENAME parameter.

The CHANNEL option determines where the output appears. By default, the output is printed to the current output channel, but CHANNEL can be set to a scalar to send it to another output channel. This output channel must have been opened in HTML style. The correspondence between channels and files on the computer is explained in the description of the OPEN directive.

The full details of the file name, including any path information, are written within the link by default. You can exclude path information from the link by setting option EXCLUDEPATH=yes.

Options: CHANNEL, EXCLUDEPATH.
Parameter: FILENAME.

Method

The link is written using PRINT with the typesetting command ~h{}, to ensure the correct style of output within the HTML file.

See also

Directive: OPEN.
Genstat Reference Manual 1 Summary section on: Input and output.

PLS

Fits a partial least squares regression model (Ian Wakeling & Nick Bratchell).

Options

PRINT = *string tokens*	Printed output required (data, xloadings, yloadings, ploadings, scores, leverages, xerrors, yerrors, scree, xpercent, ypercent, predictions, groups, estimates, fittedvalues); default esti, xper, yper, scor, xloa, yloa, ploa
NROOTS = *scalar*	Number of PLS dimensions to be extracted
YSCALING = *string token*	Whether to scale the Y variates to unit variance; (yes, no); default no
XSCALING = *string token*	Whether to scale the X variates to unit variance; (yes, no); default no
NGROUPS = *scalar*	Number of cross-validation groups into which to divide the data; default 1 (i.e. no cross-validation performed)
SEED = *scalar* or *factor*	A scalar indicating the seed value to use when dividing the data randomly into NGROUPS groups for the cross-validation or a factor to indicate a specific set of groupings to use for the cross-validation; default 0
LABELS = *text*	Sample labels for X and Y that are to be used in the printed output; defaults to the integers 1...*n* where *n* is the length of the variates in X and Y
PLABELS = *text*	Sample labels for XPREDICTIONS that are to be used in the printed output; default uses the integers 1, 2 ...

Parameters

Y = *pointers*	Pointer to variates containing the dependent variables
X = *pointers*	Pointer to variates containing the independent variables
YLOADINGS = *pointers*	Pointer to variates used to store the *Y* component loadings for each dimension extracted
XLOADINGS = *pointers*	Pointer to variates used to store the *X* component loadings for each dimension extracted
PLOADINGS = *pointers*	Pointer to variates used to store the loadings for the bilinear model for the *X* block
YSCORES = *pointers*	Pointer to variates used to store the *Y* component scores for each dimension extracted
XSCORES = *pointers*	Pointer to variates used to store the *X* component scores for each dimension extracted
B = *matrices*	A diagonal matrix containing the regression coefficients of YSCORES on XSCORES for each dimension
YPREDICTIONS = *pointers*	A pointer to variates used to store predicted *Y* values for samples in the prediction set
XPREDICTIONS = *pointers*	A pointer to variates containing data for the independent variables in the prediction set
ESTIMATES = *matrices*	An n_X+1 by n_Y matrix (where n_X and n_Y are the numbers of variates contained in X and Y respectively) used to store the PLS regression coefficients for a PLS model with NROOTS dimensions
FITTEDVALUES = *pointers*	Pointer to variates used to store the fitted values for each Y variate

LEVERAGES = *variates*	Variate used to store the leverage that each sample has on the PLS model
PRESS = *variates*	Variate used to contain the Predictive Residual Error Sum of Squares for each dimension in the PLS model, available only if cross-validation has been selected
RSS = *variates*	Variate used to store the Residual Sum of Squares for each dimension extracted
YRESIDUALS = *pointers*	Pointer to variates used to store the residuals from the *Y* block after NROOTS dimensions have been extracted, uncorrected for any scaling applied using YSCALING
XRESIDUALS = *pointers*	Pointer to variates used to store the residuals from the *X* block after NROOTS dimensions have been extracted, uncorrected for any scaling applied using XSCALING
XPRESIDUALS = *pointers*	Pointer to variates used to store the residuals from the XPREDICTIONS block after NROOTS dimensions have been extracted
†FTEST = *pointers*	Pointer to save the results from the Osten F test (when NGROUPS > 1)

Description

The regression method of Partial Least Squares (PLS) was initially developed as a calibration method for use with chemical data. It was designed principally for use with overdetermined data sets and to be more efficient computationally than competing methods such as principal components regression. If *Y* and *X* denote matrices of dependent and independent variables respectively, then the aim of PLS is to fit a bilinear model having the form $T=XW$, $X=TP'+E$ and $Y=TQ'+F$, where *W* is a matrix of coefficients whose columns define the PLS factors as linear combinations of the independent variables. Successive PLS factors contained in the columns of *T* are selected both to minimise the residuals in *E* and simultaneously to have high squared covariance with a single *Y* variate (PLS1) or a linear combination of multiple *Y* variates (PLS2). The columns of *T* are constrained to be mutually orthogonal. See Helland (1988) or Hoskuldsson (1988) for a more comprehensive description of the PLS method.

The procedure allows the calculation of PLS1 and PLS2 models with cross-validation to assist in the determination of the correct number of dimensions to include in the model. By setting the NGROUPS option the data are randomly divided into a number of groups; samples in each group are then modelled from the remaining samples only. The sum of squares of differences between these "leave out predictions" and the observed values of *Y* are called PRESS. Many tests of significance for determining the correct number of dimensions are based on comparing values of PRESS for PLS models of varying rank. Values of PRESS are used in the procedure to perform Osten's (1988) test of significance and may also be plotted out in a scree diagram. In addition to the factor scores, factor loadings and residuals, the procedure also calculates a leverage measure (Martens & Naes 1989 page 276) and a single linear combination of the *X* variables (ESTIMATES) which summarises the entire PLS model.

The procedure will fail if there are missing values present in either the X or Y variates.

To use a PLS model to make predictions from new observations on the *X* variables, two methods are available. Either the user may do this manually by using the model as specified in the estimates matrix, or the new *X* data may be specified beforehand as the pointer to variates XPREDICTIONS and the corresponding predictions obtained as YPREDICTIONS.

Output from the PLS procedure can be selected using the following settings of the PRINT option.

data	the unscaled data values (with labels).
xloadings	*X*-component loadings (columns of the matrix *W* – see

above).

`yloadings` variable loadings for the bilinear model of the matrix of dependent variables. Note that these are standardized to unit length and are not the same as the columns of the matrix *Q* above. To obtain *Q*, form the matrix *C*, whose columns are the standardized loadings, and post-multiply by the diagonal matrix supplied as the output parameter *B*.

`ploadings` variable loadings for the bilinear model of the matrix of independent variables (columns of the matrix P – see above).

`scores` *X* and *Y* component scores. The *X* component scores are the columns of the matrix *T* and are mutually orthogonal. The *Y* component scores, usually given the symbol *u*, are not in fact needed in the calculation of the PLS model unless an iterative algorithm is used (see method section). They are provided here for completeness, as sometimes it is useful to plot the *Y* component scores against the *X* component scores to give a visual indication of the degree of fit for each PLS dimension.

`leverages` measure of leverage.

`xerrors` residual sum of squares and residual standard deviations for all the independent variables. When `NGROUPS`>1 additional statistics are calculated from the cross-validated residuals, derived when each object is left out. The PRESS value is equal to the sum of squares of cross-validated standard deviations for each *X* variable multipled by $N-1$, where *N* is the total number of observations. The cross-validated standard deviations may therefore be used to measure the predictive ability of the model for each of the variables.

`yerrors` residual sum of squares and residual standard deviations for all the dependent variables (see xerrors above).

`scree` scree diagram of PRESS.

`xpercent` percentage variance explained for the *X* variables.

`ypercent` percentage variance explained for the *Y* variables.

`predictions` predicted values for any observations that were not included in the PLS model but were supplied using the `XPREDICTIONS` parameter.

`groups` details of groupings used for cross-validation.

`estimates` estimated PLS regression coefficients.

`fittedvalues` fitted values from the PLS regressions.

The default settings are `estimates`, `xpercent`, `ypercent`, `scores`, `xloadings`, `yloadings`, `ploadings`.

The data for `PLS` are supplied using the `X` and `Y` parameters, as pointers to variates containing the columns of the *X* and *Y* matrices. Other parameters allow output to be saved in appropriate data structures.

Options: `PRINT`, `NROOTS`, `YSCALING`, `XSCALING`, `NGROUPS`, `SEED`, `LABELS`, `PLABELS`.

Parameters: `Y`, `X`, `YLOADINGS`, `XLOADINGS`, `PLOADINGS`, `YSCORES`, `XSCORES`, `B`, `YPREDICTIONS`, `XPREDICTIONS`, `ESTIMATES`, `FITTEDVALUES`, `LEVERAGES`, `PRESS`, `RSS`, `YRESIDUALS`, `XRESIDUALS`, `XPRESIDUALS`, `FTEST`.

Method

Although the PLS method is often presented in terms of an iterative algorithm (Manne 1987), the X block loadings vector for the first PLS dimension (w_1) is simply the eigenvector of $X'YY'X$ corresponding to its largest eigenvalue. To find the second and subsequent dimensions, X and Y are deflated by orthogonalising with respect to the current PLS factor (t=Xw) and the eigenanalysis repeated. The above approach was adopted by Rogers (1987) in an implementation of a Genstat 4 macro. Here we adopt a very similar approach by performing a singular value decomposition on the matrix $X'Y$ which simultaneously obtains loading vectors for both data blocks (Hoskuldsson 1988, de Jong & ter Braak 1994).

It is usual to centre all variables prior to a PLS analysis, the procedure will automatically do so even if the XSCALING/YSCALING options are not set. On exit from the procedure the variates pointed to by X and Y are unchanged.

Action with RESTRICT

The procedure will work with restricted variates, fitting a PLS model to the subset of objects indicated by the restriction. If there are different restrictions on different data variates then these restrictions will be combined and the analysis performed on the subset of samples that is common to all the restrictions. Note that the unrestricted length of all of the data variates must be the same and the number of samples in the common subset must be at least three. Any restrictions on a text supplied for the LABELS option or a factor for the SEED option will be ignored. On exit from the procedure all the data variates, and if supplied the SEED factor and LABELS text, will all be returned restricted to the common subset of samples. Output data structures that correspond to the samples (i.e. XSCORES, YSCORES, FITTEDVALUES, LEVERAGES, YRESIDUAL and XRESIDUAL) will also be returned restricted to the common subset, and missing values will be used for those values that have been restricted out.

When restricted data are supplied and LABELS are also given then the appropriate subset of labels will be appear in the output; if LABELS are not defined then default labels reflecting the position of the restricted data in the unrestricted variate will be used instead.

No restrictions are allowed in the variates supplied in the XPREDICTIONS parameter or the PLABELS option.

References

Helland, I.S. (1988). On the structure of partial least squares regression. *Commun, Statist.-.Simula.Comput.*, **17**, 581-607.

Hoskuldsson, A. (1988). PLS Regression Methods. *J. Chemometrics*, **2**, 211-228.

de Jong & ter Braak (1994). Comments on the PLS kernel algorithm. *J. Chemometrics*, **8**, 169-174

Manne, R. (1987). Analysis of two partial least squares algorithms for multivariate calibration. *Chemometrics and Intell. Lab. Systems*, **2**, 187-197.

Naes, T. & Martens H. (1989). *Multivariate Calibrarion*. John Wiley, Chichester.

Osten, D.W. (1988). Selection of optimal regression models via cross-validation. *J. Chemometrics*, **2**, 39-48.

Rogers, C.A. (1987). A Genstat Macro for Partial Least Squares Analysis with Cross-Validation Assessment of Model Dimensionality. *Genstat Newsletter*, **18**, 81-92.

See also

Procedures: CCA, OPLS, RDA, RIDGE.

Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Regression analysis.

PNTEST

Calculates one- and two-sample Poisson tests (D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (test, summary, confidence); default test, summ, conf
METHOD = *string token*	Type of test required (twosided, greaterthan, lessthan); default twos
TEST = *string token*	Form of the test for one-sample test (exact, normalapproximation); default norm
S1 = *scalar*	Sample size for sample 1; default 1
S2 = *scalar*	Sample size for sample 2; default 1
CIPROBABILITY = *scalar*	The probability level for the confidence interval; default 0.95
NULL = *scalar*	The value of the probability of success under the null hypothesis for the one-sample test

Parameters

MU1 = *scalars* or *variates*	Numbers recorded in the first sample
MU2 = *scalars* or *variates*	Numbers recorded in the second sample
NORMAL = *scalars*	Saves the Normal approximation
PROBABILITY = *scalars*	Saves the probability value from the one-sample or two-sample tests
LOWER = *scalars*	Saves the lower limit of the confidence interval
UPPER = *scalars*	Saves the upper limit of the confidence interval

Description

PNTEST calculates one- and two-sample Poisson tests. The value for the mean under the null hypothesis for a one-sample test is specified by the option NULL. You can supply the sample mean m_1 as a scalar using the MU1 parameter. The sample size is then specified by the S1 option (with default 1). Alternatively, you can set MU1 to a variate containing the counts in the individual samples (and the sample size is then the number of non-missing values that it contains). With a two-sample test, parameters MU1 and MU2 similarly provide the means (m_1 and m_2) for samples 1 and 2 respectively, and the sample sizes can be specified using the S1 and S2 options.

For both one- and two-sample cases, the test is assumed to be two-sided unless otherwise requested by the METHOD option. Setting METHOD=greaterthan will give a one-sided test of the null hypothesis that $m_1 > m_2$ or NULL (for a two-sample or one-sample test, respectively). Similarly, METHOD=lessthan will produce a test of the null hypothesis $m_1 < m_2$ or NULL. A small "p-value" indicates that the data are inconsistent with the null hypothesis. The TEST option specifies the form of test used for the one-sample test; either an exact test or a Normal approximation can be selected.

Printed output is controlled by the PRINT option with settings:

summary	mean, sample size, standard error (for Normal approximation);
test	Normal approximation and probability level, or just probability level for the exact test;
confidence	confidence interval for the difference between the mean and NULL for a one-sample test, or the two means for a two-sample test.

The default is to print everything.

By default a 95% confidence interval is calculated, but this can be changed by setting the CIPROBABILITY option to the required value (between 0 and 1).

Results can be saved using the NORMAL, PROBABILITY, LOWER and UPPER parameters. NORMAL saves the Normal approximation for the one- and two-sample tests, PROBABILITY saves the probability level. LOWER and UPPER save the lower and upper limits, respectively, of the confidence interval.

Options: PRINT, METHOD, TEST, S1, S2, CIPROBABILITY, NULL.
Parameters: MU1, MU2, NORMAL, PROBABILITY, LOWER, UPPER.

Method

A standard Normal approximation is used for both the one- and two-sample tests. The exact test and confidence intervals are based on the methodology described in Chapter 4 (page 141) of Arimitage & Berry (1994).

Reference

Arimitage, P. & Berry, G. (1994). *Statistical Methods in Medical Research*. Blackwell Science, Oxford.

See also

Procedures: BNTEST, SPNTEST, TTEST.
Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Regression analysis.

POSSEMIDEFINITE

Calculates a positive semi-definite approximation of a non-positive semi-definite symmetric matrix (L.C.P Keizer, M. Malosetti & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	Controls printed output (approximation, eigenvalues, epsilon); default * i.e. none
EPSILON = *scalar*	Specifies the lowest eigenvalue for the positive semi-definite matrix; default 0.0001

Parameters

OLDSYMMETRICMATRIX = *symmetric matrices*	Symmetric matrices to approximate
NEWSYMMETRICMATRIX = *symmetric matrices*	Positive semi-definite approximations to the old symmetric matrices

Description

POSSEMIDEFINITE forms a positive semi-definite symmetric matrix to approximate an input symmetric matrix that is not positive semi-definite. The original symmetric matrix is supplied by the OLDSYMMETRICMATRIX parameter, and the new approximate matrix can be saved by the NEWSYMMETRICMATRIX parameter.

The EPSILON option specifies the lowest eigenvalue for the positive semi-definite symmetrical matrix; default 0.0001. Printed output is controlled by the PRINT option, with settings:

approximation	prints the positive semi-definite symmetric matrix approximating the original matrix,
eigenvalues	prints the eigenvalues, and
epsilon	prints the value used to set the lowest eigenvalue for the approximate matrix.

By default, nothing is printed.

Options: PRINT, EPSILON.
Parameters: OLDSYMMETRICMATRIX, NEWSYMMETRICMATRIX.

Method

POSSEMIDEFINITE uses the FLRV directive to calculate the eigenvalues and eigenvectors of the input symmetric matrix. If the matrix contains missing values they are replaced by zero. All eigenvalues below the value specified by the EPSILON option are replaced by that value. The positive semi-definite matrix is then calculated as

```
V +* D *+ TRANSPOSE(V)
```

where V is the matrix of eigenvectors, and D is a diagonal matrix containing the new eigenvalues.

See also

Procedure: LINDEPENDENCE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

PPAIR

Displays results of t-tests for pairwise differences in compact diagrams (P.W. Goedhart, H. van der Voet & D.C. van der Werf).

Options

PRINT = *string token* — What to print (items, groups); default grou

PROBABILITY = *scalar* or *symmetric matrix* — Level of significance of pairwise comparison tests; default 0.05

Parameters

TPROBABILITIES = *symmetric matrices* — Probabilities of tests of pairwise comparisons

DIFFERENCES = *symmetric matrices*, *variates* or *tables* — What to print alongside the labels of TPROBABILITIES; default *

LABELS = *texts* — Text vector labelling the output; if unset the row labels of TPROBABILITIES and the diagonal of DIFFERENCES (if set) are used

ITEMLETTERS = *texts* — Saves the letters showing the items not significantly different from each item

GROUPLETTERS = *texts* — Saves the letters showing groups of items not significantly different from each other

Description

Procedures RPAIR and PAIRTEST produce a symmetric matrix of two-sided t-probabilities for tests of all pairwise differences of estimates. PPAIR displays this matrix at a specified level of significance in two compact schematic diagrams. This is especially useful when the number of estimates is large.

Input to PPAIR is a symmetric matrix TPROBABILITIES containing probabilities of the set of pairwise comparisons. The level of significance can be set by the PROBABILITY option. A common level is specified by a scalar, while a symmetric matrix specifies a level for each comparison separately (which may be useful for some multiple comparison methods). Output is labelled by the row labels of TPROBABILITIES. If parameter DIFFERENCES is set to a symmetric matrix the diagonal of this matrix is printed alongside these labels (with number of decimals as defined at declaration of DIFFERENCES). This is especially useful if DIFFERENCES is saved by RPAIR or PAIRTEST because it then contains the estimates on the diagonal. DIFFERENCES can also be set to a variate or table. Alternatively the output can be labelled by specifying parameter LABELS.

PRINT controls which diagram is printed. PRINT=items produces a diagram which should be read line by line. Each item (represented by a letter) is followed by those items (again represented by letters) not significantly different from that item. When there are more than 52 items, letters are repeated. PRINT=groups is only useful when the TPROBABILITIES are sorted in a sensible order, for example by specifying SORT=yes in RPAIR or PAIRTEST. This produces a diagram in which items followed by a common letter are not significantly different. Such items are said to form a homogeneous group. This is similar to common underlining of items with non-significantly different estimates. In constructing this diagram the philosophy of multistage testing is followed, see the Method section. The letters can be saved, in texts, by the ITEMLETTERS and GROUPLETTERS parameters.

Options: PRINT, PROBABILITY.

Parameters: TPROBABILITIES, DIFFERENCES, LABELS, ITEMLETTERS, GROUPLETTERS.

Method

The construction of the diagram for PRINT=groups is as follows. First the difference between the first and last item of the complete set of n items is checked for significance. Then the first and last item of all subsets of $n-1$ consecutive items are checked, followed by all subsets of $n-2$ items, and so on. If non-significance is found between the first and last item of a subset, all items of the subset are said to form a homogeneous group and they receive the same letter. This is only sensible when the TPROBABILITIES are sorted according to the estimates. The diagram only consists of homogeneous groups which are not a part of a larger group.

It is obvious that items in a homogeneous group can be significantly different. This is not displayed in the diagram, although a message is printed if this occurs. If there are no significant differences within homogenous groups, both diagrams essentially contain the same information; PRINT=groups then gives a more concise representation.

Action with RESTRICT

Restrictions on DIFFERENCES and LABELS are ignored.

See also

Procedures: ALLDIFFERENCES, AMCOMPARISON, AUMCOMPARISON, PAIRTEST, RPAIR.
GenStat Reference Manual 1 Summary section on: Regression analysis.

PRCORRELATION

Calculates probabilities for product moment correlations (R.W. Payne).

Option

NOBSERVATIONS = *scalar* Number of observations from which the correlation(s) were calculated

Parameters

DATA = *scalars*, *variates*, *tables*, *matrices*, *diagonal matrices* or *symmetric matrices*
Correlations for calculating probabilities or cumulative lower probabilities for calculating equivalent deviates

CLPROBABILITY = *scalars*, *variates*, *tables*, *matrices*, *diagonal matrices* or *symmetric matrices*
Saves cumulative lower probabilities

CUPROBABILITY = *scalars*, *variates*, *tables*, *matrices*, *diagonal matrices* or *symmetric matrices*
Saves cumulative upper probabilities

PROBABILITY = *scalars*, *variates*, *tables*, *matrices*, *diagonal matrices* or *symmetric matrices*
Saves probability densities

CORRELATION = *scalars*, *variates*, *tables*, *matrices*, *diagonal matrices* or *symmetric matrices*
Saves correlations

Description

PRCORRELATION calculates probabilities and equivalent deviates for the product moment correlation coefficient. The number of observations must be supplied by the NOBSERVATIONS option, in a scalar. The DATA parameter supplies the data values in any numerical data structure (scalar, variate, matrix, table, etc.). If these are correlations, you can save various probabilities using the parameters CLPROBABILITY (cumulative lower), CUPROBABILITY (cumulative upper) and PROBABILITY (probability density). Alternatively, if DATA contains probabilities, you can save equivalent correlation values using the CORRELATION parameter.

Option: NOBSERVATIONS.
Parameters: DATA, CLPROBABILITY, CUPROBABILITY, PROBABILITY, CORRELATION.

Method

PRCORRELATION uses the fact that, for a correlation r based on n observations, the variable

$$t = r \times \sqrt{((n - 2) / (1 - r^2))}$$

has a t distribution on n-2 degrees of freedom.

Action with RESTRICT

If a DATA variate is restricted, the calculations will be preformed only on the specified units.

See also

Directive: CORRELATE.
Procedures: DCORRELATION, FCORRELATION.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

PRDOUBLEPOISSON

Calculates the probability density for the double Poisson distribution (V.M. Cave).

Options

PRINT = *string tokens*	Controls printed output (probability, summary); default prob
PLOT = *string token*	Whether to plot the *k* terms used to approximate the normalizing constant by the kpartialsum method (yes, no); default no
METHOD = *string token*	How to approximate the normalizing constant (kpartialsum, edgeworth); default kpar
LOCATION = *scalar* or *variate*	Location parameter; no default, must be set
SHAPE = *scalar* or *variate*	Shape parameter; default 1
MAXCYCLE = *scalar* or *variate*	Limits the number of terms, *k*, used to approximate the normalizing constant by the kpartialsum method; default MAX(1000, 2*LOCATION)
TOLERANCE = *scalar*	Convergence criterion used when approximating the normalizing constant by the kpartialsum method; default 1E-12

Parameters

DATA = *scalar* or *variate*	Non-negative integer values for which the double Poisson probabilities are to be calculated
DECIMALS = *scalars*	Number of decimal places for printing; default *
PROBABILITY = *variate*	Saves the probabilities

Description

PRDOUBLEPOISSON calculates the probability density for the two-parameter double Poisson distribution. The double Poisson probability density is given by

$$P(X=x) = c(\mu,\theta)\ \theta^{1/2}\ e^{-\theta\mu}\ (\ e^{-x}\ x^{x}\ /\ x!)\ (e\ \mu\ /\ x)^{\theta x}$$

for $\mu > 0$, $\theta > 0$, x = 0, 1, 2 ...

where $c(\mu,\theta)$ is the normalizing constant, μ is the location parameter, and θ is the shape parameter. The double Poisson distribution is over-dispersed when $\theta > 1$, under-dispersed when $0 < \theta < 1$, and is identical to the Poisson distribution when $\theta = 1$.

The non-negative integers, for which the double Poisson probabilities are to be calculated, are supplied by the DATA parameter.

The location parameter, μ, must be specified using the LOCATION option. The shape parameter, θ, can be set using the SHAPE option; default 1. For both the LOCATION and SHAPE options, either a single value (scalar or variate of length 1) or a variate containing the same number of values as DATA may be supplied.

The METHOD option specifies the method used to approximate the normalizing constant. The default (METHOD=kpartialsum) is to use the more accurate and reliable *k*-th partial sum method proposed by Zou *et al.* (2013). This method involves summing the first *k* terms of an infinite sum (see the *Method* section). The number of terms, *k*, is determined by the TOLERANCE and MAXCYCLE options. The TOLERANCE option can supply a scalar to specify the tolerance for convergence of the infinite sum; default 1E-12. The MAXCYCLE option places a limit on *k*, where the default is the maximum of 1000 and twice the value of the location parameter. If the infinite sum fails to converge within *k* = MAXCYCLE, the probability density is not calculated and a warning is given. MAXCYCLE may supply either a single value (scalar or variate of length 1) or a variate containing the same number of values as DATA. (However, if both LOCATION and SHAPE supply single values, only the first value of MAXCYCLE is used.) The PLOT option allows

you to request a plot of the *k* terms used to approximate the normalizing constant. By default no plot is produced.

Although the *k*-th partial sum method converges very quickly when the location parameter is small, convergence for large values of the location parameter requires a large value for *k*. The closed-form Edgeworth series method of Efron (1986) may then provide an alternative way of approximating the normalizing constant (`METHOD=edgeworth`). However, the Edgeworth series approximation is highly unreliable for small values of the location parameter, i.e. values less than about 10.

Printed output is controlled by the `PRINT` option, with settings:

`probability` (the default)	prints the probability density, and
`summary`	prints a description and a table containing; the data value, the location and shape parameters, the approximation of the normalizing constant, *k* (if `METHOD=kpartialsum`), and the probability density.

The `DECIMALS` parameter allows you to set the number of decimal places to appear in the printed output.

The `PROBABILITY` parameter can save the probability densities, in a variate.

Options: `PRINT`, `PLOT`, `METHOD`, `LOCATION`, `SHAPE`, `MAXCYCLE`, `TOLERANCE`.
Parameters: `DATA`, `DECIMALS`, `PROBABILITY`.

Method

The normalizing constant for the double Poisson distribution, $c(\mu,\theta)$, is given by an infinite sum. `PRDOUBLEPOISSON` offers two methods for approximating the constant: the *k*-th partial sum method of Zou et al. (2013), i.e. `METHOD=kpartialsum`, and the Edgeworth series method of Efron (1986), i.e. `METHOD=edgeworth`.

The *k*-th partial sum method uses the sum of the first *k* terms of the infinite sum. The number of terms is determined by the `TOLERANCE` option, which specifies the convergence criterion. The infinite sum is assumed to have converged when

$$f_{\mu,\theta}(X = k-1) > f_{\mu,\theta}(X = k)$$

and

$$f_{\mu,\theta}(X = k) < \texttt{TOLERANCE}.$$

If the infinite sum fails to converge within k = `MAXCYCLE`, the probability density is not calculated and a warning is given.

The Edgeworth series method of Efron (1986) provides a closed-form approximation to the infinite sum.

$$1 / c(\mu,\theta) \quad = \quad \sum_{x=0\ldots\infty} f_{\mu,\theta}(x)$$

The *k*-th partial sum method is more accurate and more reliable than the Edgeworth series approximation. In particular, the Edgeworth series method is highly unreliable when the location parameter, θ, is small (i.e. less than about 10), and may even produce negative values.

Action with `RESTRICT`

The `DATA`, `LOCATION`, `SHAPE` and `MAXCYCLE` variates can be restricted.

References

Efron, B. (1986). Double exponential families and their use in generalized linear regression. *Journal of the American Statistical Association*, **81**, 709-721.

Zou, Y., Geedipally, S.R. & Lord, D. (2013). Evaluating the double Poisson generalized linear model. *Accident Analysis & Prevention*, **59**, 497-505.

See also
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

PREWHITEN

Filters a time series before spectral analysis (A.W.A. Murray).

Option

PHI = *scalar*	Specifies the value of the parameter used in filtering; default 0.99

Parameters

SERIES = *variates*	Input series
FILTERED = *variates*	Output series

Description

PREWHITEN provides filtering of time series data prior to spectral analysis. Parameters SERIES and FILTERED specify the input and output series, respectively. The filtered series *y* is given by

$$y_t = x_t - \varphi \times x_{t-1}$$

where *x* is the input series. (Thus $\varphi = 1$ would give first differencing.) The value of φ is specified by the PHI option; the default value of φ=0.99 is often suitable. Alternatively, an empirical approach is to use the value

$$\varphi = (1 - 1/L)$$

where *L* is the lag at which inspection suggests that the autocorrelation in the series becomes negligible.

To "recolour" the spectrum of the series after estimation, you can multiply by

$$1/((1 + \varphi^2) - (2 \times \varphi \times \cos(2\pi \times f)))$$

where *f* is the frequency at which the spectrum is estimated.

Option: PHI. Parameters: SERIES, FILTERED.

Method

The procedure uses the TFILTER directive with two TSMs defined as follows:

```
TSM filter; ORDER=!(1,0,0); PARAM=!(1,0,0,PHI)
TSM arima;  ORDER=!(0,1,0); PARAM=!(1,0,0)
TFILTER SERIES; NEWSERIES=FILTERED; FILTER=filter; ARIMA=arima
```

The procedure is based on ideas from Granville Tunnicliffe Wilson, University of Lancaster.

Action with RESTRICT

The behaviour is as for the TFILTER directive.

See also

Directive: FOURIER.

Procedures: DFOURIER, MCROSSPECTRUM, PERIODTEST, REPPERIODOGRAM, SMOOTHSPECTRUM.

Genstat Reference Manual 1 Summary section on: Time series.

PRIMEPOWER

Decomposes a positive integer into its constituent prime powers (I. Wakeling & R.W. Payne).

Option

PRINT = *string token*	Controls printed output (decomposition); default *

Parameters

NUMBER = *scalars*	Number to be decomposed
PRIMES = *pointers*	Prime factors of NUMBER
POWERS = *pointers*	Powers of the prime factors in NUMBER

Description

Procedure PRIMEPOWER decomposes the integer specified by the NUMBER parameter into its constituent prime powers. The results can be saved using the PRIMES and POWERS parameters. These return pointers to a set of scalars storing, respectively, the relevant prime numbers and their powers. If NUMBER is not a positive integer, the pointers will each contain a single scalar containing a missing value. The decomposition can also be printed by setting option PRINT=decomposition.

Option: PRINT.
Parameters: NUMBER, PRIMES, POWERS.

Method

PRIMEPOWER uses the standard Genstat calculation directives.

See also

Procedure: NCONVERT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

PRKTAU

Calculates probabilities for Kendall's rank correlation coefficient τ (D.B. Baird).

No options

Parameters

`N` = *scalars*	Sizes of the first groups of observations
`TAU` = *scalars*	Values of Kendall's τ statistic
`CLPROBABILITY` = *scalars*	Cumulative lower probability of `TAU`
`CUPROBABILITY` = *scalars*	Cumulative upper probability of `TAU`
`PROBABILITY` = *scalars*	Probability density of `TAU`
`LPROBABILITIES` = *variates*	Probability densities of −1...`TAU`
`LTAU` = *variates*	Values of Tau at corresponding values of `LPROBABILITIES`

Description

`PRKTAU` calculates various probabilities for the Kendall's rank correlation coefficient, τ (tau). The τ statistic arises from Kendall's rank correlation test, which can be used to give a nonparametric assessment as to whether paired samples are correlated. τ is calculated as

```
T / NCOMBINATIONS(N; 2)
```

where T is

$$\sum_{i=1...N} \{ \sum_{j=i...N} \{ \text{Sign}(x_i - x_j) \times \text{Sign}(y_i - y_j) \} \}.$$

The number of sample pairs of observations is specified by the `N` parameter, and the `TAU` parameter specifies the value of the Kendall rank correlation coefficient for which the probabilities are required. The `CLPROBABILITY` and `CUPROBABILITY` parameters can specify scalars to save the cumulative lower and upper probabilities, pr($s \leq \tau$) and pr($s > \tau$) respectively. `PROBABILITY` can save the probability density at τ, pr($s = \tau$), and `LPROBABILITIES` can save a variate containing the densities for −1...τ, and `LTAU` can save the values of τ for the elements in `LPROBABILITIES`.

Options: none.

Parameters: `N`, `TAU`, `CLPROBABILITY`, `CUPROBABILITY`, `PROBABILITY`, `LPROBABILITIES`, `LTAU`.

Method

The procedure calculates the coefficents of the generating function for the Kendall rank correlation coefficient under the null hypothesis using recurrence functions (See van de Weil *et al*. 1999). The central limit theorem is used when *N* exceeds 35, and a Normal approximation of the cumulative density function is returned.

Reference

van de Wiel, M.A. Di Bucchianico, A. & van de Laan, P. (1999). Symbolic computation and exact distributions of nonparametric test statistics. *The Statistician*, **48**, 507-516.

See also

Procedure: `KTAU`.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

PRMANNWHITNEYU

Calculates probabilities for the Mann-Whitney U statistic (D.B. Baird & J.H. Klotz).

No options

Parameters

N1 = *scalars*	Sizes of the first groups of observations
N2 = *scalars*	Sizes of the second groups of observations
U = *scalars*	Values of the U statistic
TIES = *scalars*	Number of tied observations; default 0
CLPROBABILITY = *scalars*	Cumulative lower probability of U
CUPROBABILITY = *scalars*	Cumulative upper probability of U
PROBABILITY = *scalars*	Probability density of U
LPROBABILITIES = *variates*	Probability densities of 0...U
EXIT = *scalars*	Set to 1 if it has not been possible to calculate the probabilities when there are ties, otherwise 0

Description

PRMANNWHITNEYU calculates various probabilities for the Mann-Whitney U statistic. This statistic arises from the Mann-Whitney U test, which can be used to give a nonparametric assessment as to whether two samples arise from the same probability distribution. If the samples are $\{x_i: i=1...n_1\}$ and $\{y_j: j=1...n_2\}$, then the Mann-Whitney U statistic is defined as the number of pairs (x_i, y_j) with $x_i < y_j$. In Genstat, U can be calculated by the MANNWHITNEY procedure (which calls PRMANNWHITNEYU to obtain the required probability values).

The number of samples in the two sets of observations are specified by the N1 and N2 parameters, respectively. The U parameter specifies the value of the U statistic for which the probabilities are required, and the TIES parameter supplies the number of tied observations (if any). PRMANNWHITNEY may not be able to calculate the probabilities in every Genstat implementation when there are ties, and so there is also a parameter EXIT that you can set to check whether there have been problems (if the calculation has been successful EXIT=0, otherwise EXIT=1). The CLPROBABILITY and CUPROBABILITY parameters can specify scalars to save the cumulative lower and upper probabilities, pr($u \leq$ U) and pr($u >$ U) respectively. PROBABILITY can save the probability density at U, pr($u =$ U), and LPROBABILITIES can save a variate containing the densities for 0...U.

Options: none.

Parameters: N1, N2, U, TIES, CLPROBABILITY, CUPROBABILITY, PROBABILITY, LPROBABILITIES, EXIT.

Method

The procedure calculates the coefficents of the generating function for the Mann-Whitney statistic under the null hypothesis using recurrence functions. The central limit theorem is used when the smaller of N1 and N2 exceeds 50, and a Normal approximation of the CDF is returned. (See Harding 1983). A separate program, that uses the method of Klotz & Cheung (1995), is called using PASS when there are ties. This may not be feasible in every Genstat implementation.

References

Harding, E.F. (1983) An efficient, minimal-storage procedure for calculating the Mann-Whitney U, Generalised U and similar distributions. *Applied Statistics*, **33**, 1-6.

Klotz, J.H. & Cheung, Y.K. (1995). The Mann Whitney Wilcoxon distribution using linked lists. *Statistica Sinica*, **7**, 805-813.

See also
Procedure: MANNWHITNEY.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

PROBITANALYSIS

Fits probit models allowing for natural mortality and immunity (R.W. Payne).

Options

PRINT = *string tokens*	Printed output required (model, summary, estimates, correlations, fittedvalues, monitoring, effectivedoses); default mode, summ, esti, fitt, effe
TRANSFORMATION = *string token*	Transformation to be used (probit, logit, complementaryloglog); default prob
MORTALITY = *string token*	Whether to estimate natural mortality (omit, estimate); default omit
IMMUNITY = *string token*	Whether to estimate natural immunity (omit, estimate); default omit
GROUPS = *factor*	Defines groups for an analysis of parallelism; default * i.e. no groups
SEPARATE = *string tokens*	Which parameters (apart from intercept) should be estimated separately for different groups (slope, mortality, immunity, notintercept); default * i.e. none
LD = *scalar* or *variate*	Effective, or lethal, doses to be estimated, other than 50
CIPROBABILITY = *scalar*	Probability level for the confidence interval of effective doses; default 0.95, i.e. a 95% confidence interval
LOGBASE = *string token*	Base of antilog transformation to be applied to LD's (ten, e); default * i.e. none
DISPERSION = *scalar*	Controls the use of a heterogeneity factor in the calculation of s.e.s etc; with the default of 1 no factor is used, a missing value * estimates the heterogeneity from the residual deviance
FITMETHOD = *string token*	Method to use to fit the model (generalizednonlinear, nonlinear) default nonl for Wadley's problem, otherwise gene
MAXCYCLE = *scalar*	Maximum number of iterations for fitting the model; default 30

Parameters

Y = *variates*	Number of subjects responding in each batch
DOSE = *variates*	Dose received by each batch of subjects
NBINOMIAL = *variates*, *scalars* or *factors*	Variate specifying the number of subjects in each batch, or factor specifying groupings of the observations assumed to have equal expected total numbers of subjects in Wadley's problem; if omitted, assumes Wadleys's problem with all observations having the same expected total number of subjects
INITIAL = *variates*	Initial values for parameters
STEPLENGTHS = *variates*	Step lengths for parameters
LDESTIMATES = *variates*	Saves estimates of the effective, or lethal, doses
LDLOWER = *variates*	Saves lower values of the confidence intervals for the estimates of the effective, or lethal, doses (for FITMETHOD=gene only)

`LDUPPER` = *variates*	Saves upper values of the confidence interval values for the estimates of the effective, or lethal, doses (for `FITMETHOD=gene` only)

Description

Probit analysis is a way of modelling the relationship between a stimulus, like a drug, and a quantal response (success/failure). It is assumed that for each subject, there is a certain level of dose of the stimulus below which it will unaffected, but above which it will respond. This level of dose, known as its tolerance, will vary from subject to subject within the population.

For example, it is often assumed that the tolerance of houseflies to logarithm of the dose of an insecticide will follow a Normal distribution; so, if we were to plot the proportion of the population with each tolerance against log dose, we would obtain the familiar bell-shaped curve. Likewise, if we plot the probability that a randomly-selected individual will respond, against the logarithm of dose, we would obtain a sigmoid (S-shaped) curve limited below by zero and above by one. To make the relationship linear, it is usual to transform the y-axis either to probits or to Normal equivalent deviates. In Genstat

Probit(*P*%) = NED(*P*%/100)

The Normal equivalent deviate may be familiar as the transformation that is used to produce "probability" graph paper.

In probit analysis, we are interested in estimating the equation of that line. This can be done by perfoming an experiment in which there are several batches of subjects, each of which is given a different dose of the stimulus. The data then consists of a variate indicating the number of subjects that responded out of each batch, a variate to show the dose given to each batch, and a final variate for the total numbers of subjects in the batches; these are specified by parameters `Y`, `DOSE` and `NBINOMIAL`, respectively.

The `NBINOMIAL` parameter can be omitted if the total numbers cannot be measured, as in some fumigation experiments ("Wadley's problem"; see for example Finney 1971, pages 202-8). The assumption is that the total numbers receiving the doses will come from the same Poisson distribution, and the mean of this distribution is then estimated in the analysis. Alternatively, `NBINOMIAL` can specify a factor to indicate groupings of the doses whose total numbers are expected to come from the same distributions.

The `PRINT` option controls printed output:

`model`	details of the model that has been fitted,
`summary`	summary analysis-of-variance table,
`estimates`	parameter estimates and standard errors,
`correlations`	correlations between parameter estimates,
`fittedvalues`	fitted values and residuals,
`monitoring`	information about the fitting process, and
`effectivedoses`	effective, or lethal, doses (see parameter `LD` below).

By default, `PRINT=mode,summ,esti,fitt,effe`.

The `TRANSFORMATION` option allows other transformations to be selected. Putting `TRANSFORMATION=logit` requests a logit transformation:

logit(*P*%) = log(*P*% / (100 – *P*%))

This is very like the probit but approaches zero (to the left) and one (to the right) rather more slowly. The other possibility is the complementary log-log (=log(–log(100–*P*%)), which is relevant to the "one-hit" model (that is infection processes where just one infected particle is sufficient to cause the response).

Sometimes, subjects may respond even in the absence of any dose. For example, with some short-lived insects, some would have died simply from natural causes during the period of the experiment. By setting option `MORTALITY=estimate` this natural mortality can be included in the model and estimated. Similarly, there may be subjects that will not respond, no matter how

high the dose. Setting option IMMUNITY=estimate will include and estimate a parameter for natural immunity.

It is also often of interest to fit study the way in which the model varies for different groups of subjects. For example, there may be groups of batches of subjects, each of which is given a different drug. The GROUPS option should then specify the group to which each batch of subjects belongs, and option SEPARATE indicates which parameters of the model (slope, mortality, and/or immunity) should have separate estimates. Separate parameters are always fitted for the intercept unless you include the setting notintercept. So, if SEPARATE is left at its default value, parallel lines will be fitted with identical values for any estimates of mortality and immunity.

The LD option can request the estimation of one or more effective (or lethal) doses, specifying a scalar if there is just one, or a variate if there are several. The LOGBASE option is useful if the doses have been transformed to logarithms before calling PROBITANALYSIS. If you use LOGBASE to specify the base of the logarithms (ten or e), the back-transformed lethal doses will be printed as well.

The estimates of the effective (or lethal) doses can be saved, in a variate, by the LDESTIMATES parameter. Also, when model is fitted as a generalized nonlinear model (see the FITMETHOD option, below), the lower and upper values of the confidence intervals for the estimates can be saved by the LDLOWER and LDUPPER parameters, respectively. If LOGBASE is set, these are all back-transformed. The CIPROBABILITY option specifies the probability level for the confidence intervals; the default is 0.95, i.e. 95% confidence intervals.

The DISPERSION option can be used to request use of a heterogeneity factor in the calculation of the standard errors of the slopes and lethal doses (see Finney 1971, pages 70-74). The standard assumptions for probit analysis are that the observations have binomial distributions in probit lines and planes, or Poisson distributions in Wadley's problem. Under these circumstances, the residual deviance will follow a Chi-square distribution. The residual deviance should on average be equal to its number of degrees of freedom. A significantly large value may indicate that there are other (possibly unknown) factors affecting the subjects, for example that the conditions were not uniform during the experiment. Alternatively it may occur because the subjects did not react independently, for example because there were sub-populations of genetically related individuals. If the large Chi-square seems to arise because the residuals are larger in general than expected (overdispersion) and not because of systematic deviations from the fitted relationship, it is sensible to increase the standard errors by a heterogeneity factor equal to the residual mean deviance. This can be requested by setting option DISPERSION=*. Alternatively DISPERSION can be set to a known value if one is available.

When the FITMETHOD option is set to generalizednonlinear, the model is fitted as a generalized nonlinear model, using the FIT directive. The alternative setting, nonlinear, fits it as a nonlinear model using FITNONLINEAR. Apart from minor numerical differences, the two methods should generate the same results. Generalized nonlinear models allow a confidence region to be generated for lethal doses, and these are used as default for all situations except Wadley's problem. The nonlinear method is more accurate, and is thus used as the default for the more difficult situation presented by Wadley's problem. However, there is the limitation that you cannot use the notintercept setting of the SEPARATE option with the nonlinear method.

The final two parameters, INITIAL and STEPLENGTHS, allow initial values and step lengths to be specified for the optimization. For a generalized nonlinear model, the order of parameters is: total(s) for Wadley's problem (if appropriate), mortality parameters (if any) and immunity parameters (if any); the slopes and intercepts are fitted as regression parameters. For a nonlinear model, the order of parameters is: LD50(s), slope(s), mortality parameters (if any) and immunity parameters (if any); the totals for Wadley's problem, if required, as fitted as linear parameters. The MAXCYCLE option sets a limit on the number of iteractions used during fitting (default 30). Parameter estimates, fitted values, residuals, and so on, can be saved after running the procedure, by using the RKEEP directive in the usual way.

Options: PRINT, TRANSFORMATION, MORTALITY, IMMUNITY, GROUPS, SEPARATE, LD, CIPROBABILITY, LOGBASE, DISPERSION, FITMETHOD, MAXCYCLE.
Parameters: Y, DOSE, NBINOMIAL, INITIAL, STEPLENGTHS, LDESTIMATES, LDLOWER, LDUPPER.

Method
For FITMETHOD=generalizednonlinear a calculated link is used to take account of any mortality or immunity parameters, and a calculated distribution to allow estimation of totals for Wadley's problem. The fitting is carried out by FIT (with the CALCULATION option set if any totals, mortality or immunity parameters are to be estimated), and procedure FIELLER is used to obtain LD values.

For FITMETHOD=nonlinear initial values are obtained, if necessary, using the Genstat facilities for generalized linear models, ignoring any mortality or immunity. Expressions specifying the model are defined in sets of nested IF-blocks, taking account of the settings for example of TRANSFORMATION and GROUPS. The fitting is carried out by the FITNONLINEAR directive, and any extra LD values are estimated using RFUNCTION.

Action with RESTRICT
The Y variate, the DOSE variate, or the GROUPS factor can be restricted to indicate that the model is to be fitted only to a subset of the units.

Reference
Finney, D.J. (1971). *Probit Analysis (third edition)*. Cambridge University Press, Cambridge.

See also
Procedures: FIELLER, WADLEY.
Genstat Reference Manual 1 Summary section on: Regression analysis.

PRSPEARMAN

Calculates probabilities for Spearman's rank correlation statistic (D.B. Baird).

No options

Parameters

N = *scalars*	Numbers of pairs of observations
CORRELATION = *scalars*	Values of the signed rank statistic
CLPROBABILITY = *scalars*	Cumulative lower probability of CORRELATION
CUPROBABILITY = *scalars*	Cumulative upper probability of CORRELATION
PROBABILITY = *scalars*	Probability density of CORRELATION
UPROBABILITIES = *variates*	Probability densities of CORRELATION...1
UCORRELATION = *variates*	Values of CORRELATION at corresponding elements of UPROBABILITIES

Description

PRSPEARMAN calculates various probabilities for Spearman's rank correlation coefficient (see procedure SPEARMAN). These can be used to give a nonparametric assessment of whether paired samples are correlated.

$$\text{correlation} = \sum_{i=1\ldots N} ((R_i-(N+1)/2) \times (S_i-(N+1)/2)) / (N \times (N^2-1) / 12$$

where R_i and S_i are the ranks of X_i and Y_i respectively.

The number of sample pairs of observations is specified by the N parameter, and the CORRELATION parameter specifies the value of the rank correlation for which the probabilities are required. The CLPROBABILITY and CUPROBABILITY parameters can specify scalars to save the cumulative lower and upper probabilities,

$$\Pr.(s \leq \text{CORRELATION})$$

and

$$\Pr.(s > \text{CORRELATION})$$

respectively. PROBABILITY can save the probability density at CORRELATION,

$$\Pr.(s == \text{CORRELATION}),$$

UPROBABILITIES can save a variate containing the densities for CORRELATION...1, and UCORRELATION can save the values of CORRELATION for the elements in UPROBABILITIES.

Options: none.

Parameters: N, CORRELATION, CLPROBABILITY, CUPROBABILITY, PROBABILITY, UPROBABILITIES, UCORRELATION.

Method

The procedure uses PASS to call an external program which calculates the coefficients of the generating function for the Spearman rank correlation coefficient under the null hypothesis using recurrence functions (see van de Weil *et al*. 1999). A t approximation is used when N exceeds 20.

Action with RESTRICT

Restrictions are not applicable to any of the parameters.

Reference

van de Wiel, M.A., Di Bucchianico, A. & van de Laan, P. (1999). Symbolic computation and exact distributions of nonparametric test statistics. *The Statistician*, **48**, 507-516.

See also
Procedure: SPEARMAN.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

PRWILCOXON

Calculates probabilities for the Wilcoxon signed-rank statistic (D.B. Baird & J.H. Klotz).

No options

Parameters

N = *scalars*	Sizes of the first groups of observations
SIGNEDRANK = *scalars*	Values of the signed rank statistic
DATA = *variates*	Data variate holding differences between each pair of samples (required only if ties are to be allowed for)
CLPROBABILITY = *scalars*	Cumulative lower probability of SIGNEDRANK
CUPROBABILITY = *scalars*	Cumulative upper probability of SIGNEDRANK
PROBABILITY = *scalars*	Probability density of SIGNEDRANK
LPROBABILITIES = *variates*	Probability densities of 0...SIGNEDRANK
EXIT = *scalars*	Set to a positive error code if it has not been possible to calculate the probabilities when there are ties, otherwise 0

Description

PRWILCOXON calculates various probabilities associated with the Wilcoxon signed-rank statistic (or matched-pairs statistic). This statistic arises from the Wilcoxon test, which can be used to give a nonparametric assessment of whether paired samples arise from the same probability distribution, or of whether a single sample has a given median. The Wilcoxon test operates on a variate of differences between paired samples. It calculates the ranks of the absolute values of the differences, and then the sum of the ranks for the negative and for the positive differences. The statistic is the smaller of these two sums. In Genstat, this can be calculated by the WILCOXON procedure (which calls PRWILCOXON to obtain the required probability values). PRWILCOXON works on the sum of the ranks of the positive differences, which takes values from 0 to N×(N+1)/2.

The number of sample pairs of observations is specified by the N parameter, and the SIGNEDRANK parameter specifies the value of the signed rank statistic for which the probabilities are required. If there are ties in the data, you should also supply the original data variate, using the DATA parameter. The CLPROBABILITY and CUPROBABILITY parameters can specify scalars to save the cumulative lower and upper probabilities, pr($s \leq$ SIGNEDRANK) and pr($s >$ SIGNEDRANK) respectively. PROBABILITY can save the probability density at SIGNEDRANK, pr($s =$ SIGNEDRANK), and LPROBABILITIES can save a variate containing the densities for 0...SIGNEDRANK.

The probabilities are exact for values of N up to 100, and also for values of N between 100 and 200 provided SIGNEDRANK is less than 10001.

Options: none.

Parameters: N, SIGNEDRANK, DATA, CLPROBABILITY, CUPROBABILITY, PROBABILITY, LPROBABILITIES, EXIT.

Method

The procedure calculates the coefficents of the generating function for the Signed Rank statistic under the null hypothesis using recurrence functions (See van de Weil *et al*. 1999). The central limit theorem is used when N exceeds 200, and a Normal approximation of the CDF is returned. A separate program, that uses the method of Klotz & Cheung (1995), is called through PASS when there are ties. This may not be feasible in every Genstat implementation.

References

van de Wiel, M.A., Di Bucchianico, A. & van de Laan, P. (1999). Symbolic computation and exact distributions of nonparametric test statistics. *The Statistician*, **48**, 507-516.

Klotz, J.H. & Cheung, Y.K. (1995). The Mann Whitney Wilcoxon distribution using linked lists. *Statistica Sinica*, **7**, 805-813.

See also

Procedure: WILCOXON.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

PSPLINE

Calculates design matrices to fit a P-spline as a linear mixed model (S.J. Welham).

Options

NSEGMENTS = *scalar*	Specifies the number of segments between boundaries; default * obtains a value automatically
DEGREE = *scalar*	Degree of polynomial used to form the underlying spline basis functions; default 3
DIFFORDER = *scalar*	Differencing order for penalty; default 2
LOWER = *scalar*	Specifies the lower boundary; default takes the minimum value in X
UPPER = *scalar*	Specifies the upper boundary; default takes the maximum value in X
ORTHOGONALIZETO = *variate*	Variate to use to get an orthogonalized basis; default * i.e. orthogonalization with respect to X
SCALING = *scalar*	Scaling of the XRANDOM terms; (automatic, none); default auto

Parameters

X = *variates*	The explanatory variate for which the basis functions are required
XFIXED = *matrices*	Saves the design matrix to define the fixed terms (excluding the constant) for fitting the P-spline
XRANDOM = *matrices*	Saves the design matrix to define the random terms for fitting the P-spline
KNOTS = *variates*	Saves the internal knots and boundaries used to form the basis functions
PX = *variates*	Specifies x-values at which predictions are required
PFIXED = *matrices*	Saves the design matrix for the fixed terms (excluding the constant) for the spline at the prediction points
PRANDOM = *matrices*	Saves the design matrix for the random terms for the spline at the prediction points

Description

This procedure generates the fixed and random terms required to fit a P-spline (Eilers & Marx, 1995) as a linear mixed model, using REML estimation of the smoothing parameter. The explanatory variate values at which the spline is to be calculated are specified in a variate using the X parameter. The full range of the spline can be specified by the LOWER and UPPER options; by default the lower limit is equal to the minimum value of X and the upper limit is equal to the maximum value. The region between these bounds is divided into a number of equal segments, specified by the NSEGMENTS option. The boundaries of these segments form the set of knots used to form the spline basis functions, and can be saved as a variate using the KNOTS parameter. If NSEGMENTS is unset, the number of segments is determined automatically as

$$\min([p/4], 35) + 1$$

(Ruppert 2002) where p is the number of unique values of the variate X and $[r]$ denotes the integer part of the number r.

The DEGREE option specifies the degree of polynomial that is used to form the underlying spline basis functions. The default, DEGREE=3, gives a cubic spline.

The ORTHOGONALIZETO option specifies a variate to use in orthogonalization. The set of random spline terms will then be orthogonal to the fixed terms when evaluated at the specified values. For most data sets, it is recommended to set ORTHOGONALIZETO to the variate X (the

default). The random terms will then be orthogonal to the fixed terms, and fitted values corresponding to the fixed model will represent the whole of the polynomial trend in the fitted spline. For very large data sets, this calculation can be onerous and can be approximated by making the two bases orthogonal at the knots. No orthogonalization is carried out if `ORTHOGONALIZETO` is set to a scalar value (e.g. `ORTHOGONALIZETO=0`).

The spline terms are saved as two matrices. The terms required to be fitted as fixed terms can be saved using the `XFIXED` parameter. This matrix does not include the constant term as this is added by default as part of a mixed model. When `DIFFORDER` is set to one, this is a null term and no matrix will be returned. The terms to be fitted as random can be saved using the `XRANDOM` parameter.

The random terms can be scaled so that, for a random spline matrix `Z`,

```
TRACE(Z *+ T(Z)) = NROWS(Z)
```

This ensures that the average contribution of `Z` to the variance of an observation is equal to one, and hence the overall contribution from the term is equal to the spline variance component. This removes possible computational instabilities, and improves interpretability of the spline variance component. This scaling is imposed by default, but can be avoided by setting option `SCALING=none`.

The spline terms required for prediction via `VPREDICT` can be saved using the `PXFIXED` and `PXRANDOM` parameters. The `PX` parameter defines the set of x-values at which the predictions are to be made.

Options: `NSEGMENTS`, `DEGREE`, `DIFFORDER`, `LOWER`, `UPPER`, `ORTHOGONALIZETO`, `SCALING`.
Parameters: `X`, `XFIXED`, `XRANDOM`, `KNOTS`, `PX`, `PXFIXED`, `PXRANDOM`.

Method

The P-spline of degree k with differencing order d and r knots, evaluated on variate `X`, minimizes the penalized sum of squares

$$(y - B\,\alpha)'\, R^{-1}\, (y - B\,\alpha) + \lambda\, \alpha'\, \Delta_d'\, \Delta_d\, \alpha$$

where

- B is a matrix of $b = r + k + 1$ B-spline basis functions of degree k with r equally-spaced knots evaluated at the values in `X`,
- α is a vector of $r+k+1$ unknown spline coefficients,
- λ is a smoothing parameter, and
- Δ_d is a $(b - d) \times b$ differencing matrix of order d.

This penalized sum of squares is reformulated as the estimating equations from a mixed model of the form

$$y = X\tau + Z\,u + e$$

where

- X is a design matrix containing k basis functions $x^{\{0\ldots d-1\}}$, with associated unknown parameters τ
- e is a vector of residual errors with variance $\sigma^2 R$, and

$$Z = B\,U\,S^{-1}$$

where

$$\Delta_d' = U\,S\,V'$$

is the design matrix for a set of $(b - d)$ independently and identically distributed Normal random effects $u = S\,U'\,\alpha$ with variance $\sigma_s^2 I$.

Fitting this mixed model, with known λ set equal to σ^2/σ_s^2, produces estimates that minimize the penalized sum of squares. In addition, we can estimate the smoothing parameter using `REML` via the variance component σ_s^2. This can be generalized straightforwardly to mixed models with additional fixed and random terms.

The implementation in this procedure allows the random design matrix to be orthogonalized

with respect to the fixed design matrix at a given variate. For orthogonalization with respect to the variate X, this is achieved by using random design matrix

$Z^* = (\mathrm{I} - X(X'X)^{-1}X')\mathrm{Z}$

The entirety of the polynomial trend is then captured by the fixed model. Orthogonalization with respect to a variate t is calculated as

$Z^* = Z - X(T'T)^{-1}T'\ B(\mathrm{t})\ U\ S^{-1}$

where T is a matrix holding $t^{\{0\ldots k\}}$, and B(t) is the appropriate B-spline basis evaluated at t.

When the random matrix is scaled so that trace($Z^*Z^{*\prime}$) is equal to the number of rows of Z^*, the average contribution of the spline term to the variance of each unit ($\sigma_s^2 \times$ diag($Z^*Z^{*\prime}$)) is equal to σ_s^2. This makes the spline variance component value directly comparable with the residual variance.

Note that the constant function is not included in the fixed design matrix generated by PSPLINE, as this term is added automatically to the linear mixed model by the default option setting, CONSTANT=estimate, in the VCOMPONENTS statement.

The design matrices for use in prediction are calculated by evaluating the same set of basis functions at the predict points specified by the PX option.

Action with RESTRICT

The input structures must not be restricted.

References

Currie, I.D. & Durban, M., (2002). Flexible smoothing with P-splines: a unified approach. *Statistical Modelling*, **2**, 333-349.

Eilers, P.H.C. & Marx, B.D. (1996). Flexible smoothing with B-splines and penalties. *Statistical Science*, **11**, 89-121.

Ruppert, D. (2002). Selecting the number of knots for penalised splines. *Computational & Graphical Statistics*, **11**, 735-757.

See also

Directive: VCOMPONENTS.

Procedures: LSPLINE, NCSPLINE, PENSPLINE, RADIALSPLINE, TENSORSPLINE, SPLINE.

Function: SSPLINE.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

PTAREAPOLYGON

Calculates the area of a polygon (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
AREA = *scalars*	Scalars to receive the areas of the polygons

Description

This procedure takes as input two variates containing the coordinates of a polygon (specified by the XPOLYGON and YPOLYGON parameters) and returns the area of the polygon. The area may be saved in a scalar specified using the parameter AREA.

Printed output is controlled by the PRINT option. The default setting of summary prints the area of the polygon under the heading AREA.

Option: PRINT.
Parameters: YPOLYGON, XPOLYGON, AREA.

Method

A procedure PTCHECKXY is called to check that XPOLYGON and YPOLYGON have identical restrictions. PTAREAPOLYGON then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. It then calls a procedure PTPASS to call a Fortran program to calculate the area of the polygon.

Action with RESTRICT

If XPOLYGON and YPOLYGON are restricted, only the subset of values specified by the restriction will be used in the calculations.

See also

Procedures: DPOLYGON, PTAREAPOLYGON, PTCLOSEPOLYGON, PTSINPOLYGON.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTBOX

Generates a bounding or surrounding box for a spatial point pattern (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string token*	What to print (summary); default summ
METHOD = *string token*	Type of box to form (bounding, surrounding); default boun

Parameters

Y = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
X = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
YBOX = *variates*	Variates to receive the vertical coordinates of the bounding or surrounding boxes
XBOX = *variates*	Variates to receive the horizontal coordinates of the bounding or surrounding boxes
YFRACTION = *scalars*	How much to extend the extremes of the vertical coordinates of each surrounding box as a fraction of the range of the vertical coordinates; default 0.1
XFRACTION = *scalars*	How much to extend the extremes of the horizontal coordinates of each surrounding box as a fraction of the range of the horizontal coordinates; default 0.1

Description

This procedure takes as input two variates containing the coordinates of a spatial point pattern (specified by the X and Y parameters) and returns the coordinates of either a bounding or a surrounding box, according to the setting of the METHOD option. The default, METHOD=bounding, provides a bounding box, defined as the smallest rectangle such that all the events in the spatial point pattern lie inside the box or on its boundary. The coordinates of the bounding box are the coordinates of its four corners, in the order lower left, lower right, upper right, and upper left. The surrounding box (METHOD=surrounding) is a rectangle which contains all the points. It is obtained by extending the vertical and horizontal edges of the bounding box by specified fractions of the range of values in X and Y, respectively. The parameters XFRACTION and YFRACTION can be used to specify the proportional extension required in each direction; the default value of both parameters is 0.1. The coordinates of the surrounding box are the coordinates of its four corners in the order lower left, lower right, upper right, upper left. The coordinates of the bounding or surrounding box can be saved using the parameters XBOX and YBOX.

Printed output is controlled by the PRINT option. The default setting of summary prints the coordinates of the bounding or surrounding box under the headings XBOX and YBOX.

Option: PRINT.
Parameters: Y, X, YBOX, XBOX, YFRACTION, XFRACTION.

Method

A procedure PTCHECKXY is called to check that X and Y have identical restrictions. The minimum, maximum and range of the horizontal (X) and vertical (Y) coordinates are then calculated. For a bounding box, the coordinates are calculated as (min(X), min(Y)), (max(X), min(Y)), (max(X), max(Y)), (min(X), max(Y)). For a surrounding box the coordinates are

(min(X) – XFRACTION × range(X), min(Y) – YFRACTION × range(Y)),
(max(X) + XFRACTION × range(X), min(Y) – YFRACTION × range(Y)),
(max(X) + XFRACTION × range(X), max(Y) + YFRACTION × range(Y)),
(min(X) – XFRACTION × range(X), max(Y) + YFRACTION × range(Y)).

Action with RESTRICT
If X and Y are restricted, only the subset of values specified by the restriction will be included in the calculations.

See also
Procedures: CONVEXHULL, PTSINPOLYGON.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTCLOSEPOLYGON

Closes open polygons (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson)

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

OLDYPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
OLDXPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NEWYPOLYGON = *variates*	Vertical coordinates of the closed polygons
NEWXPOLYGON = *variates*	Horizontal coordinates of the closed polygons

Description

A polygonal region of two-dimensional space is represented in Genstat by the coordinates of a sequence of points which define the boundary of the polygon with the last point implicitly connected to the first point. If the first and last pairs of coordinates are the same then the polygon is said to be closed, otherwise it is open. Sometimes it is necessary to work with a closed polygon, for example, when drawing a polygon onto a graphics device as a series of line segments. This procedure takes as input a set of coordinates which define a polygon. The parameters OLDXPOLYGON and OLDYPOLYGON specify variates containing the coordinates. The output of the procedure is a closed polygon, which is identical to the input polygon if it is already closed and otherwise consists of the input polygon with the first pair of coordinates repeated at the end. The coordinates of the closed polygon may be saved using the parameters NEWXPOLYGON and NEWYPOLYGON.

Printed output is controlled by the PRINT option. The default setting of summary prints the horizontal and vertical coordinates of the closed polygon under the headings NEWXPOLYGON and NEWYPOLYGON.

Option: PRINT.
Parameters: OLDYPOLYGON, OLDXPOLYGON, NEWYPOLYGON, NEWXPOLYGON.

Method

A procedure PTCHECKXY is called to check that OLDXPOLYGON and OLDYPOLYGON have identical restrictions. It then checks whether the first and last pairs of coordinates are the same. If they are, the DUPLICATE directive is used to copy OLDXPOLYGON to NEWXPOLYGON and OLDYPOLYGON to NEWYPOLYGON. If they are different, NEWXPOLYGON and NEWYPOLYGON are declared as variates with one more value than their old counterparts, and the EQUATE directive is used to copy the values from OLDXPOLYGON to NEWXPOLYGON and OLDYPOLYGON to NEWYPOLYGON (so that the first element of each old variate is repeated at the end of the corresponding new one).

Action with RESTRICT

If OLDXPOLYGON and OLDYPOLYGON are restricted, only the subset of values specified by the restriction will be included in the calculations.

See also

Procedures: DPOLYGON, PTCLOSEPOLYGON, PTSINPOLYGON.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTDESCRIBE

Gives summary and second order statistics for a point process (R.P. Littlejohn & R.C. Butler).

Options

PRINT = *string token*	Whether to print (statistics); default stat
SELECTION = *string tokens*	What to print (interval, trend, poisson, icorrelation, ispectrum, cspectrum, cintensity, vtcurve, all); default inte
REPRESENTATION = *string token*	How the point process is represented in the DATA variate (time, interval, zeroone); default time
GRAPHICS = *string token*	Style of graphical output, or GRAPHICS=* to avoid any graphs (lineprinter, highresolution); default high

Parameters

DATA = *variates*	Variate containing point process to be analysed
START = *scalars*	Initial time (if REPRESENTATION=time); default 0
LENGTH = *scalars*	Length of time over which process is observed; default takes the time of the last event
CITAU = *scalars*	Window width for calculating count intensity; default 0.5 × mean interval length
VTTAU = *scalars*	Window width for calculating variance-time curve; default 0.5 × mean interval length
SAVE = *pointers*	Pointer to save calculated values

Description

A point process, or series of events, is characterized both by the times at which events occur, and the intervals between events. The Poisson process is the most basic point process, with Poisson counts in any interval, and independent exponentially distributed intervals between events.

A comprehensive account of methods for analysing point processes is given by Cox & Lewis (1966). PTDESCRIBE implements many of the test and summary statistics they give and should be used in conjunction with the text for a full discussion of the motivation and context of their use. All equations referred to below are from Cox & Lewis (1966).

The DATA variate may contain either the times at which events occur, the intervals between events, or a sequence of 0's and 1's, with 1's indicating the times of events on an integer time scale. The option REPRESENTATION specifies which of these is used. If REPRESENTATION=time and the process is measured from some time other than zero, the initial time should be given in the parameter START. Otherwise the START time is assumed to be zero. The first interval is taken to lie between the START time and the first event. If the process is observed beyond the last event, the total duration of the process should be given in the parameter LENGTH. Checks are carried out on START, LENGTH and the length of each interval, and the procedure terminates if these are inconsistent. If REPRESENTATION=time, the DATA variate may be restricted, facilitating the analysis of truncated or thinned point processes.

If SAVE is set, time and interval are saved, together with summary interval or second order statistics specified by SELECTION as detailed below. SAVE sets up a pointer, with each element labeled by the name of the relevant statistics saved. For example, if SAVE=clstats, then the intervals between the events will be saved in clstats['interval'].

The option SELECTION can be used to obtain any combination of eight available analyses, with the PRINT and GRAPHICS options controlling the output. The default setting is SELECTION=interval, while SELECTION=all gives all eight analyses. In what follows, the number of events is denoted by *N* and the variate carrying the times of events by *time*. The rate

of a point process is calculated as the reciprocal of the average interval length.

`interval` – plots data and summarises the interval distribution

print:	summary statistics for the interval process.
graph:	times of events; histogram of the intervals between events; histogram of the intervals with bins appropriate for the exponential distribution.
save:	`summary` summary statistics.

`trend` – tests for trend in the process

print:	an N(0,1) test statistic (Ch 3.3 (11)), which is optimal against certain specifications of trend; Bartlett's test for the homogeneity of variance of groups of 3, 8 and 20 contiguous intervals.

`poisson` – tests whether the point process is Poisson

print:	Kolmogorov-Smirnov tests for the empirical distribution function of times of events (Ch 6.2 (27-29, 38)) and for Durbin's order statistic transformation of the intervals (Ch 6.2 (43)); Moran's test against a gamma renewal process for the empirical distribution function (Ch 6.2 (43)); N(0,1) test for trend (see `trend` above) is applied to Durbin's transformed process.
graph:	log survivor function of the interval distribution, compared to the Poisson case (a straight line through the origin with *slope* = −*rate*); plots of the empirical distribution function of times of events and Durbin's order statistics with Kolmogorov-Smirnov bounds.

`icorrelation` – autocorrelations for the interval sequence.

print:	the first ($N/2-1$) end-adjusted autocorrelations (Ch 5.2 (17, 18)) for the interval sequence and their standardization; the end-adjustments are derived using the autocorrelations from `CORRELATE`.
graph:	plot of the autocorrelations of the interval sequence and 95% confidence bounds.
save:	order the order of the autocorrelations, `icorrelation` the autocorrelations of the interval sequence.

`ispectrum` – periodogram for the interval process

print:	the periodogram for the interval process (Ch 5.3 (6, 8)) obtained from `FOURIER` divided by ($2\pi N\sigma^2$), where σ^2 is the variance of the interval lengths; since for the Poisson process the ordinates of the periodogram are iid exponentially distributed r.v.s, the ordinates are also tested as the intervals of a Poisson process as provided for by the `SELECTION` settings `trend` and `poisson` above.
graph:	the periodogram and Poisson level ($\pi/2$) plotted against frequency; plot of the scaled cumulative periodogram with Kolmogorov-Smirnov bounds.
save:	`ifrequency` frequencies at which periodogram is calculated, `ispectrum` interval periodogram.

`cspectrum` – periodogram for the count process

print:	periodogram for the count process (Ch 5.5 (16)) calculated at frequencies $2\pi\omega = 2\pi n/T$, for n=1...2N, T=$time_N - time_1$.
graph:	count periodogram and Poisson level (=2) graphed against

frequency.
save: `cfrequency` frequencies at which periodogram is calculated, `cspectrum` interval periodogram.

`cintensity` – intensity function for the counting process
print: intensity function for the counting process (Ch 5.4(v) (20)) calculated for times `CITAU` × (*j* – 0.5), *j*=1...integer-part($time_N$ / (2×`CITAU`)); if `CITAU` is not set, `PTDESCRIBE` sets it to 0.5 times the average interval length; a preliminary screening precludes an inappropriate setting of `CITAU`.
graph: intensity function with asymptotic 95% confidence intervals for the Poisson level, the intensity for which = rate, plotted against time.
save: `citime` times for which intensity is calculated, `cintensity` intensity function.

`vtcurve` variance-time curve V(*t*) and index of dispersion I(*t*)
print: V(*t*) scaled by 1–*time*/`LENGTH` (Ch 5.4(iii) (12) and following), and I(*t*) (Ch 4.5(3)) calculated for times `VTTAU` × *j*, *j*=1...integer-part(*T*/(*2*×`VTTAU`)); the setting of `VTTAU` is screened to preclude inappropriate values, and if unset is assigned the value 0.5 times the average interval length.
graph: V(*t*) and I(*t*) against time.
save: `vtime` times at which V(*t*) and I(*t*) are calculated, `vtcurve` V(*t*), `dispersion` I(*t*).

Options: `PRINT`, `SELECTION`, `REPRESENTATION`, `GRAPHICS`.
Parameters: `DATA`, `START`, `LENGTH`, `CITAU`, `VTTAU`, `SAVE`.

Method
The procedure tests of whether a point process is a Poisson process and calculates summary statistics in the time and frequency domains for a point process following Cox & Lewis (1966). Most statistics are obtained using `CALCULATE`, with `FOURIER` being used for `ispectrum` and `CORRELATE` for the pre-adjusted autocorrelations.

Action with `RESTRICT`
`DATA` may be restricted only if `REPRESENTATION=time`, in which case only the units not excluded by the restriction are involved in the analysis.

Reference
Cox, D.R. & Lewis, P.A.W. (1966). *The Statistical Analysis of Series of Events*. Methuen, London.

See also
Procedures: `CDESCRIBE`, `DESCRIBE`.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTGRID

Generates a grid of points in a polygon (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NPOINTS = *scalars*	How many points to generate
YSTEP = *scalars*	Spacings to use between columns of the grid
XSTEP = *scalars*	Spacings to use between rows of the grid
YGRID = *variates*	Variates to receive the vertical coordinates of the points in the grid
XGRID = *variates*	Variates to receive the horizontal coordinates of the points in the grid

Description

This procedure generates a grid of points in a polygon specified by the parameters XPOLYGON and YPOLYGON. The size of the grid may be specified in either of two ways. The first method is to specify the total number of points to be generated using the parameter NPOINTS. The value supplied for NPOINTS must be a positive integer. This method will produce a square grid, the number of rows and columns being approximately equal to SQRT(NPOINTS). The second method is to specify the required spacing between rows and columns of the grid using the parameters XSTEP and YSTEP. The values supplied for XSTEP and YSTEP should be on the scale of the coordinates of the polygon. If the parameter NPOINTS is set then any values specified for XSTEP and YSTEP will be ignored. The coordinates of the points which are generated may be saved using the parameters XGRID and YGRID.

Printed output is controlled by the PRINT option. The default setting of summary prints the horizontal and vertical coordinates of the points in the grid under the headings XGRID and YGRID.

Option: PRINT.
Parameters: YPOLYGON, XPOLYGON, NPOINTS, YSTEP, XSTEP, YGRID, XGRID.

Method

A procedure PTCHECKXY is called to check that XPOLYGON and YPOLYGON have identical restrictions. PTBOX is used to calculate the bounding box for the polygon specified by XPOLYGON and YPOLYGON. A grid of points spanning the bounding box is created according to the settings of NPOINTS (appropriately scaled to produce the equivalent density of points on the bounding box), XSTEP and YSTEP. Any points which fall outside the specified polygon are then removed using PTSINPOLYGON.

Action with RESTRICT

If XPOLYGON and YPOLYGON are restricted, only the subset of values specified by the restriction will be included in the calculations.

See also
Procedure: DPOLYGON.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTINTENSITY

Calculates the overall density for a spatial point pattern (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	What to print (summary); default summ

Parameters

Y = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
X = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
DENSITY = *scalars*	Scalars to receive the density of the spatial point patterns, i.e. the number of points per unit area

Description

This procedure takes as input two variates containing the coordinates of a spatial point pattern (specified by the X and Y parameters) and the coordinates of a polygon containing the points (specified using the XPOLYGON and YPOLYGON parameters). The procedure returns the density of the spatial point pattern, which is defined to be the number of points per unit area. The density may be saved in a scalar specified by the parameter DENSITY.

Printed output is controlled by the PRINT option. The default setting of summary prints the density under the heading DENSITY.

Option: PRINT.
Parameters: Y, X, YPOLYGON, XPOLYGON, DENSITY.

Method

A procedure PTCHECKXY is called to check that X and Y have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The area of the polygon is then calculated using PTAREAPOLYGON and then the density is calculated as the number of points divided by the area.

Action with RESTRICT

If X and Y are restricted, only the subset of values specified by the restriction will be included in the calculations. XPOLYGON and YPOLYGON may also be restricted as long as the same restrictions apply to both parameters.

See also

Procedure: PTDESCRIBE.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTKERNEL2D

Performs kernel smoothing of a spatial point pattern (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string tokens*	What to print (grid, monitoring); default grid, moni

Parameters

Y = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
X = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
HZERO = *scalars*	What kernel width to use for each pattern; no default – this parameter must be set
NY = *scalars*	Numbers of rows to use in the grid of kernel density estimates; default 20
NX = *scalars*	Numbers of columns to use in the grid of kernel density estimates; default 20
YGRID = *variates*	Variates to receive the vertical coordinates at which each kernel function has been evaluated
XGRID = *variates*	Variates to receive the horizontal coordinates at which each kernel function has been evaluated
ZGRID = *matrices*	Matrices of dimension NY by NX to receive the grid of density estimates

Description

This procedure performs kernel smoothing of a spatial point pattern using the methods of Diggle (1985) and Berman & Diggle (1989). The kernel density estimate at a point (x, y) represents the intensity of events at that location, and is denoted by kde(x, y). The method implemented in GSplancs uses a quartic kernel function, whereby

$$\text{kde}(x, y) = \sum_i \left((1 - \textit{distance}_i / (2 \times H_0) \right)^2),$$

where the summation is over all the events in the pattern, $\textit{distance}_i$ is the distance from event i to the point (x, y), and H_0 specifies the kernel width. Increasing the value of H_0 produces smoother density estimates.

The data required by the procedure are the coordinates of the points in the pattern (specified using the parameters X and Y) and the coordinates of a polygon within which smoothing is to be performed (specified using the parameters XPOLYGON and YPOLYGON). The kernel width must be specified using the parameter HZERO. The procedure calculates kernel density estimates at a grid of points spanning the specified polygon. The parameters NX and NY specify the numbers of columns and rows to be used in the grid; the default value for both parameters is 20. The output of the procedure is a matrix of kernel density estimates; any elements of the matrix which correspond to points outside the specified polygon will be returned as missing values.

The ZGRID parameter can save the kernel density estimates as a matrix with NY rows and NX columns, with the columns corresponding to values of the horizontal coordinate (x) arranged in ascending order, and the columns corresponding to values of the vertical coordinate (y) in ascending order. (So, for example, if these are plotted using DSURFACE or DSHADE, the YORIENTATION option should be left with its default setting of reverse to reverse the y-

coordinates.)

Printed output is controlled using the PRINT option. The settings available are monitoring (which prints details about the parameter settings for the kernel smoothing process) and grid (which prints the grid of kernel density estimates).

Option: PRINT.
Parameters: Y, X, YPOLYGON, XPOLYGON, HZERO, NY, NX, YGRID, XGRID, ZGRID.

Method

A procedure PTCHECKXY is called to check that X and Y have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The procedure then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. It then calls a procedure PTPASS to call a Fortran program to calculate edge-corrected kernel density estimates for the grid of points spanning the polygon. Finally, the MVINSERT function is used to replace estimates for grid points which lie outside the polygon by missing values.

Action with RESTRICT

If X and Y are restricted, only the subset of values specified by the restriction will be included in the calculations. XPOLYGON and YPOLYGON may also be restricted, as long as the same restrictions apply to both parameters.

References

Berman, M. & Diggle, P.J. (1989). Estimating weighted integrals of the second-order intensity of a spatial point process. *Journal of the Royal Statistical Society, Series B*, **51**, 81-92.

Diggle, P.J. (1985). A kernel method for smoothing point process data. *Applied Statistics*, **34**, 138-147.

See also

Procedures: KERNELDENSITY, MSEKERNEL2D, PTK3D.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTK3D

Performs kernel smoothing of space-time data (D.A. Murray, P.J. Diggle & B.S. Rowlingson).

Option

PRINT = *string token*	Controls printed output (grid, monitoring); default grid

Parameters

Y = *variates*	Vertical coordinates of the spatial point pattern
X = *variates*	Horizontal coordinates of the spatial point pattern
TIMES = *variates*	Times for each event
XGRID = *variates*	The values of x to compute kernel function
YGRID = *variates*	The values of y to compute kernel function
ZGRID = *variates*	The values of z, or time dimension, to compute kernel function
HXY = *scalars*	What quartic kernel width to use in the XY direction
HZ = *scalars*	What quartic kernel width to use in the Z or time direction
GRID = *pointers*	Pointer to matrices containing the kernel smoothed values

Description

This procedure performs kernel smoothing of 3 dimensional, or space-time data. The method implemented uses a quartic kernel function as in the PTKERNEL2D procedure.

The data required by the procedure are the coordinates of the events in the spatial pattern (specified using the parameters X and Y) and the times of the events (specified by TIMES). The XGRID, YGRID and ZGRID parameters specify the 3 dimensions or space-time domain over which to evaluate the kernel smoothing. The kernel width must be specified for the X and Y direction using the HXY parameter, and in the Z (time) direction using the parameter HZ. The procedure calculates kernel density estimates at a grid of points spanning the specified polygon.

The GRID parameter can be used to save the kernel density estimates as a pointer to matrices where each matrix has rows corresponding to values of the horizontal coordinate (x) arranged in ascending order, and the columns corresponding to values of the vertical coordinate (y) in ascending order. Each matrix in the pointer represents a different time-slice and are arranged in ascending order.

Printed output is controlled using the PRINT option. The settings available are monitoring (which prints details about the parameter settings for the kernel smoothing process) and grid (which prints the grids of kernel density estimates).

Option: PRINT.
Parameters: Y1, X1, TIMES, XGRID, YGRID, ZGRID, HXY, HZ, GRID.

Method

A procedure PTCHECKXY is called to check that X, Y and TIMES have identical restrictions. The procedure then calls PTPASS to call a Fortran program to calculate the kernel density estimates for the grid of points spanning the 3 dimensional array.

See also

Procedures: KERNELDENSITY, MSEKERNEL2D, PTKERNEL2D.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTREMOVE

Removes points interactively from a spatial point pattern (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string token*	What to print (summary, monitoring); default summ, moni
WINDOW = *scalar*	Which graphics window to use for the plot; default 1

Parameters

OLDY = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
OLDX = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
NEWY = *variates*	Variates to receive the vertical coordinates of the original points minus the deleted points of each pattern
NEWX = *variates*	Variates to receive the horizontal coordinates of the original points minus the deleted points of each pattern

Description

PTREMOVE uses the DREAD directive to delete points from a spatial point pattern. The coordinates of the existing points must be supplied using the parameters OLDX and OLDY. These points will be plotted on the current graphics device using DPTMAP with a pen setting of SYMBOLS=1. The WINDOW option may be used to specify the graphics window to use for the plot.

The operation of DREAD may vary slightly from one system to another. The Users' Note supplied with Genstat explains how to read points and terminate input on specific devices. The usual method for reading points is to click the left mouse button at the required position. The usual way to terminate input is to click the right mouse button. The points read using DREAD will be echoed using a pen setting of SYMBOLS=2. The coordinates of the new spatial point pattern containing the original points minus any points which have been deleted may be saved using the parameters NEWX and NEWY.

Printed output is controlled using the PRINT option. The settings available are monitoring (which prints the coordinates of the points to be deleted) and summary (which prints the coordinates of the new pattern consisting of the original points minus any that have been deleted under the headings NEWX and NEWY). The default setting is for both monitoring and summary.

Options: PRINT, WINDOW.
Parameters: OLDY, OLDX, NEWY, NEWX.

Method

A procedure PTCHECKXY is called to check that OLDX and OLDY have identical restrictions. DPTMAP is then used to draw a map of the original point pattern. The DREAD directive is used to read the coordinates of points to be deleted. Finally, the coordinates for the deleted points are removed from the original points using the SUBSET procedure and the coordinates of the undeleted points are stored in new variates.

Action with RESTRICT

If OLDX and OLDY are restricted, only the subset of values specified by the restriction will be included in the calculations.

See also
Procedure: DPTREAD, DRPOLYGON.
Genstat Reference Manual 1 Summary section on: Graphics, Spatial statistics.

PTROTATE

Rotates a point pattern (W. van den Berg).

Options

ANGLE = *scalar*	Angle, in degrees over which the point pattern is to be rotated; no default – must be set
HUB = *string token*	Whether the point pattern is to be rotated around the origin or around the centroid (`origin`, `centroid`); default `orig`

Parameters

OLDY = *variates*	Vertical coordinates of each spatial point pattern
OLDX = *variates*	Horizontal coordinates of each spatial point pattern
NEWY = *variates*	Save the vertical coordinates of the rotated point patterns; if this unset, these replace the original values in OLDY
NEWX = *variates*	Save the horizontal coordinates of the rotated point patterns; if this unset, these replace the original values in OLDX
ROTATION = *matrices*	Save the rotation matrices

Description

PTROTATE rotates a point pattern. The vertical and horizontal coordinates must be supplied as variates using the parameters OLDY and OLDX. The angle over which the point pattern must be rotated must be supplied, in degrees, by the ANGLE option. When a positive angle is supplied the rotation is clockwise. A negative number results in a counter clockwise rotation. By default the rotation is around the origin, but you can set option HUB=centroid to perform the rotation around the centroid of the point pattern.

The vertical coordinates of the rotated pattern can be saved, in a variate, using the NEWY parameter; if this is unset, the rotated pattern replaces the original pattern in OLDY. Similarly, the horizontal coordinates can be saved using the NEWX parameter, or in the original variate supplied by OLDX. The rotation matrix can be saved using the ROTATION parameter.

Options: ANGLE, HUB.
Parameters: OLDY, OLDX, NEWY, NEWX, ROTATION.

Method

A matrix is formed with 2 columns consisting of the coordinates supplied to parameters OLDY and OLDX. This matrix is post-multiplied by a 2×2 matrix, containing COS(ANGLE) on the diagonal, and plus and minus SIN(ANGLE) in the upper-right and lower-left cells.

Action with RESTRICT

Any restrictions on OLDX and OLDY are removed.

See also

Directive: ROTATE.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

PTSINPOLYGON

Returns points inside or outside a polygon (M.A. Mugglestone, S.A. Harding, B.Y.Y. Lee, P.J. Diggle & B.S. Rowlingson).

Options

PRINT = *string token*	What to print (summary); default summ
METHOD = *string token*	Whether to select points inside or outside the polygon (inside, outside); default insi

Parameters

OLDY = *variates*	Vertical coordinates of each spatial point pattern; no default – this parameter must be set
OLDX = *variates*	Horizontal coordinates of each spatial point pattern; no default – this parameter must be set
YPOLYGON = *variates*	Vertical coordinates of each polygon; no default – this parameter must be set
XPOLYGON = *variates*	Horizontal coordinates of each polygon; no default – this parameter must be set
NEWY = *variates*	Variates to receive the vertical coordinates of points inside (or outside) the polygons
NEWX = *variates*	Variates to receive the horizontal coordinates of points inside (or outside) the polygons

Description

This procedure takes as input two variates containing the coordinates of a spatial point pattern (specified by the OLDX and OLDY parameters) and another two variates containing the coordinates of the polygon (specified by the XPOLYGON and YPOLYGON parameters). The output of the procedure depends upon the setting of the METHOD option. The default setting of inside returns the events of the spatial point pattern which lie inside the polygon. Setting the option to outside returns the events which lie outside the polygon. Note that any events which lie on the boundary of the polygon will be regarded as being outside the polygon.

The coordinates of the points satisfying the condition implied by the setting of METHOD can be saved using the parameters NEWX and NEWY. If no points satisfy the condition, then the structures specified by NEWX and NEWY will be declared as variates of undefined length and with no values.

Printed output is controlled by the PRINT option. The default setting of summary prints the coordinates of the points satisfying the condition implied by the setting of METHOD under the headings NEWX and NEWY.

Options: PRINT, METHOD.
Parameters: OLDY, OLDX, YPOLYGON, XPOLYGON, NEWY, NEWX.

Method

A procedure PTCHECKXY is called to check that OLDX and OLDY have identical restrictions. A similar check is made on XPOLYGON and YPOLYGON. The procedure then calls PTCLOSEPOLYGON to close the polygon specified by XPOLYGON and YPOLYGON. It then calls a procedure PTPASS to call a Fortran program to determine which of the points in OLDX and OLDY are inside the polygon. Finally, the RESTRICT directive is used to select the points which are inside/outside the polygon, according to the setting of the METHOD option.

Action with `RESTRICT`
If `OLDX` and `OLDY` are restricted, only the subset of values specified by the restriction will be included in the calculations. `XPOLYGON` and `YPOLYGON` may also be restricted, as long as the same restrictions apply to both parameters.

See also
Procedures: `DPOLYGON`, `PTAREAPOLYGON`, `PTCLOSEPOLYGON`, `INSIDE`.
Genstat Reference Manual 1 Summary section on: Spatial statistics.

QBESTGENOTYPES

Sorts individuals of a segregating population by their genetic similarity with a defined target genotype, using the identity by descent (IBD) information at QTL positions for one or more traits (M. Malosetti & F.A. van Eeuwijk).

Options

PRINT = *string tokens*	What to print (summary); default summ
PLOT = *string tokens*	What to plot (haplotypes); default hapl
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL); default F2
IBDWINDOW = *scalar*	Size of the window around the QTL position to use to construct the haplotypes; default 10
TRAITS = *text*	Names of the traits whose QTL information is to be used; default is to use all the traits
SELECTION = *variate*	Indicator variate with values defining whether each trait should be maximized (1), minimized (−1) or remain unchanged (0); if unset, the default is to maximize every trait
%BESTGENOTYPES = *scalar*	Specifies the percentage of the best genotypes to display in the output and plots; default 10

Parameters

GENFILENAME = *texts*	Name of a Flapjack genotype file
MAPFILENAME = *texts*	Name of a Flapjack map file
FJQTLFILENAME = *texts*	Name of a file to supply the QTL results
QTRAITS = *texts*	Names of the traits affected by each QTL
QCHROMOSOMES = *factors*	Factor defining the linkage group of each QTL
QPOSITIONS = *variates*	Position of each QTL within the linkage group
QNAMES = *texts*	Name of each QTL
QEFFECTS = *variates*	Individual QTL effects
QBESTSAVE = *pointers*	Saves similarities with the target genotype, and their ranks, across and per trait

Description

QBESTGENOTYPE is a procedure for post processing of QTL mapping results. It uses the identity by descent (IBD) information of genotypes at QTL positions to define genetic similarities with a target genotype. It can deal with QTL information for one or several traits simultaneously.

By default, QBESTGENOTYPE prints the results, but you can suppress this by setting option PRINT=*. It also produces a plot of the best genotypes, and this can be suppressed by setting option PLOT=*. The %BESTGENOTYPES specifies the percentage of genotypes, at the top of the ranking, that are to be displayed or plotted (default 10%).

The POPULATIONTYPE option specifies the population type (default F2), and the IBDWINDOW option defines the size of the window around the QTL position in which IBD probabilities are estimated (default 10 cM). The traits from which QTL information will be used are specified by the TRAITS option. The direction of selection for each trait is defined in a variate supplied by the SELECTION option. This contains the value one if the aim is to increase the trait, minus one if it should be decreased, or zero if it should not change.

The GENFILENAME and MAPFILENAME parameters must supply the names of Flapjack files with the marker scores and map information, respectively. The FJQTLFILENAME parameter can be used to supply the QTL information as a Flapjack QTL file (for example saved by the QFLAPJACK procedure). Alternatively the QTL information can be supplied (in vectors with equal lengths) by the following parameters: QTRAITS (names of the traits), QNAMES (names of

the QTLs), QCHROMOSOMES (chromosome locations), QPOSITIONS (positions within the chromosomes) and QEFFECTS (QTL effects).

The QBESTSAVE parameter can save the results of the selection, in a pointer with 3 elements. The first, labelled 'Genotypes', saves the names of the genotypes. The second, labelled 'Ranking', is itself a pointer with an element 'All traits' for the combined ranking, and then an element for each individual trait (labelled by the trait name). The third, labelled 'Similarity', stores the similarities in a pointer like that used for the ranks.

Options: PRINT, PLOT, POPULATIONTYPE, IBDWINDOW, TRAITS, SELECTION, %BESTGENOTYPES.

Parameters: GENFILENAME, MAPFILENAME, FJQTLFILENAME, QTRAITS, QNAMES, QCHROMOSOMES, QPOSITIONS, QEFFECTS, QBESTSAVE.

Method

The QIBDPROBABILITIES procedure is used to calculate IBD probabilities around the QTL positions. A target genotype is defined combining the individual QTL effects and selection objectives per trait. The FSIMILARITY directive is used to estimate a similarity of each genotype to the target per trait, and an overall similarity across traits is obtained as the average similarity across traits (i.e. traits are equally weighted).

Action with RESTRICT

Restrictions are not allowed.

See also

Procedures: QFLAPJACK, QIBDPROBABILITIES, QREPORT.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QCANDIDATES

Selects QTLs on the basis of a test statistic profile along the genome (M.P. Boer & J.T.N.M. Thissen).

Options

PRINT = *string token*	What to print (summary); default summ
THRESHOLD = *scalar*	Threshold for the test statistic; default 0
QTLWINDOW = *scalar*	Minimum distance in cM between two peaks to be selected as two QTLs; default 10

Parameters

STATISTICS = *variates*	Test statistic along the genome; must be set
CHROMOSOMES = *factors*	Chromosome for each locus; must be set
POSITIONS = *variates*	Position on the chromosome for each locus; must be set
IDLOCI = *texts*	Labels for the loci
QTLCANDIDATES = variates	Saves the index numbers of the selected QTLs

Description

QCANDIDATES selects the peaks in the test statistic profile along the genome. These profiles, calculated with the procedures QSQTLSCAN or QMQTLSCAN, must be supplied by the STATISTICS parameter. Information about chromosome number and position on the chromosome must be provided by the CHROMOSOMES and POSITIONS parameters respectively. Names to identify the loci can be specified using the IDLOCI parameter.

The selection depends on the THRESHOLD and SEPARATION options. Positions with a STATISTICS score greater than THRESHOLD are eligible for selection. The maximum STATISTICS score along the genome is selected as the first QTL. Then futher STATISTICS scores are selected, one at a time, at each stage taking the maximum score of those with a distance of at least QTLWINDOW from the QTLs that have already been selected. The search continues until no further statistics scores are found above THRESHOLD and with a distance of QTLWINDOW or more from the other selected QTLs.

Options: PRINT, THRESHOLD, QTLWINDOW.
Parameters: STATISTICS, CHROMOSOMES, POSITIONS, IDLOCI, QTLCANDIDATES.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedures: QMQTLSCAN, QSQTLSCAN.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QCOCHRAN

Performs Cochran's *Q* test for differences between related samples (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (test); default test
METHOD = *string token*	Form of the test (exact, chisquare); default exac for small samples, otherwise chis
GROUPS = *factor*	Defines the groups if there only one variable supplied for the DATA
STATISTIC = *scalar*	Scalar to save the Q value
PROBABILITY = *scalar*	Scalar to save the probability for the Q Test
MAXTIME = *scalar*	Defines a limit for the maximum time for calculating the exact test; default * i.e. no limit.

Parameter

DATA = *variates*	List of related samples, or variate containing all the samples (the GROUPS option must then be set to indicate the variable recorded in each unit belongs)

Description

Cochran's *Q* test is an extension to the McNemar test for related samples that provides a method for testing for differences between three or more matched sets of frequencies or proportions. The matching samples can be based on *k* characteristics of *N* individuals that are associated with the response. Alternatively *N* individuals may be observed under *k* different treatments or conditions (e.g. different questions or one question at different times).

The data must be supplied as dichotomous variables containing 0 to represent failure (or absence), and 1 to represent success (or presence). The variables can be stored in separate variates and the DATA parameter set to list them all. Alternatively, all the data can be stored in a single variate, and the GROUPS option set to a factor to indicate which variable is recorded in each unit of the variate. (QCOCHRAN then assumes that the individuals are recorded in the same order for each variable.)

In its original form, the test leads to a chi-square test (see the Method section). However, this may be inaccurate when there are small numbers of subjects or samples. Consequently QCOCHRAN also provides an exact probability (based on the exact distribution of *Q* under a permutation model). The form of the test can be set to either chi-square or exact by using the METHOD option. The default is to use the exact test if the number of values in the samples is less than 4 and the product of this value with the number of samples is less than 24, otherwise the chi-square method is used. The *Q* statistic can be saved using the STATISTIC parameter, and the probability can be saved using the PROBABILITY parameter.

Although QCOCHRAN uses an efficient algorithm for calculating the exact probability, the time and memory required for this calculation can become impractible as the number of samples and values increases. Therefore, for large problems, the chi-square approximation should be used. However, for the exact calculation the MAXTIME option can be used to supply the maximum amount of time (in seconds) that will be used to calculate the exact probabilty. If this is time exceeded, the computation is terminated.

The PRINT option controls printed output, with settings:

test	the *Q* value and probability (the default).

Options: PRINT, METHOD, GROUPS, STATISTIC, PROBABILITY, MAXTIME.
Parameter: DATA.

Method

The Cochran Q Test is calculated by:

$$Q = (k \times (k - 1) \times \sum_{j=1...k}\{(T_j - Tbar)^2\}) / (k \times \sum_{j=1...k}\{u_i\} - \sum_{j=1...k}\{u_i^2\})$$

where k is the number of samples, T_j is the sum of 1's in the jth column, *Tbar* is the mean of the T_j's, and u_i is the number of 1's in the ith row. Under the null hypothesis this has an approximate chi-square distribution with $(k-1)$ degrees of freedom.

The exact test is calculated using the permutation method of Patil (1975).

Action with `RESTRICT`

If a parameter is restricted the statistics will be calculated using only those units included in the restriction.

References

Patil K,D. (1975). Cochran's Q Test: exact distribution. *Journal of the American Statistical Association*, **70**, 186-189.

Siegel S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Procedures: `CATRENDTEST`, `MCNEMAR`.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

QDESCRIBE

Calculates descriptive statistics of molecular markers (M.P. Boer & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (chromosomes, genome); default chro
DISTANCE = *scalar*	Distance between chromosomes (for plotting purposes); default 10

Parameters

CHROMOSOMES = *factors*	Chromosome for each locus; must be set
POSITIONS = *variates*	Position on the chromosome for each locus; must be set
IDLOCI = *texts*	Labels for the loci
CUMPOSITIONS = *variates*	Saves the cumulative positions of the loci along the genome
NLOCI = *variates*	Saves the number of loci on each chromosome
FIRST = *variates*	Saves the index number of the first locus of each chromosome
LAST = *variates*	Saves the index number of the last locus of each chromosome
LENGTHS = *variates*	Saves the lengths of the chromosomes
MIDDLEPOSITIONS = *variates*	Saves the middle positions of the chromosomes (as cumulative positions)
SEPARATION = *variates*	Saves the positions of the gaps between chromosomes (as cumulative positions)
GENOMELENGTH = *scalars*	Saves the length of the genome
TOTLENGTH = *scalars*	Saves the total length of the genome, including added gaps between chomosomes

Description

QDESCRIBE is mainly used as an auxiliary procedure for plotting the QTL scan profiles, calculated by the QSQTLSCAN and QMQTLSCAN procedures, and plotted by the DQSQTLSCAN and DQMQTLSCAN procedures. QDESCRIBE calculates statistics of the genome and chromosomes on the basis of chromosome identifier and locus position. The chromosome identifiers of the loci are specified by the CHROMOSOMES parameter, and the positions of the loci on the chromosomes are specified by the POSITIONS parameter. If the loci have names, these can be specified by the IDLOCI parameter. The calculated statistics can be saved using the parameters CUMPOSITIONS, NLOCI, FIRST, LAST, LENGTHS, MIDDLEPOSITIONS, SEPARATION, GENOMELENGTH and TOTLENGTH.

Options: PRINT, DISTANCE.

Parameters: CHROMOSOMES, POSITIONS, IDLOCI, CUMPOSITIONS, NLOCI, FIRST, LAST, LENGTHS, MIDDLEPOSITIONS, SEPARATION, GENOMELENGTH, TOTLENGTH.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedures: DQMQTLSCAN, DQSQTLSCAN, QMQTLSCAN, QSQTLSCAN.

Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

QDISCRIMINATE

Performs quadratic discrimination between groups i.e. allowing for different variance-covariance matrices (D.B. Baird).

Options

PRINT = *string tokens*	Printed output from the analysis (allocation, counts, distance, probabilities, specificity, summary, table, validation, vcovariance); default spec, summ, vali
VALIDATIONMETHOD = *string token*	Validation method to use to calculate error rates (bootstrap, crossvalidation, jackknife, prediction); default cros
NSIMULATIONS = *scalar*	Number of bootstraps or cross-validation sets; default 50
NCROSSVALIDATIONGROUPS = *scalar*	Number of groups for cross-validation, default 10

Parameters

DATA = *pointers*	Each pointer contains a training set of variates to be used to form a quadratic discrimination
GROUPS = *factors*	Define groupings for the units in each training set
PRIORPROBABILITIES = *variates*	Prior probabilities of group membership; default * i.e. equal
SEED = *scalars*	Seed for the random numbers used in bootstrapping or cross-validation; default 0 continues from the previous generation or (if none) initializes the seed automatically
ERRORRATE = *scalars*	Saves the validation error rate
SPECIFICITY = *matrices*	Saves the specificity table
ALLOCATION = *factors*	Saves the groups allocated by the discriminant rule
PROBABILITIES = *matrices* or *pointers*	Save posterior probabilities of membership of the groups (in the columns of a matrix or the variates in a pointer) for the units in the training set (in the rows)

Description

QDISCRIMINATE performs a quadratic discrimination analysis to identify members of a set of groups using their observations on a set of variates. The quadratic discrimination rule assumes that the values of the variates within each group are distributed with a multi-variate Normal distribution, and that the variance-covariance matrix of the distributions are different for each group. This differs from the more familiar linear discriminant analysis, performed by procedure DISCRIMINATE, where the groups are assumed to have the same variance-covariance matrix.

The variates to be used to discriminate between the groups are specified in a pointer by the DATA parameter, and the membership of the groups is specified in a factor by the GROUPS parameter. The non-missing units of the GROUPS factor provide a training set to estimate the discriminant rule. Units that you would like to allocate to groups using the discriminant rule should be included in the data set with missing values in the GROUPS factor.

You can specify prior probabilities for the groups using the PRIORPROBABILITIES option; by default the groups are all assumed to be equally likely. You can use this to allow for unequal costs of mis-allocation by weighting the prior probabilities like this:

```
PRIORPROBABILITIES = Cost * Prior / SUM(Cost * Prior)
```

where Cost is a variate defining the cost of mis-allocation for each group.

Printed output is controlled by the option PRINT, with settings:

allocation	the allocated group for each unit,
counts	number of units in each group with a complete set of observations,
distance	generalized pairwise distance between group means,
probabilities	the posterior probability of being allocated to each group,
specificity	specificity of allocation (i.e. the proportion of each group that is assigned correctly),
summary	summary of the model fitting,
table	table of counts of training units allocated to each group,
validation	the error rate, and
vcovariance	variance-covariance matrices for the groups

The default is PRINT=spec,summ,vali.

The VALIDATIONMETHOD option specifies the validation method, with settings for prediction, cross-validation, jackknife and bootstrap. Prediction calculates the error rate as the proportion of the training set that were misallocated. Cross-validation works by randomly splitting the units into a number of groups specified by the NCROSSVALIDATIONGROUPS option (default 10). It then omits each of the groups, in turn, and predicts how the the omitted units are allocated to the discrimination groups. Jackknifing leaves the units out one at a time, and uses the rest of the data to predict the group of the omitted unit. The bootstrap method works by drawing a bootstrap sample of units (a random sample of units with replacement of the same size as the original sample), and predicting the units that are not present in the random sample. The resulting bootstrap error rate is then calculated as a weighted average of the error rate of the omitted observations and the predictive error rate of the bootstrap sample. The weights used are 0.632 and 0.368 respectively, and so this is known as the *632 rule*.

The NSIMULATIONS option sets the number of simulations for cross-validation or bootstrapping; default 50.

The SEED parameter provides the seed for the random numbers used for the randomizations during in the simulations. The default value of 0 continues an existing sequence of random numbers, if none have been used in the current Genstat job, it initializes the seed automatically using the computer clock.

The ERRORRATE parameter can save the validation error rates. The SPECIFICITY parameter can save the proportion of each group that is assigned correctly. The ALLOCATION parameter can save the assigned groups, and the PROBABILITIES parameter can save the posterior probabilities of the groups.

Options: PRINT, VALIDATIONMETHOD, NSIMULATIONS, NCROSSVALIDATIONGROUPS.
Parameters: DATA, GROUPS, PRIORPROBABILITIES, SEED, ERRORRATE, SPECIFICITY, ALLOCATION, PROBABILITIES.

Method

The FSSPM directive is used to calculate the variance-covariance matrices of the groups. The posterior probability of belonging to each group are then calculated for each unit, and its membership is assigned to the most likely group. For more details, see e.g. Hastie *et al*. (2001) or McLachlan (1992).

Action with RESTRICT

The input variates and factor may be restricted (but any restrictions must be identical). The restricted units are omitted from the analysis.

References

Hastie, T., Tibshirani, R. & Friedman J. (2001). *The Elements of Statistical Learning*. Springer, New York.

McLachlan, G.J. (1992). *Discriminant Analysis and Statistical Pattern Recognition*. Wiley, Hoboken, New Jersey.

See also

Directive: CVA.

Procedures: CVAPLOT, DBIPLOT, DISCRIMINATE, SDISCRIMINATE.

Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

QEIGENANALYSIS

Uses principal components analysis and the Tracy-Widom statistic to find the number of significant principal components to represent a set of variables (M. Malosetti & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, scores); default summ
NROOTS = *scalar*	Number of principal components to retain; default saves the significant components
PLOT = *string tokens*	What to plot (eigenvalues, %variance); default eige, %var
PROBABILITY = *scalar*	Specifies the significance level; default 0.05
SCALING = *string token*	Whether to scale the principal component scores by the square roots of their singular values (singularvalues, none); default none
STANDARDIZE = *string token*	How to standardize the DATA variates (frequency, none); default freq
TITLE = *text*	General title for the plots

Parameters

DATA = *pointers*	Data variates; must be set
SCORES = *pointers*	Pointer of variates to store the scores of the significant axes for each set of DATA variates
EVALUES = *variates*	Saves the eigenvalues of the significant principal components
NEFFECTIVE = *scalars*	Saves the effective number of columns of the marker data matrix
%VARIANCE = *variates*	Saves the percentage variances explained by the significant principal components
CUM%VARIANCE = *variates*	Saves the cumulative percentage variances explained by the significant principal components

Description

QEIGENANALYSIS performs a principal component analysis on a set of variables, supplied by the DATA parameter, and determines the number of significant components according to the significance level specified by the PROBABILITY option (default 0.05). You can set the number of principal component axes to retain by using the NROOTS option; if this is unset, the significant components are saved. By default the variates are standardized before doing the analysis, but you can set option STANDARDIZE=none to suppress this. The scores of the significant principal components can be saved, in a pointer of variates, using the SCORES parameter. You can set option SCALING=singularvalues to scale the scores by the square roots of their singular values; by default they are not scaled.

The PRINT option controls printed output, with settings:

summary	to print the Tracy-Widom statistics of the significant principal components,
scores	to print the scores of the significant principal components.

The default is PRINT=summary.

The PLOT option selects the graphs to plot, with settings:

eigenvalues	plots eigenvalues against the number of principal components, and
%variance	plots the percentage variance explained and cumulative

percentage variance explained, against the number of principal components.

The default is to plot both graphs. The TITLE option can supply a title for the graphs.

The EVALUES parameter can be used to save the eigenvalues, and the %VARIANCE and CUM%VARIANCE parameters can save the percentage variances and cumulative percentage variances explained by the significant principal components. The NEFFECTIVE parameter can save the effective number of columns of the marker data matrix, estimated as described by Patterson *et al.* (2006).

Options: PRINT, NROOTS, PLOT, PROBABILITY, SCALING, STANDARDIZE, TITLE.
Parameters: DATA, SCORES, EVALUES, NEFFECTIVE, %VARIANCE, CUM%VARIANCE.

Method

QEIGENANALYSIS implements the method described by Patterson *et al*. (2006). It uses the SVD directive to perform the principal components analysis, and iteratively calculates the Tracy-Widom statistic for the principal components until one is found to be non-significant. Missing values in the marker score data of each marker are replaced by the means of the marker scores of that marker. The significance of the principal components is assessed using tabulated values of the Tracy-Widom density function.

Action with RESTRICT

Restrictions are not allowed.

Reference

Patterson, N., Price, A.L., Reich, D. (2006). Population structure and eigenanalysis. *PLoS Genetics*, **2**, e190. doi:10.1371/journal.pgen.0020190

See also

Procedures: QLDDECAY, QMASSOCIATION, QSASSOCIATION.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QEXPORT

Exports genotypic data for QTL analysis (D.A. Murray).

Options

OUTFILENAME = *text*	Name of the file to receive the data
MAPFILENAME = *text*	Name of the associated map file for Flapjack or MapQTL(R)
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP, AMP); must be set
NGENERATIONS = *scalar*	Number of generations for a RIL population
NAME = *text*	Name for the header in a .loc file
MISSING = *text*	Character to represent a missing genotype in Flapjack or R/QTL format; default '-'
SEPARATOR = *text*	Character to separate data values in Flapjack format; default separates them by tabs
ASEPARATOR = *text*	Character to separate allele values in Flapjack format; default '/'
FJROWS = *string token*	Specifies whether the genotypes or markers are to be stored on the rows in Flapjack format (genotypes, markers); default geno

Parameters

MKSCORES = *pointers*	Genotype codes for each marker
CHROMOSOMES= *factors*	Linkage groups for the markers
POSITIONS = *variates*	Positions within the linkage groups of markers
MKNAMES = *texts*	Marker names
MKSETS = *factors*	Marker sets
IDMGENOTYPES = *texts*	Labels for genotypes
PARENTS = *pointers*	Parent information
IDPARENTS = *texts*	Labels used to identify the parents

Description

QEXPORT saves genotypic map data for QTL analysis. The data are saved to the file specified by the OUTFILENAME option. The format of the file is specified by the file extension, and can be either a Flapjack text genotype file (.txt), a MapQTL(R) Locus genotype file (.loc) or an R/QTL separate genotype file (.csv). If a Flapjack genotype file or a MapQTL(R) Locus genotype file name is supplied, the associated map information can be saved by setting the MAPFILENAME option to a file name with the extension .txt for Flapjack or .map for MapQTL(R). QEXPORT can thus be used to save data in Flapjack format to use with the QIBDPROBABILITIES procedure.

The type of population must be specified using the POPULATIONTYPE option. The genotypic data can be exported for F2, first generation backcross (BC1), recombinant inbred lines (RIL) and DH1 (double-haploid) populations to any of the file types. The BCxSy (backcross inbred lines), CP (cross pollinator) and AMP (association mapping) populations can be exported only using the Flapjack format. If a RIL population is being exported to MapQTL(R), the number of generations should be specified using the NGENERATIONS option. Also, for exporting to MapQTL(R), the NAME option allows you to include a name for the population (which must not contain spaces).

The marker scores should be supplied in a pointer to a set of factors using the MKSCORES parameter. Each factor within the pointer should contain data for a marker, where the same factor labels are supplied in the same order. For the BC1, DH1, F2, RIL, BCxSy and CP populations, the

parent information can be supplied in a pointer to a set of texts using the PARENTS parameter. Note that the PARENTS parameter must be set for a CP population. Each text should contain the parent allele, where the position within the pointer determines the parent: for example, the first text represents parent 1, the second text parent 2 and so on. For the BC1, DH1, F2, RIL and BCxSy populations, if the PARENTS parameter is not set, then the parent information is automatically generated where parent 1 is allocated allele 1 and parent 2 is allocated allele 2. The labels for the parents can be supplied in a text using the IDPARENTS parameter.

By default, in Flapjack or R/QTL files, missing alleles are represented using the '-' character, but an alternative can be supplied using the MISSING option. In Flapjack genotype files, the separator used between marker genotype scores can be supplied using the SEPARATOR option, and the separator used between alleles using the ASEPARATOR option. For the Flapjack genotype format, the FJROWS option indicates whether the genotypes or markers are stored in the rows of the file; by default the genotypes are in the rows.

The linkage groups for each marker are supplied in a factor by the CHROMOSOMES parameter. The names of the markers are supplied in a text using the MKNAMES parameter, and the marker positions are supplied in a variate using the POSITIONS parameter. For the .csv file format, a grouping factor identifying marker sets can be supplied using the MKSETS parameter.

The genotype labels to be stored in a Flapjack or R/QTL file can be specified using the IDMGENOTYPES parameter. If this parameter is not set, the labels will be generated automatically using the values 1 to *n*, where *n* is the number of genotypes.

Options: OUTFILENAME, MAPFILENAME, POPULATIONTYPE, NGENERATIONS, NAME. MISSING, SEPARATOR, ASEPARATOR, FJROWS.

Parameters: MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, MKSETS, IDMGENOTYPES, PARENTS, IDPARENTS.

Method

The .csv file format uses an extended version of the R/QTL comma-delimited separate file for genotype data (.csvsr and .csvs), where there is an optional column for marker sets. For exporting large data sets to .csv format, the procedure uses the Dataload.dll library. In the .csvsr file format, the first row specifies the genotype or id, and there must be a column of data associated with the phenotypic data with exactly the same information. The first cell should be the name of an identifier for the genotype id, and the first row in columns 2 and 3 should be blank. Also, if marker sets are included in the file, the first row of column 4 should be left blank. Starting from row 2 in the file, the first column gives the marker names, the second column gives the linkage group for each marker, and the third column gives the positions of the markers within the linkage groups. The fourth column can be used to contain the marker sets. The remaining columns give the marker genotypes.

Action with RESTRICT

All restrictions are ignored.

See also

Procedures: EXPORT, QIMPORT, QIBDPROBABILITIES.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QFACTOR

Allows the user to decide to convert texts or variates to factors (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (replication, summary); default summ
MAXCATEGORY = *scalar*	Maximum number of distinct values that a VECTOR may contain if it is to be converted; default 10
QUERY = *string token*	Whether to ask the user if each VECTOR with no more than MAXCATEGORY distinct values is to be converted

Parameter

VECTOR = *variates* or *texts*	Vectors to be converted into factors

Description

The QFACTOR procedure provides a convenient way of converting variates or texts to factors. The variates or texts are specified by the VECTOR parameter. The MAXCATEGORY option defines the maximum number of distinct values that each VECTOR may contain if it is to be converted (default 10). The QUERY option determines whether to ask the user whether to convert each VECTOR that has no more than MAXCATEGORY categories; with the default setting, QUERY=no, they are all converted.

The PRINT option controls printed output, with settings:

replication	to print the replication of the levels of each converted VECTOR;
summary	to print the number of values and levels of each converted VECTOR or, alternatively, a comment to say that the VECTOR has not been converted.

Options: PRINT, MAXCATEGORY, QUERY.
Parameter: VECTOR.

Method

The QUESTION procedure is used to obtain the user's decision, and the GROUPS directive is used to convert the variate or text to a factor.

See also

Directive: QDIALOG.
Procedures: QLIST, QUESTION.
Genstat Reference Manual 1 Summary sections on: Program control, Calculations and manipulation.

QFLAPJACK

Creates a Flapjack project file from genotypic and phenotypic data (D.A. Murray).

Options

WORKDIRECTORY = *text*	Working directory to use for files; default current Genstat working directory
FJPATH = *text*	Path specifying the location of Flapjack; by default QFLAPJACK searches for a version of Flapjack installed within C:\program files (x86)\Flapjack or C:\program files\Flapjack
DECIMALSYMBOL = *string token*	Controls whether to use the locale (automatic) or English (dot) representation of decimal marks (automatic, dot); default auto

Parameters

FJFILENAME = *texts*	Name of the Flapjack project file to create
TRAITS = *pointers*	Pointer to variates containing the phenotypic trait data
GENOTYPES = *factors*	Genotype factor associated with the traits
ENVIRONMENTS = *factors*	Environment factor
GENFILENAME = *texts*	Name of a Flapjack genotype file
MAPFILENAME = *texts*	Name of a Flapjack map file
FJTRAITFILENAME = *texts*	Name of a file to supply the trait data, or to save them if the TRAITS and GENOTYPES parameters are also set
FJQTLFILENAME = *texts*	Name of a file to supply the QTL results, or to save them if the QSAVE parameter is also set
QSAVE = *pointers*	Information and results saved from an earlier QTL analysis

Description

Flapjack is a tool for graphical genotyping and haplotype visualization that can routinely handle the large data volumes generated by high throughput SNP and comparable genotyping technologies. Its visualizations are rendered in real-time allowing for rapid navigation and comparisons between lines, markers and chromosomes. QFLAPJACK can be used to create a Flapjack project file containing genotypic and phenotypic data along with QTL results. The name of the Flapjack project file is specified by the FJFILENAME parameter.

To use QFLAPJACK, the Flapjack software must be installed on the current system. The location of the Flapjack to use to create the project file is specified using the FJPATH option. If FJPATH is not specified, QFLAPJACK searches for a version of Flapjack installed within the directories

```
C:\program files (x86)\Flapjack
```

or

```
C:\program files\Flapjack
```

The genotypic marker and map data must be supplied in Flapjack genotype and map text files. The name of the genotype file is specified using the GENFILENAME parameter, and the name of the map file is specified using the MAPFILENAME parameter. These files can be created from Genstat data structures using the QEXPORT procedure.

Phenotypic data can be added to the project file from a text file whose the name is specified by the FJTRAITFILENAME parameter. The data within the file should be tab separated, with the first column containing the genotype labels, and the remaining columns containing the traits. Alternatively, you can use the TRAITS parameter to supply the trait data, in a pointer to a set of

variates, and the GENOTYPES parameter to supply the associated genotype factor. For multi-environment trials, you should also use the ENVIRONMENTS parameter to supply a factor to identify the environments. If you specify FJTRAITFILENAME as well as TRAITS and GENOTYPES, the phenotypic data will be saved in the file whose name is specified by FJTRAITFILENAME; this can then be used in a later QFLAPJACK command.

The results from a QTL analysis can be included within the Flapjack project file by using the FJQTLFILENAME parameter to supply a text file containing the results. Details of the file layout for importing supplementary QTL data within a Flapjack project are given within the Flapjack application. Alternatively, the QSESTIMATE and QMESTIMATE procedures allow you to save results from a linkage analysis in a pointer, using their QSAVE parameters. This can then be used as the setting of the QSAVE parameter of QFLAPJACK to supply the QTL data. If you specify FJQTLFILENAME as well as QSAVE, the QTL data will be saved in the file whose name is specified by FJQTLFILENAME; this can then be used in a later QFLAPJACK command.

By default, the working directory will be the current directory. However, an alternative directory can be supplied using the WORKDIRECTORY option.

The DECIMALSYMBOL option controls the type of decimal marks to use in the Flapjack project file, with the following settings:

automatic	uses the locale settings, and
dot	uses the English-style dot (.).

Options: WORKDIRECTORY, FJPATH, DECIMALSYMBOL.
Parameters: FJFILENAME, TRAITS, GENOTYPES, ENVIRONMENTS, GENFILENAME, MAPFILENAME, FJTRAITFILENAME, FJQTLFILENAME, QSAVE.

Method
In Windows the project file is formed using the Createproject.exe executable within the Flapjack installation. A bat file containing the command line to form the project file is created, and then executed using the SUSPEND directive.

Action with RESTRICT
Any data restrictions are ignored.

See also
Directive: SUSPEND.
Procedures: QMESTIMATE, QREPORT, QSESTIMATE.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QGSELECT

Obtains a representative selection of genotypes by means of genetic distance sampling or genetic distance optimization (J. Jansen & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, monitoring); default summ
NCLUSTERS = *scalar*	The number of genotypes to be selected; must be set
METHOD = *string token*	Method to be used (sampling, optimization); default samp

Parameters

GENOTYPES = *factors*	Genotype factor; must be set
SIMILARITY = *symmetric matices*	Input similarity matrix for each selection; must be set
PRIORGROUPS = *factors*	Defines prior groupings of the genotypes
SELECTED = *variates*	Logical variate indicating whether a genotype is selected (1) as cluster centre or not (0)
NEIGHBOURS = *variates*	Saves the nearest cluster centres of the genotypes
DISTANCES = *variates*	Saves the distances of the genotypes to the nearest cluster centre
SEED = *scalars*	Seed for randomization at the start; default 0

Description

QGSELECT selects a representative subset of genotypes using a similarity matrix, provided by the SIMILARITY parameter.

The METHOD option specifies whether to use genetic distance sampling or genetic distance optimization, by setting it to one of the following settings:

sampling	genetic distance sampling using the method of Jansen & Van Hintum (2006), or
optimization	genetic distance optimization based on K-medoids cluster analysis (Kaufman & Rouseeuw 1990).

The default is METHOD=sampling.

The factor identifying the genotypes must be supplied by the GENOTYPES parameter, and the number of genotypes to be selected must be specified by the NCLUSTERS option. Prior information about the grouping of the genotypes can be supplied using the PRIORGROUPS factor.

The SEED parameter specifies the seed to use to randomize the genotypes at the start. The default value of zero continues an existing sequence, or (if none) initializes the seed automatically.

The genotype selection can be saved by the SELECTED parameter, in a logical variate containing one for each genotype selected as a cluster centre, and zero for the genotypes that are not selected. The NEIGHBOURS parameter saves the nearest cluster centre for each genotype, and the DISTANCES parameter saves the distances of each genotype to the nearest cluster centre.

The PRINT option controls the printed output, with settings:

summary	for a summary of the selection, and
monitoring	for monitoring information.

Options: PRINT, NCLUSTERS, METHOD.

Parameters: MKNAMES, SIMILARITY, PRIORGROUPS, SELECTED, NEIGHBOURS, DISTANCES, SEED.

Action with `RESTRICT`
Restrictions are not allowed.

References
Jansen, J. & Th.J.L. van Hintum (2006). Genetic distance sampling: a novel sampling method for obtaining core collections using genetic distances with an application to cultivated lettuce. *Theor. Appl. Genet.*, **114**, 421-428.

Kaufman, P. & P.J. Rousseuw (1990). Finding Groups in Data: an Introduction to Cluster Analysis. Wiley, New York.

See also
Procedure: `QMKSELECT`.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QIBDPROBABILITIES

Reads molecular marker data and calculates IBD probabilities (M.P. Boer & J.T.N.M. Thissen).

Options

`PRINT` = *string tokens*	What to print (`summary`, `loci`); default `summ`
`STEPSIZE` = *scalar*	Maximum stepsize along the genome; default 10^6, i.e. the IBD probabilities are calculated only at the marker positions
`METHOD` = *string token*	Method of calculation for IBD probabilities of RIL populations (`approximate`, `exact`); default `appr`
`POPULATIONTYPE` = *string token*	Type of population (`BC1`, `DH1`, `F2`, `RIL`, `BCxSy`, `CP`); must be set
`NGENERATIONS` = *scalar*	Number of generations of selfing for a `RIL` population
`NBACKCROSSES` = *scalar*	Number of backcrosses for a `BCxSy` population
`NSELFINGS` = *scalar*	Number of selfings for a `BCxSy` population
`MAPPINGFUNCTION` = *string token*	Mapping function (`haldane`, `kosambi`); default `hald`

Parameters

`MKSCORES` = *pointers*	Genotype codes for each marker; must be set
`CHROMOSOMES` = *factors*	The chromosome where each marker is located; must be set
`POSITIONS` = *variates*	The position on the chromosome of each marker; must be set
`MKNAMES` = *texts*	Marker names; must be set
`IDMGENOTYPES` = *texts*	Labels for the genotypes
`PARENTS` = *pointers*	Parent information; must be set
`IDPARENTS` = *texts*	Labels used to identify the parents; must be set
`PEDIGREE` = *pointers*	Defines the parents of the offspring
`ADDITIVEPREDICTORS` = *pointers*	Saves the additive genetic predictors
`ADD2PREDICTORS` = *pointers*	Saves the second (paternal) additive genetic predictors if `POPULATIONTYPE` is `CP`
`DOMINANCEPREDICTORS` = *pointers*	Saves the dominance genetic predictors if `POPULATIONTYPE` is `F2`, `RIL`, `BCxSy` or `CP`
`SCHROMOSOMES` = *factors*	Saves the chromosome where each locus is located
`SPOSITIONS` = *variates*	Saves the position on the chromosome of each locus
`LOCI` = *variates*	Saves the index number of each locus
`IDLOCI` = *texts*	Saves the locus labels
`MKLOCI` = *variates*	Saves a logical variate indicating whether each locus is a marker
`NLOCI` = *scalars*	Saves the number of loci
`NGENOTYPES` = *scalars*	Saves the number of genotypes
`APROBABILITIES` = *pointers*	Saves probabilities of the genotypes being equal to parent A
`BPROBABILITIES` = *pointers*	Saves probabilities of the genotypes being equal to parent B
`HPROBABILITIES` = *pointers*	Saves the probabilities of the genotypes being heterozygous
`ACPROBABILITIES` = *pointers*	Saves the probabilities of the genotypes being AC when `POPULATIONTYPE` is `CP`

ADPROBABILITIES = *pointers*	Saves the probabilities of the genotypes being AD when POPULATIONTYPE is CP
BCPROBABILITIES = *pointers*	Saves the probabilities of the genotypes being BC when POPULATIONTYPE is CP
BDPROBABILITIES = *pointers*	Saves the probabilities of the genotypes being BD when POPULATIONTYPE is CP
OUTFILENAME = *texts*	Name of the Genstat workbook file (*.gwb) to be created

Description

QIBDPROBABILITIES calculates conditional genotypic probabilities at specific chromosome positions. The marker scores must be set by the MKSCORES parameter and the map data by the CHROMOSOMES, POSITIONS and MKNAMES parameters. The IDMGENOTYPES parameter can be set to label the genotypes. The marker scores of the parents must be set by the PARENTS parameter, and the corresponding labels of the parents must be set by the IDPARENTS parameter. The PEDIGREE parameter can provide a pointer containing factors to identify the parents of the offspring. This parameter must be set for multiple populations.

The POPULATIONTYPE option must specify the population type. For recombinant inbred lines (POPULATIONTYPE = RIL), the NGENERATIONS option specifies the number of generations; default 3. By default, with RIL populations, the conditional genotypic probabilities are calculated by an approximate method, but you can set option METHOD=exact to use an exact method instead. For backcross inbred lines (POPULATIONTYPE = BCxSy), the NBACKCROSSES and NSELFINGS options must be set to define the number of backcrosses to the first parent and the number of selfings, respectively.

The STEPSIZE option determines the maximum step size for the calculation of the conditional probabilities. A large value (like the default value 10^6) causes conditional probabilities to be calculated only at the marker positions.

The MAPPINGFUNCTION option defines the mapping function, which can be the Haldane or the Kosambi mapping function; default haldane.

For population types BC1, DH1, F2, RIL and BCxSy the calculated probabilities can be saved by the APROBABILITIES, BPROBABILITIES and HPROBABILITIES parameters: APROBABILITIES saves the probabilities that the genotypes are homozygous for the parent A allele, BPROBABILITIES saves the probabilities that the genotypes are homozygous for the parent B allele, and HPROBABILITIES saves the probabilities that the genotypes are heterozygous. From these probabilities the ADDITIVEPREDICTORS and the DOMINANCEPREDICTORS are calculated. For all population types, except backcross populations (BC1), the genetic predictors for the additive effects are given by

```
ADDITIVEPREDICTORS = APROBABILITIES - BPROBABILITIES
```

and

```
DOMINANCEPREDICTORS = HPROBABILITIES
```

For a backcross population (BC1), they are given by

```
ADDITIVEPREDICTORS = 0.5*APROBABILITIES - 0.5*BPROBABILITIES
```

For a cross pollinated population (CP), the parents are heterozygote. For a particular locus, let AB be the genotype of the first parent, and CD the genotype of the second parent, where allele A (C) is inherited from the mother of the first (second) parent, and allele B (D) is inherited from the father of the first (second) parent. The progeny can have 4 different genotypes, namely AC, AD, BC, and BD. The calculated probabilities corresponding to the four possible genotypes can be saved by the ACPROBABILITIES, ADPROBABILITIES, BCPROBABILITIES and BDPROBABILITIES parameters. The ADDITIVEPREDICTORS, ADD2PREDICTORS and the DOMINANCEPREDICTORS can be calculated from these probabilities, as follows. The genetic

predictors for the maternal additive effects are given by

```
ADDITIVEPREDICTORS = BDPROBABILITIES + BCPROBABILITIES \
                   - ADPROBABILITIES - ACPROBABILITIES
```

the genetic predictors for the paternal additive effects by

```
ADD2PREDICTORS = BDPROBABILITIES - BCPROBABILITIES \
               + ADPROBABILITIES - ACPROBABILITIES
```

and genetic predictors for the dominance effects by

```
DOMINANCEPREDICTORS = BDPROBABILITIES - BCPROBABILITIES \
                    - ADPROBABILITIES + ACPROBABILITIES
```

The number of chromosome positions (loci) where conditional probabilities have been estimated can be saved by the NLOCI parameter, and the number of genotypes can be saved by the NGENOTYPES parameter. The labels of the loci can be saved by the IDLOCI parameter. The CHROMOSOMES and POSITIONS parameters can save the map information of the loci: CHROMOSOMES saves the chromosome numbers, and POSITIONS saves the positions on the chromosome where conditional probabilities were calculated. A unique index number for each locus can be saved by the LOCI parameter. The MKLOCI parameter saves a logical variate storing one if the locus is a marker, otherwise zero.

The PRINT option controls the printed output. The summary setting prints the number of loci and the number of genotypes, and the loci setting prints all the loci index numbers together with the IDLOCI, CHROMOSOMES and POSITIONS values.

The OUTFILENAME parameter can be used to save the information in a Genstat workbook file. This parameter should not contain an extension as the extension is automatically set as .gwb. The LOCI, IDLOCI, CHROMOSOMES and POSITIONS structures are written to a sheet named LOCI, and the ADDITIVEPREDICTORS variates are written to a sheet named ADDPREDICTORS. The ADD2PREDICTORS and/or DOMINANCEPREDICTORS variates, when relevant, are written to sheets named ADD2PREDICTORS and DOMPREDICTORS respectively.

Options: PRINT, STEPSIZE, METHOD, POPULATIONTYPE, NGENERATIONS, NBACKCROSSES, NSELFINGS, MAPPINGFUNCTION.

Parameters: MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, IDMGENOTYPES, PARENTS, IDPARENTS, PEDIGREE, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, SCHROMOSOMES, SPOSITIONS, LOCI, IDLOCI, MKLOCI, NLOCI, NGENOTYPES, APROBABILITIES, BPROBABILITIES, HPROBABILITIES, ACPROBABILITIES, ADPROBABILITIES, BCPROBABILITIES, BDPROBABILITIES, OUTFILENAME.

Method

QIBDPROBABILITIES calls an external algorithm in the dynamic link library genetics.dll.

See also

Procedures: QEXPORT, QIMPORT.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QIMPORT

Imports genotypic and phenotypic data for QTL analysis (D.A. Murray).

Options

PRINT = *string token*	What to print (catalogue, errorreport); default cata, erro
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP, AMP); must be set
MISSING = *text*	Character representing a missing genotype in Flapjack or R/QTL format; default '-'
SEPARATOR = *text*	Character separating data values in Flapjack format; default separates them by tabs
ASEPARATOR = *text*	Character separating allele values in Flapjack format; default '/'
FJROWS = *string token*	Specifies whether the genotypes or markers are stored on the rows in Flapjack format (genotypes, markers); default geno
NPARENTS = *scalar*	Number of parents in Flapjack file; default 0 for population AMP, 4 for CP, and 2 otherwise
NMKERROR = *scalar*	For data in Flapjack format, this sets a limit on the number of markers that may be found to contain errors before the import is abandoned; default 200
MKREMOVE = *string token*	Whether to remove markers with errors in Flapjack format automatically (yes, no); default no

Parameters

FILENAME = *texts*	Name of the file for import
MAPFILENAME = *texts*	Name of the map file (Flapjack or MapQTL(R))
PHEFILENAME = *texts*	Name of the phenotypic file (MapQTL(R))
MKSCORES = *pointers*	Saves the genotype codes for each marker
TRAITS = *pointers*	Saves the trait data from the phenotypic file
CHROMOSOMES = *factors*	Saves linkage groups for each marker
POSITIONS = *variates*	Saves positions of the markers within linkage groups
MKNAMES = *texts*	Saves the marker names
MKSETS = *factors*	Saves marker sets
IDMGENOTYPES = *texts*	Labels for genotypes
PARENTS = *pointers*	Saves the parent information
IDPARENTS = *texts*	Saves the labels used to identify the parents
IDFILENAME = *texts*	Specifies a file containing genotype labels for MapQTL(R) files; if unset, they are assumed to be in the .loc file
EXCLUDEMARKERS = *texts*	Specifies the names of any markers to exclude from an import in Flapjack format
MKERRORS = *texts*	In Flapjack format, this saves the names of any markers that contain errors
ERRORLOCATIONS = *pointers*	In Flapjack format, this saves a pointer to texts that identify any errors in the marker-by-genotype (individual) scores
OUTFILENAME = *texts*	Specifies the name of a Genstat workbook (.gwb) file to save the marker scores and associated information

Description

QIMPORT loads genotypic and phenotypic data for QTL analysis. The name of the genotypic data file to be imported is specified by the FILENAME parameter. The format of the file to be imported is specified by the file extension, and can be either a Flapjack text genotype file (.txt), a MapQTL(R) Locus genotype file (.loc) or a comma-delimited text (.csv). The format of the .csv file is an extended R/QTL separate genotype data .csv file format, which can include an extra column for the marker sets.

If a Flapjack genotype or MapQTL(R) Locus genotype file name is supplied, the associated map information can supplied by setting the MAPFILENAME option to a file name with the extension .txt for Flapjack or .map for MapQTL(R). For Flapjack and R/QTL formats, the POPULATIONTYPE option must be set to specify the population from which the genotypes come. For MapQTL(R), the population is determined from the .loc file. The MISSING option can specify a character to identify missing genotypes in Flapjack genotype files and R/QTL files. By default, Genstat expects the genotype data in Flapjack files to be tab-delimited, but the SEPARATOR option can be used to specify an alternative separator. Similarly, by default, Genstat expects the alleles for each genotype to be separated using a '/' character, but an alternative can be supplied using the ASEPARATOR option. For the Flapjack genotype format, the FJROWS option indicates whether the genotypes or markers are stored in the rows of the file; by default the genotypes are in the rows.

The marker scores for the genotypes are stored in a set of factors in the pointer supplied by the MKSCORES parameter. Each factor within the pointer will contain data for a marker, with factor labels supplied in the same order.

When importing genotypic data the linkage groups for each marker, marker names and positions are saved using the CHROMOSOMES, MKNAMES and POSITIONS parameters, respectively. If a .csv file is imported, any marker sets within the file can be saved using the MKSETS parameter. The grouping factor identifying marker sets in a .csv file can be saved using the MKSETS parameter.

For BC1, DH1, F2, RIL, BCxSy and CP populations, the parent information and associated names can be saved using the PARENT and IDPARENTS parameters respectively.

The genotype labels can be saved using the IDMGENOTYPES parameter. By default, for MapQTL(R) locus and map files, the genotype labels are the values 1 to *n*. However, Genstat allows individual names to be included at the bottom of the locus file, below the genotype data. The file should then include the instruction

```
Individual names:
```

followed by each individual name on a separate line in the same order as that in which the genotypes are specified for each locus. Alternatively, a text file containing the genotype labels can be supplied using the IDFILENAME parameter; each individual name should then be on a separate line in the same order as that in which the genotypes are specified for each locus in the .loc file.

For data in Flapjack format, markers can be excluded by setting the EXCLUDEMARKERS parameter to a text containing the names of the markers to omit. When importing Flapjack genotypic data, the parental and individual scores are checked for errors. You can set option MKREMOVE=yes to remove any markers that are found to contain errors, automatically from the imported data. The NMKERROR option sets a limit on the number of markers that may be found to contain errors before the import is abandoned; default 200. The names of any markers that containing errors in the parent or individual genotype scores can be saved, in a text, using the MKERROR parameter. The ERRORLOCATIONS parameter can save a pointer containing a text with marker names and a text with genotype names, identifying the marker × genotype locations of any marker score errors.

The PRINT option specifies the output to be displayed, with settings:

catalogue	produces a summary listing attributes of the data that have

been read and, for phenotypic data, a list of the data structures that have been imported,

`errorreport` gives a report of any errors in genotypic data that have been read in Flapjack format.

Phenotypic data in MapQTL(R) quantatitive data files (`.qua`) can be imported by supplying the name of the file with the `PHEFILENAME` parameter. The `TRAITS` parameter can be set to a pointer to store the identifiers (i.e. column names) read from the file. The pointer can then be used to refer to the variates containing the loaded data.

The `OUTFILENAME` can specify the name of a Genstat workbook (`.gwb`) file to save the marker scores and associated information.

Options: `PRINT`, `POPULATIONTYPE`, `MISSING`, `SEPARATOR`, `ASEPARATOR`. `FJROWS`, `NPARENTS`, `NMKERROR`, `MKREMOVE`.

Parameters: `FILENAME`, `MAPFILENAME`, `PHEFILENAME`, `MKSCORES`, `TRAITS`, `CHROMOSOMES`, `POSITIONS`, `MKNAMES`, `MKSETS`, `IDMGENOTYPES`, `PARENTS`, `IDPARENTS`, `IDFILENAME`, `EXCLUDEMARKERS`, `MKERRORS`, `ERRORLOCATIONS`, `OUTFILENAME`.

Method

See the `QEXPORT` procedure for further details of the file formats. Data in Flapjack format are read and checked using the `Dataload` dll, and the valid data are passed back to Genstat using temporary files.

See also

Procedures: `IMPORT`, `QEXPORT`, `QIBDPROBABILITIES`.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QKINSHIPMATRIX

Forms a kinship matrix from molecular markers (L.C.P. Keizer & J.T.N.M. Thissen).

Options

PRINT = *string token*	What to print (summary); default summ
METHOD = *string token*	Method to use for the calculation (correlation, dice); default dice

Parameters

MKSCORES = *pointers*	Pointer with the marker scores; must be set
IDMGENOTYPES = *texts*	Labels for the genotypes
KMATRIX = *symmetric matrices*	Saves the kinship matrix
OUTFILENAME = *texts*	Name of the file to receive the kinship matrix

Description

QKINSHIPMATRIX forms a kinship matrix from the marker scores specified by the MKSCORES parameter. The IDMGENOTYPES parameter can provide a text with row (and column) labels for the matrix.

The METHOD option specifies the method to use to calculate the coefficients of coancestries of the kinship matrix, with settings:

dice	calculates the similarities by the FSIMILARITY directive with test type dice, and
correlation	uses simple correlation coefficients.

The kinship matrix can be saved using the KMATRIX parameter, in a symmetic matrix. It can also be saved in an output file, by supplying the file name using the OUTFILENAME parameter.

By default QKINSHIPMATRIX prints summary information about the marker scores and the method used, but you can set option PRINT=* to suppress this.

Options: PRINT, METHOD.
Parameters: MKSCORES, IDMGENOTYPES, KMATRIX, OUTFILENAME.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedure: QSASSOCIATION.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QLDDECAY

Estimates linkage disequilibrium (LD) decay along a chromosome (M. Malosetti & J.T.N.M. Thissen).

Options

PRINT = *string token*	What to print (progress); default *
PLOT = *string tokens*	What to plot (ldmatrix, lddecay); default ldde
RELATIONSHIPMODEL = *string token*	What model to use to account for genetic relatedness (eigenanalysis, subpopulations, null); default eige
SCORES = pointer	Provides the scores of significant principal components, obtained from an eigenvalue analysis
SUBPOPULATIONS = *factor*	Defines groupings of genotypes into subpopulations
CHRANALYSE = *scalar*	Defines which chromosome to analyse, using a level of the CHROMOSOMES factor
MAX%MISSING = *scalar*	Markers with more than the specified % of missing values will be excluded from the LD calculations; default 20
MAXDISTANCE = *scalar*	Defines the maximum distance between markers to show in LD plots; default 30
TITLE = *text*	General title for the plots
YTITLE = *text*	Title for the y-axis
XTITLE = *text*	Title for the x-axis

Parameters

MKSCORES = *pointers*	Genotype codes for each marker; must be set
CHROMOSOMES= *factors*	Linkage groups for the markers; must be set
POSITIONS = *variates*	Positions within the linkage groups of markers; must be set
DISTANCES = *symmetric matrices*	Saves the distances between markers
R2 = *symmetric matrices*	Saves the value of r^2 between markers

Description

QLDDECAY estimates linkage disequilibrium (LD) between pairs of markers on a chromosome. The association between two markers is assessed by a linear regression model, with one marker set as response and the second one as regressor, and LD is expressed in terms of r^2 values.

The model to account for genetic relatedness between genotypes is specified by the RELATIONSHIPMODEL option, with one of the following settings:

eigenanalysis	infers the underlying genetic substructure in the population by retaining the most significant principal components from the molecular marker matrix (Patterson *et al*. 2006) – the scores of the significant axes are used as covariables in the regression model, which is effectively an approximation to the structuring of the genetic variance covariance matrix by a coefficient of coancestry matrix (kinship matrix);
subpopulations	includes a factor supplied by the SUBPOPULATIONS option in the regression model (imposing a constant covariance between genotypes within the same subpopulation);
null	makes no correction for genetic relatedness.

By default RELATIONSHIPMODEL=eigenanalysis; the scores of the significant axes are then calculated by the QEIGENANALYSIS procedure with options STANDARDIZE=frequency and SCALE=none. Alternatively, scores calculated elsewhere can be supplied, in a pointer, using the option SCORES.

LD is estimated per chromosome. It is not calculated between markers with too many missing values. The threshold is specified by the MAX%MISSING option; default 20 (i.e. 20%). While LD is calculated along the whole of the chromosome, one expects LD decay at relatively short distances. Therefore, when plotting r^2 values versus marker distances, only pairs of markers that are closer than the value specified by the MAXDISTANCE option are displayed (default 30).

The marker scores are supplied by the MKSCORES parameter, in a pointer containing a factor for each marker. The corresponding map information for the markers is supplied by the CHROMOSOMES and POSITIONS parameters. The CHRANALYSE option must be set to specify the chromosome for which the analysis is to be performed.

The parameter MKNAMES can be used to supply marker names that will be used to name rows and columns of output matrices. The DISTANCE parameter can save a symmetric matrix of distances between the markers, and the R2 parameter can save a symmetric matrix of r^2 values between markers.

The PRINT option can be set to progress, to monitor the progress of the analysis.

The PLOT option selects the graphs to plot, with settings:

lddecay	plots the probability values for the deviance ratios, on a −log10 scale, against the marker distance, and
ldmatrix	gives a shade plot of the LD matrix.

By default PLOT=lddecay. The TITLE option can be used to provide a title for the graphs, and the YTITLE and XTITLE options can supply titles for the y- and x-axis, respectively.

Options: PRINT, PLOT, RELATIONSHIPMODEL, SCORES, SUBPOPULATIONS, CHRANALYSE, MAX%MISSING, MAXDISTANCE, TITLE, YTITLE, XTITLE.
Parameters: MKSCORES, CHROMOSOMES, POSITIONS, DISTANCES, R2.

Method

QLDDECAY handles any type of marker, taking the first allele as reference (if a bi-allelic marker) or the most frequent allele if a marker has multiple alleles. The procedure fits a linear regression with one marker taken as response and a second one used as regressor. To account for genetic relatedness, the model can also include extra covariables (either principal component scores, or a grouping factor). Models are fitted using RYPARALLEL to perform several fits in parallel. From each fit the r^2 value is stored as measure of LD between the markers. Plots are produced to display results according to the settings of the PLOT option.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedures: QEIGENANALYSIS, QMASSOCIATION, QSASSOCIATION.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QLINKAGEGROUPS

Forms linkage groups using marker data from experimental populations (J. Jansen, J.T.N.M. Thissen & M.P. Boer).

Options

PRINT = *string token*	What to print (summary); default summ
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, CP); must be set
USEPENALTY = *string token*	Whether to increase the number of recombinations by 0.5 recombination per informative meiosis for each missing marker score (yes, no); default no
THRESHOLD = *scalar* or *variate*	Threshold for the recombination frequency at which markers are said to be linked; default 0.2

Parameters

MKSCORES = *pointers*	Marker scores for each marker; must be set
CHROMOSOMES = *factors* or *pointers*	Saves the linkage groups of the markers
MKNAMES = *texts*	Names of the markers; must be set
PARENTS = *pointers*	Marker scores of the parents; must be set
SMKSCORES = *pointers*	Saves the marker scores factors according to the SMKNAMES parameter
SCHROMOSOMES = *factors* or *pointers*	Saves the sorted linkage groups
SMKNAMES = *texts* or *pointers*	Saves the names of the markers according to the SCHROMOSOMES parameter
SPARENTS = *pointers*	Saves the parent information according to the SMKNAMES parameter when POPULATIONTYPE=CP

Description

QLINKAGEGROUPS forms linkage groups of markers using marker data from experimental populations. The marker scores of the genotypes are supplied in a pointer by the MKSCORES parameter. This contains a set of factors (with levels all in the same order), each one with the data for one of the markers. The names of the markers must be supplied, in a text, using the MKNAMES parameter. The marker scores of the parents must be supplied using the PARENTS parameter.

First QLINKAGEGROUPS calculates the recombination frequencies from the marker scores. The calculation depends on the population type, which must be specified by the POPULATIONTYPE option. You can set option USEPENALTY=yes to impose a penalty for missing data: the number of recombinations is then increased by 0.5 recombination per informative meiosis for each missing marker score.

Next the recombination frequencies are used to determine whether markers are linked, using a threshold provided by the THRESHOLD option; this can be either a scalar or a variate (default 0.2). If POPULATIONTYPE=CP, parent information must be supplied by the PARENTS parameter. The linkage groups can be saved using the CHROMOSOMES parameter. If THRESHOLD is a scalar, this saves a factor, otherwise it saves a pointer of factors (one for each value in the variate).

The parameters beginning with the prefix S can be used to save information, sorted into ascending order according to the levels of the CHROMOSOMES factor(s). The SCHROMOSOMES parameter saves either a single factor or a pointer of factors, according to whether THRESHOLD is a scalar or a variate. These contain all values of the linkage group designated '1', followed by the linkage group designated '2', and so on. Similarly the SMKNAMES parameter saves either a text or a pointer of texts. These contain the names of the markers, starting with those of the

first CHROMOSOMES level, then the second level, and so on. They are sorted alphabetically within each CHROMOSOMES level. The marker scores and parent information are saved by the SMKSCORES and SPARENTS parameters, respectively. These save pointers with either one or two levels of suffixes, according to whether THRESHOLD is a scalar or a variate. The information that they contain is sorted according to the SMKNAMES text.

The PRINT option controls the printed output. The summary setting prints the number of markers in each linkage group.

Options: PRINT, POPULATIONTYPE, USEPENALTY, THRESHOLD.
Parameters: MKSCORES, CHROMOSOMES, MKNAMES, PARENTS, SMKSCORES, SCHROMOSOMES, SMKNAMES, SPARENTS.

Method
The recombination frequencies are calculated by the QRECOMBINATIONS procedure, using the two-point method. Linkage groups are formed using depth-first search from a symmetric matrix of links.

Action with RESTRICT
Restrictions are not allowed.

Reference
Cormen T.H., Leiserson, C.E., Rivest R.L. & Stein, C. (2001). *Introduction to Algorithms, 2nd edition*. MIT Press and McGraw-Hill, Cambridge, Massachusetts.

See also
Procedures: QMAP, QRECOMBINATIONS.
Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

QLIST

Gets the user to select a response interactively from a list (R.W. Payne).

Option

HELP = *text*	Help information for the QUESTION

Parameters

ALTERNATIVES = *texts*	Alternatives from which each choice is to be made
CODES = *texts*	Codes to use to represent each set of alternatives
PREAMBLE = *texts*	Preamble for the question used to select from each set of alternatives
CHOICE = *texts*	Alternative chosen from each set
NCHOICE = *scalars*	Numbers of the chosen alternatives (0 if exit has been chosen instead)

Description

The QUESTION procedure provides a convenient way of getting the user to choose a response from a short list. However, the size constraints of the standard computer screen mean that this does not work effectively for lists of more than about 16 items. QLIST overcomes this limitation by repeated calls of QUESTION. Each call displays 16 choices, together with the option of exiting without making a selection or, after all the choices have been displayed, of repeating the list.

Option: HELP.
Parameters: ALTERNATIVES, CODES, PREAMBLE, CHOICE, NCHOICE.

Method

QLIST makes repeated use of the QUESTION procedure until a response is obtained.

See also

Directive: QDIALOG.
Procedures: QFACTOR, QUESTION.
Genstat Reference Manual 1 Summary sections on: Program control, Calculations and manipulation.

QMAP

Constructs genetic linkage maps using marker data from experimental populations (J. Jansen, J.T.N.M. Thissen & M.P. Boer).

Options

PRINT = *string token*	What to print (map, monitoring, summary); default summ
PLOT = *string token*	What to plot (frequencies, map); default map
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, CP); must be set
USEPENALTY = *string token*	Whether to increase the number of recombinations by 0.5 recombination per informative meiosis for each missing marker score (yes, no); default no
SPATIALMETHOD = *string token*	Which method to use for clustering (sampling, optimization, none); default opti for population CP, samp otherwise
NGROUPS = *scalar*	Number of groups for clustering; default 10
MAPCHROMOSOMES = *variate*, *text* or *scalar*	Allows a subset of chromosomes to be mapped; default * i.e. all the chromosomes
LINKAGEPHASES = *string token*	Controls estimation of linkage phases for population type CP (estimate, omit); default esti
TITLE = *text*	General title for the graph
OUTFILENAME = *text*	Name (without extension) of the Flapjack files to be created

Parameters

MKSCORES = *pointers*	Marker scores for each marker; must be set
CHROMOSOMES = *factors*	Factor defining the linkage groups
POSITIONS = *variates*	Saves the positions of markers
MKNAMES = *texts*	Names of the markers; must be set
IDMGENOTYPES = *texts*	Names of the genotypes
PARENTS = *pointers*	Marker scores of the parents; must be set
IDPARENTS = *texts*	Labels to identify the parents
SMKSCORES = *pointers*	Saves the scores of the markers, sorted according to the markers in the SCHROMOSOMES factor (if CHROMOSOMES is set) and the SPOSITIONS variate
SCHROMOSOMES = *factors*	Saves the sorted linkage groups
SPOSITIONS = *variates*	Saves the sorted positions of markers (within the sorted linkage groups if CHROMOSOMES is set)
SMKNAMES = *texts*	Saves the names of the markers, sorted according to the SCHROMOSOMES factor (if CHROMOSOMES is set) and the SPOSITIONS variate
SPARENTS = *pointers*	Saves the marker scores of the parents, sorted according to the markers in the SCHROMOSOMES factor (if CHROMOSOMES is set) and the SPOSITIONS variate
SEED = *scalars*	Seed for the random numbers used for spatial sampling; default 0

Description

QMAP calculates the order and positions of the markers per chromosome or linkage group. The marker scores of the genotypes are supplied in a pointer by the MKSCORES parameter. This

contains a set of factors (with levels all in the same order), each one with the data for one of the markers. If the CHROMOSOMES parameter is set, the calculation of the positions is done separately for each of its levels (otherwise the markers are assumed to belong to the same linkage group). The MAPCHROMOSOMES option can be set to specify that the calculations are done only for only a subset of the CHROMOSOMES. The names of the markers must be supplied (in a text) using the MKNAMES parameter, and the names of the genotypes must be supplied (also in a text) using the IDMGENOTYPES parameter.

The POPULATIONTYPE option must be set to specify the type of population from which the marker scores have been obtained. The marker scores of the parents must be supplied using the PARENTS parameter. The names of the parents can be supplied using the IDPARENTS parameter. For population types DH1, BC1, F2 and RIL the calculation of the positions starts with the calculation of the number of recombinations per linkage group. The USEPENALTY option controls whether the number of recombinations is increased by 0.5 recombination per informative meiosis for each missing marker score.

This is followed by a spatial clustering. The SPATIALMETHOD option specifies whether this uses random sampling or spatial optimization, or you can set SPATIALMETHOD=none to suppress the clustering. The SEED option specifies the seed for the random numbers used for random sampling; the default of zero selects the seed at random, using the computer clock, or continues the existing sequence of random numbers if any have been used already, earlier in the job. The NGROUPS option specifies the number of groups; the default of 10 will usually lead to recombination frequencies between the markers that form the cluster centres of about 0.1. The cluster centres are used to obtain a framework map. After ordering the markers, recombination frequencies between adjacent markers are calculated using the multi-point maximum likelihood method. The positions of the markers can be saved, in a variate, using the POSITIONS parameter. For population type CP, you can set option LINKAGEPHASES=omit to suppress determination of the linkage phases in both parents.

By default QMAP displays a genetic map, but you can set PLOT=* to suppress this. The TITLE option allows you to supply a title for the graph. Also, unless you set option PRINT=*, QMAP prints the number of linkage groups and the minimum, mean and maximum of the POSITIONS values per linkage group.

The parameters beginning with the prefix S can be used to save information sorted in ascending order according to the levels of the CHROMOSOMES factor. The SCHROMOSOMES factor contains all values of the linkage group designated '1', followed by the linkage group designated '2', and so on. The SMKNAMES parameter contains the names of the markers, starting with those of the first CHROMOSOMES level, then the second level, and so on. They are sorted alphabetically within each CHROMOSOMES level. The marker scores are saved by the SMKSCORES parameter, and are sorted according to the SMKNAMES text. The parent information that can be saved by the SPARENTS parameter is sorted in the same way.

The OUTFILENAME option can be used to save the sorted marker scores and positions in two Flapjack files. This parameter should not contain an extension as the extension is defined automatically as .txt. The name is extended with '_geno' for the marker scores, and with '_map' for the positions.

Options: PRINT, PLOT, POPULATIONTYPE, USEPENALTY, SPATIALMETHOD, NGROUPS, MAPCHROMOSOMES, LINKAGEPHASES, TITLE, OUTFILENAME.

Parameters: MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, IDMGENOTYPES, PARENTS, IDPARENTS, SMKSCORES, SCHROMOSOMES, SPOSITIONS, SMKNAMES, SPARENTS, SEED.

Method

QMAP calculates the order of markers using simulated annealing in conjunction with spatial sampling or optimization. The spatial sampling/optimization is used to obtain a framework map;

it reduces the size of the optimization problem and leads to a reduction of the effects of errors on the marker ordering. When using spatial sampling, at each step of the sampling process one marker is selected at random and all markers within a given distance, known as the *sampling radius*, of that marker are excluded from further sampling. Distance between markers is measured by their recombination frequencies. Sets of markers sampled in this way are more or less evenly spread along the chromosomes. The sampling radius is varied in order to order to obtain a set of markers of fixed size. When using spatial optimization, a set of framework markers is obtained by minimizing the average distance between all markers and the nearest marker in the set of framework markers, using simulated annealing. The set of markers obtained by spatial sampling is used as starting configuration for spatial optimization. Multi-point maximum likelihood estimates of recombination frequencies between adjacent markers on the genetic linkage map are obtained by the EM algorithm using a hidden Markov model.

Action with `RESTRICT`
Restrictions are not allowed.

References
Jansen, J., de Jong, A.G. & van Ooijen, J.W. (2001). Constructing dense genetic linkage maps. *Theor. Appl. Genet.*, **102**, 1113-1122.

Jansen, J. (2005). Construction of linkage maps in full-sib families of diploid outbreeding species by minimizing the number of recombinations in hidden inheritance vectors. *Genetics*, **170**, 2013-2025.

Lander, E.S. & Green, P. (1987). Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci. USA*, **84**, 2363-2367.

See also
Procedures: `DQMAP`, `QLINKAGEGROUPS`, `QRECOMBINATIONS`.
Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

QMASSOCIATION

Performs multi-environment marker-trait association analysis in a genetically diverse population using bi-allelic and multi-allelic markers (M. Malosetti & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, progress); default summ
PLOT = *string tokens*	What to plot (profile, map); default prof, map
RELATIONSHIPMODEL = *string token*	What model to use to account for genetic relatedness (eigenanalysis, subpopulations, null); default eige
VCMODEL = *string token*	Specifies the variance-covariance model for the set of environments (identity, diagonal, cs, hcs, outside, fa, unstructured, best); default best
CRITERION = *string token*	Defines which criterion is used to compare the different covariance structures (aic, sic); default sic
MINORALLELE = *scalar*	Frequency of minor alleles; default 0.05
THRESHOLD = *scalar*	Threshold value for significant LD, on the −log10 scale; default 2
SUBPOPULATIONS = *factor*	Defines groupings of genotypes into subpopulations
MODELPART = *string token*	Defines which part of the model should include SUBPOPULATIONS if RELATIONSHIPMODEL is set to subpopulations, or the principal components scores if RELATIONSHIPMODEL is set to eigenanalysis (fixed, random); default rand
SCALING = *string token*	Whether to scale the scores by the square roots of their singular values if RELEATIONSHIPMODEL is set to eigenanalysis (singularvalues, none); default sing
STANDARDIZE = *string token*	Whether to standardize the marker scores according to their frequencies (frequency, none); default freq
TITLE = *text*	General title for the plots
YTITLE = *text*	Title for the y-axis
XTITLE = *text*	Title for the x-axis

Parameters

TRAIT = *variates*	Phenotypic trait to analyse; must be set
GENOTYPES = *factors*	Genotype factor; must be set
ENVIRONMENTS = *factors*	Environment factor; must be set
MKSCORES = *pointers*	Genotype codes for each marker; must be set
CHROMOSOMES = *factors*	Linkage groups for the markers; must be set
POSITIONS = *variates*	Positions within the linkage groups of markers; must be set
MKNAMES = *texts*	Marker names
WALDSTATISTICS = *variates*	Saves the Wald test statistics
NDF = *variates*	Saves the degrees of freedom associated to the Wald test
MINLOG10P = *variates*	Saves the associated probability values of the Wald test statistics, on a −log10 scale
QSAVE = *pointers*	Saves a pointer with information and results for the significant effects
DFILENAME = *texts*	Name of the graphics file for the plots

Description

QMASSOCIATION performs a mixed model marker-trait association analysis (also known as linkage disequilibrium mapping) with data from a single-environment trial. When testing for marker-trait association in a genetically diverse population, it is necessary to account for population structure, which introduces non-independence between genotypes as a result of common genetic background. In addition, the multi-environment context requires to the definition of the variance covariance model to use for the random genetic effects in the different environments; this is specified by the VCMODEL option. The default is to fit all models and select the best one according to the criterion given by the CRITERION option, either the Schwarz Information Criterion (the default), or the Akaike Information Criterion.

The trait response variate is supplied by the TRAIT parameter, and the corresponding environment and genotype factors must be specified by the ENVIRONMENTS and GENOTYPES parameters, respectively. The marker scores are supplied in a pointer by the MKSCORES pointer. The length of the MKSCORES pointer must be equal to the number of markers, and each structure of the pointer must be a factor. The corresponding map information for the markers must be given by the CHROMOSOMES and POSITIONS parameters. Labels for the markers can be supplied by the MKNAMES parameter.

The model to account for genetic relatedness between genotypes is specified by the RELATIONSHIPMODEL option, with one of the following settings:

eigenanalysis	infers the underlying genetic substructure in the population by retaining the most significant principal components from the molecular marker matrix (Patterson *et al*. 2006) – the scores of the significant axes are used as covariables in the mixed model, which effectively is an approximation to the structuring of the genetic variance covariance matrix by a coefficient of coancestry matrix (kinship matrix);
subpopulations	includes a factor supplied by the SUBPOPULATIONS option in the mixed model; and
null	makes no correction for genetic relatedness.

By default RELATIONSHIPMODEL=eigenanalysis. The scores of the significant axes are then calculated by the QEIGENANALYSIS procedure. The STANDARDIZE and SCALING options control whether the MKSCORES factors are standardized and scaled.

The threshold for significant marker trait association (on a −log10 scale) is defined by the THRESHOLD option. The default value is 2.

The MINORALLELE option defines the frequency q below which alleles are considered rare. Rare alleles are automatically pooled together. Markers whose major frequency allele is greater than or equal to $1-q$ are considered close to fixation and are not used in the analysis.

The MODELPART option controls whether the principal components scores (if RELATIONSHIPMODEL=eigenanalysis) or the subpopulations factor (if RELATIONSHIPMODEL=subpopulations) are included as random or fixed terms (default random).

The PRINT option controls printed output, with settings:

summary	to print the list of markers with a significant association with the trait, and
progress	to monitor the progress of the analysis.

The default is PRINT=summary.

The PLOT option controls what graphs are produced, with settings:

profile	plots a genome wide profile of the -log10(P) of the test statistic, and
map	plots a map with the location of the detected significant

markers, highlighting whether or not the marker showed significant interaction with the environment.

By default both are plotted. The TITLE option can be used to provide a title for the graph, and the YTITLE and XTITLE options can supply titles for the y- and x-axis, respectively. By default, the plot is sent to the screen. However, you can supply a file for the plot, using the DFILENAME parameter. You can discover the types of graphics file that are supported by running the command.

```
DHELP possible
```

The Wald test statistics, their numbers of degrees of freedom and the associated probability values on a −log10 scale can be saved by the WALDSTATISTICS, NDF and MINLOG10P parameters, respectively. The QSAVE parameter can be used to save a pointer containing information and results for the significant markers. The elements of the pointer are labelled as follows to simplify their subsequent use:

'procedure'	stores the string 'QMASSOCIATION' to indicate the source of the results,
'index'	index numbers of the significant markers,
'mkname'	marker names,
'chromosomes'	chromosomes,
'positions'	positions,
'minlog10p'	probability values on a −log10 scale,
'nalleles'	number of alleles,
'interaction'	an indicator of whether there was a significant interaction with the environment,
'allele'	label of the relevant allele,
'frequency'	allele frequencies,
'effects'	effects,
'seeffects'	standard errors of the effects, and
'sed'	mean, minimum and maximum standard error of differences of the effects.

The elements 'procedure', 'mkname' and 'interaction' are text structures; 'index', 'positions', 'minlog10P' and 'nalleles' are variates; 'allele', 'frequency', 'effects', 'seeffects' and 'sed' are pointers; 'chromosomes' is a factor.

Options: PRINT, PLOT, RELATIONSHIPMODEL, VCMODEL, CRITERION, MINORALLELE, THRESHOLD, SUBPOPULATIONS, MODELPART, SCALING, STANDARDIZE, TITLE, YTITLE, XTITLE.

Parameters: TRAIT, GENOTYPES, ENVIRONMENTS, MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, WALDSTATISTICS, NDF, MINLOG10P, QSAVE, DFILENAME.

Method

QMASSOCIATION performs a mixed model marker-trait association analysis, or LD mapping, in the context of multiple environments. Consequently, it requires two major aspects to be handled in the statistical model: first it needs to account for the heterogeneous genetic relatedness between individuals in the population (sometimes referred as "population structure"); and second it needs to model the genetic correlations between environments, since same the individuals are measured across environments.

Depending on the model settings, the model for marker trait association may included the following terms: an intercept μ, an environment main effect (E_j), the effects associated with k principal components ($PCscore_{ki}$), the effects of genotype groups ($Group_k$), the effects of the tested markers (MK) and their interactions with the environment, and the effects of genotypes (G_i) and their interactions with the environments.

The RELATIONSHIPMODEL option specifies which of the three possible models to use for the relatedness, and the MODELPART option controls whether these terms are treated as fixed or random.

Model	Fixed	Fixed or random	Fixed	Random
Eigenanalysis	$\mu + E_j +$	$\Sigma_i \{ PCscore_{ki} + (PCscore_{ki}.E_j \} +$	$MK + MK.E_j +$	$G_i + G_i.E_j$
Subpopulations	$\mu + E_j +$	$Group_k + Group_k.E_j +$	$MK + MK.E_j +$	$G_i + G_i.E_j$
Null	$\mu + E_j +$		$MK + MK.E_j +$	$G_i + G_i.E_j$

The next step is to define the variance-covariance model for the random genotype and genotype by environment interaction terms. The VCMODEL option allows you either to define a specific model or, with the best setting, to select the model automatically by fitting all possible models and choosing the best one using the Schwarz or Akaike information criterion.

A Wald test is then used for each marker, individually, to test the null hypothesis that its effect is zero in every environment. The most frequent allele is set as the reference level. This is done by removing the marker main effect from the model in the VCOMPONENTS statement, which means leaving only the term MK.E. (As a result, the term MK.E should not be interpreted as marker-by-environment interaction, but as marker-environment specific effects.) If the null hypothesis is rejected, a second test is performed to check whether the marker-by-environment interaction is significant. This is done by refitting the model, but this time including the marker main effect. If the marker-by-environment interaction is found to be non-significant, marker main effects are stored. Otherwise environment-specific marker effects are stored.

Action with RESTRICT

Restrictions are not allowed.

Reference

Patterson, N., Price, A.L., Reich, D. (2006). Population structure and eigenanalysis. *PLoS Genetics*, **2**, e190. doi:10.1371/journal.pgen.0020190

See also

Procedures: QEIGENANALYSIS, QLDDECAY, QSASSOCIATION, QREPORT.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMATCH

Matches different data structures to be used in QTL estimation (L.C.P. Keizer & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, details); default summ
GEN%MISSING = *scalar*	Percentage of missing values allowed for a genotype; default 50
MK%MISSING = *scalar*	Percentage of missing values allowed for a marker; default 50
MK%EXTREME = *scalar*	Extreme allele percentage allowed for a marker; default 5
GENSELECTION = *variate*	Logical variate containing the value one for the genotypes to retain and zero for those to remove (supersedes the options GEN%MISSING, MK%MISSING and MK%EXTREME)
MKSELECTION = *variate*	Logical variate containing the value one for the markers to retain and zero for those to remove (supersedes the options GEN%MISSING, MK%MISSING and MK%EXTREME)
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP, AMP); must be set
OUTFILEPREFIX = *text*	Prefix for the output file names; default * i.e. files not saved

Parameters

TRAITS = *pointers* or *variates*	Quantitative traits
GENOTYPES = *factors*	Genotype factors corresponding to the traits
ENVIRONMENTS = *factors*	Environment factors corresponding to the traits
MKSCORES = *pointers*	Marker scores; must be set
CHROMOSOMES = *factors*	Chromosomes corresponding to the markers
POSITIONS = *variates*	Positions on the chromosomes corresponding to the markers
MKNAMES = *texts*	Names of the markers
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the markers
PARENTS = *pointers*	Parent information
IDPARENTS = *texts*	Labels used to identify the parents
KMATRIX = *symmetric matrices*	Kinship matrices containing coefficients of coancestries
SUBPOPULATIONS = *factors*	Groups of genotypes
STRAITS = *pointers* or *variates*	Saves the sorted quantitative traits
SGENOTYPES = *factors*	Saves the sorted genotype factors
SENVIRONMENTS = *factors*	Saves the sorted environment factors
SMKSCORES = *pointers*	Saves the sorted marker scores; must be set
SCHROMOSOMES = *factors*	Saves the sorted chromosomes corresponding to the markers
SPOSITIONS = *variates*	Saves the sorted positions on the chromosomes corresponding to the markers
SMKNAMES = *texts*	Saves the sorted names of the markers
SIDMGENOTYPES = *texts*	Saves the sorted labels for the genotypes
SPARENTS = *pointers*	Saves the sorted parent information
SIDPARENTS = *texts*	Saves the sorted labels used to identify the parents
SKMATRIX = *symmetric matrices*	Saves the sorted kinship matrices

SSUBPOPULATIONS = *factors* Saves the sorted groups of genotypes

Description

QMATCH matches the various data structures that can be used in QTL detection. These include molecular marker information of sets of genotypes, map information, phenotypic information, and also genetic relatedness information in the form of genotype groupings and kinship matrices. QMATCH can be used to align all these data for further analyses.

Molecular marker information is supplied by the MKSCORES, MKNAMES and IDMGENOTYPES parameters; MKSCORES must be set. The type of population from which the genotypes come must be specified using the POPULATIONTYPE option. If parental genotypes are known (designed crosses), the marker scores of the parents can be supplied by the PARENTS parameter, and their labels can be specified by the IDPARENTS parameter. Molecular map information is supplied by the CHROMOSOMES and POSITIONS parameters. Phenotypic data are specified by the TRAITS parameter, as a variate for a single trait, or as a pointer containing several variates for more than one trait. The GENOTYPES parameter supplies a factor defining the genotype of each trait observation, and the ENVIRONMENTS parameter can supply a factor defining the environment of each observation when the data are from a multi-environment trial. Genetic relatedness information, used in association mapping analyses, can be given as a kinship matrix using the KMATRIX parameter, or a grouping factor using the SUBPOPULATIONS parameter.

QMATCH matches the different data sets together, with respect to the same set of genotypes (MKSCORES and TRAITS), or the same set of markers (MKSCORES and the map structures). The non-common genotypes and/or markers are removed.

In addition to subsetting the data, the procedure can also be used to remove genotypes and/or markers with too many missing values. The GEN%MISSING option sets a threshold on the percentage of missing values within each genotype (default 50); genotypes with more than that percentage of missing scores are excluded. Similarly, the MK%MISSING option sets a threshold on the percentage of missing values within each marker (default 50); markers with more than that percentage of missing scores are excluded. This can also be done with the MK%EXTREME option; markers are then excluded if one allele percentage of that marker is greater than the MK%EXTREME value.

In some situations you may already know which markers or genotypes you want to remove. If so, you can set the GENSELECTION and MKSELECTION options (and the GEN%MISSING, MK%MISSING and MK%EXTREME options are then ignored). The setting of each option is a logical variate containing the value one for the genotypes or markers (respectively) to retain, and zero for those that are to be removed. If any of these two options is set, no checks are carried out using the GEN%MISSING, MK%MISSING and MK%EXTREME options.

The modified data structures can be saved using the parameters beginning with the prefix S. The SMKSCORES parameter, which must be set, saves the marker scores. If only the MKSCORES and SMKSCORES parameters are specified, the SMKSCORES variates are sorted according to the labels of the MKSCORES pointer. If the MKNAMES and/or the IDMGENOTYPES parameters are also specified, sorting is then done according to their values. If the map structures (CHROMOSOMES and POSITIONS) are also set, the SMKSCORES variates are first sorted in ascending order according to the levels of the CHROMOSOMES factor, and then within each chromosome (linkage group) in ascending order of the POSITIONS. If the SMKNAMES, SCHROMOSOMES, SPOSITIONS, SPARENTS and SIDPARENTS are set, their values are sorted in the same way. The structures corresponding to the traits (i.e. STRAITS, SGENOTYPES and SENVIRONMENTS) are sorted in the same way as the SIDMGENOTYPES text; if these structures contain values from more than one environment, the sorting according to the values of SIDMGENOTYPES is done within each environment. Finally, if the KMATRIX and/or the SUBPOPULATIONS parameters are set, their sorted values can be saved by the SKMATRIX and SSUBPOPULATIONS parameters, respectively.

The OUTFILEPREFIX option can be used to define the initial part of the names of files to save

the modified data. The text supplied by the option should not contain an extension, as the extension is defined automatically for the different files. The saved marker scores are stored in a flapjack file with '_geno.txt' added to OUTFILEPREFIX, the saved map structures in a flapjack map file with '_map.txt' added, and the saved phenotypical structures in a Genstat spreadsheet file with '_pheno.gsh' added. The saved kinship matrix and the saved subpopulations structures are also stored in Genstat spreadsheet files, with '_kmat.gsh' and '_subpop.gsh' added, respectively.

The PRINT option controls the printed output, with settings:

summary	for a general summary of the changes, and
details	for details of the omitted genotypes and markers, etc.

Options: PRINT, GEN%MISSING, MK%MISSING, MK%EXTREME, GENSELECTION, MKSELECTION, POPULATIONTYPE, OUTFILEPREFIX.

Parameters: TRAITS, GENOTYPES, ENVIRONMENTS, MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, IDMGENOTYPES, PARENTS, IDPARENTS, KMATRIX, SUBPOPULATIONS, STRAITS, SGENOTYPES, SENVIRONMENTS, SMKSCORES, SCHROMOSOMES, SPOSITIONS, SMKNAMES, SIDMGENOTYPES, SPARENTS, SIDPARENTS, SKMATRIX, SSUBPOPULATIONS.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedure: QMKDIAGNOSTICS.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMBACKSELECT

Performs a QTL backward selection for loci in multi-environment trials or multiple populations (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

`PRINT` = *string tokens*	What to print (`summary`, `model`, `components`, `effects`, `means`, `stratumvariances`, `monitoring`, `vcovariance`, `deviance`, `Waldtests`, `missingvalues`, `covariancemodels`); default `summ`
`POPULATIONTYPE` = *string token*	Type of population (`BC1`, `DH1`, `F2`, `RIL`, `BCxSy`, `CP`); must be set
`ALPHALEVEL` = *scalar*	Defines a significance level; default 0.05
`VCMODEL` = *string token*	Defines the variance-covariance model for the set of environments (`identity`, `diagonal`, `cs`, `hcs`, `outside`, `fa`, `fa2`, `unstructured`); default `cs` for multi-environment trials, and `diagonal` for multiple populations
`VCPARAMETERS` = *string token*	Whether to re-estimate the variance-covariance model parameters (`estimate`, `fix`); default `esti`
`VCSELECT` = *string token*	Whether to re-select the variance-covariance model (`no`, `yes`); default `no`
`CRITERION` = *string token*	Criterion to use for model selection (`aic`, `sic`); default `sic`
`FIXED` = *formula*	Defines extra fixed effects
`UNITFACTOR` = *factor*	Saves the units factor required to define the random model when `UNITERROR` is to be used
`MVINCLUDE` = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (`explanatory`, `yvariate`); default `expl`, `yvar`
`MAXCYCLE` = *scalar*	Limit on the number of iterations; default 100
`WORKSPACE` = *scalar*	Number of blocks of internal memory to be set up for use by the `REML` algorithm; default 100

Parameters

`TRAIT` = *variates*	Quantitative trait to be analysed; must be set
`GENOTYPES` = *factors*	Genotype factor; must be set
`ENVIRONMENTS` = *factors*	Environment factor; must be set for a multi-environment trial
`POPULATIONS` = *factors*	Population factor; must be set for a multiple-population analysis
`UNITERROR` = *variates*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default `*` i.e. omitted
`VCINITIAL` = *pointers*	Initial values for the parameters of the variance-covariance model
`SELECTEDMODEL` = *texts*	`VCMODEL` setting for the selected covariance structure
`ADDITIVEPREDICTORS` = *pointers*	Additive genetic predictors; must be set
`ADD2PREDICTORS` = *pointers*	Second (paternal) set of additive genetic predictors
`DOMINANCEPREDICTORS` = *pointers*	Dominance genetic predictors
`CHROMOSOMES` = *factors*	Chromosomes corresponding to the genetic predictors;

	must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
IDLOCI = *texts*	Labels for the loci
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the genetic predictors
QTLCANDIDATES = *variates*	Specifies the locus index numbers from which to start the selection; must be set
QTLSELECTED = *variates*	Saves the index numbers of the selected QTLs
INTERACTIONS = *variates*	Saves a logical variate indicating whether each selected QTL showed a significant (1) or non-significant (0) QTL-by-environment or QTL-by-population interaction
DOMSELECTED = *variates*	Saves a logical variate indicating whether each selected QTL showed a significant (1) or non-significant (0) effect of the DOMINANCEPREDICTORS
DOMINTERACTIONS = *variates*	Saves a logical variate indicating whether each selected QTL showed a significant (1) or non-significant (0) dominance-by-environment or dominance-by-population interaction
WALDSTATISTICS = *variates*	Saves the Wald test statistics
PRWALD = *variates*	Saves the associated Wald probabilities

Description

QMBACKSELECT selects QTLs by backward selection from a list of candidate QTLs (loci) in multi-environment trials. Alternatively, it can analyse data from multiple populations. It uses means per genotype-environment or genotype-population combinations as phenotypic data, but weights can be attached to the means (see the UNITERROR parameter and the UNITFACTOR option below). The response variable must be specified by the TRAIT parameter, and the corresponding environment and genotype factors must be specified by the ENVIRONMENTS and GENOTYPES parameters, respectively. The POPULATIONTYPE option must be set to specify the population from which the genotypes have been derived. For a multiple-population analysis, the POPULATIONS parameter should be set (to a factor) instead of ENVIRONMENTS.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive effects can be included in the model by specifying dominance genetic predictors using the DOMINANCEPREDICTORS parameter (e.g. in a F2 population). In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified, the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked.

The set of candidate QTLs must be supplied by the QTLCANDIDATES parameter. The model assumes ENVIRONMENTS (or POPULATIONS) as a fixed term, and GENOTYPES as a random term. Extra fixed effects can be defined using the FIXED option. A multi-Normal distribution is assumed for the random genetic effects, with mean vector 0 and variance-covariance matrix $\mathbf{\Sigma}$. The VCMODEL option defines the model to use for $\mathbf{\Sigma}$. See the VGESELECT procedure for details of the available models; the default is to use compound symmetry for multi-environment trials, and diagonal for multiple populations. Initial values for the parameters in the variance-covariance model can be specified by the VCINITIAL parameter. The VCPARAMETERS option

controls whether the variance-covariance parameters are re-estimated at each step of the backward selection (`VCPARAMETERS=estimate`), or whether they are fixed at the defined initial values (`VCPARAMETERS=fix`). The `VCSELECT` option defines whether an extra check is made at each step on the variance-covariance model, to assess whether a simpler model is more suitable than the current model (based on the criterion defined by the `CRITERION` option). The `SELECTEDMODEL` parameter stores the final variance-covariance model that is selected. The significance level to use at each step of the backward selection process is given by the `ALPHALEVEL` option (default 0.05).

The `MVINCLUDE`, `MAXCYCLE` and `WORKSPACE` options operate in the same way as these options of the `REML` directive. The `UNITERROR` parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The `UNITFACTOR` option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using `VKEEP`).

The `PRINT` option specifies the output to be displayed. The `summary` setting prints the information about the QTLs retained in the model, and the other settings correspond to those in the `PRINT` option of the `REML` directive.

The list of selected QTLs can be saved by the `QTLSELECTED` parameter, and a logical variate that indicates whether the selected QTL showed a significant QTL-by-environment (or QTL-by-population) interaction can be saved by the `INTERACTIONS` parameter. This interaction is the combined effect of the `ADDITIVEPREDICTORS`, `ADD2PREDICTORS` and `DOMINANCEREDICTORS` pointers if specified. After the final step of the backward selection, extra tests are performed if the `DOMINANCEPREDICTORS` parameter is set. If the selected QTL has no interaction effect with environment (or population), a test is performed of whether the dominance effect has a significant contribution in the combined QTL effect. If dominance is significant, the corresponding units of the logical variate saved by the `DOMSELECTED` parameter are set to one; the other units are set to zero. If the selected QTL has significant interaction with environment (or population), a test is performed of whether the dominance-by-environment (or dominance-by-population) interaction has a significant contribution in the combined QTL-by-environment (or QTL-by-population) interaction. If the dominance-by-environment (or dominance-by-population) interaction is significant, the corresponding units of the logical variate saved by `DOMINTERACTIONS` parameter are set to one; the other units are set to zero. The Wald test and associated probability values for the combined effects (including the possible not-significant dominance and dominance-by-environment or dominance-by-population interactions) of the selected QTLs can be saved by the `WALDSTATISTICS` and `PRWALD` parameters, respectively.

Options: `PRINT`, `POPULATIONTYPE`, `ALPHALEVEL`, `VCMODEL`, `VCPARAMETERS`, `VCSELECT`, `CRITERION`, `FIXED`, `UNITFACTOR`, `MVINCLUDE`, `MAXCYCLE`, `WORKSPACE`.

Parameters: `TRAIT`, `GENOTYPES`, `ENVIRONMENTS`, `POPULATIONS`, `UNITERROR`, `VCINITIAL`, `SELECTEDMODEL`, `ADDITIVEPREDICTORS`, `ADD2PREDICTORS`, `DOMINANCEPREDICTORS`, `CHROMOSOMES`, `POSITIONS`, `IDLOCI`, `IDMGENOTYPES`, `QTLCANDIDATES`, `QTLSELECTED`, `INTERACTIONS`, `DOMSELECTED`, `DOMINTERACTIONS`, `WALDSTATISTICS`, `PRWALD`.

Method

`QMBACKSELECT` starts with the following mixed models, which include a set L of candidate QTLs:

1) $y_{ij} = \mu + E_j + \Sigma_{l \in L}\, x_{il}^{add}\, \alpha_{jl}^{add} + GE_{ij}$

if only `ADDITIVEPREDICTORS` are specified

2) $y_{ij} = \mu + E_j + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_{jl}^{add} + x_{il}^{dom}\, \alpha_{jl}^{dom}) + GE_{ij}$

if `DOMINANCEPREDICTORS` are also specified

3) $y_{ij} = \mu + E_j + \Sigma_{l \in L} (x_{il}^{add} \alpha_{jl}^{add} + x_{il}^{add2} \alpha_{jl}^{add2} + x_{il}^{dom} \alpha_{jl}^{dom}) + GE_{ij}$

if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_{ij} is the trait value of genotype i in environment (or population) j, E_j is the environment (or population) main effect, x_{il}^{add} are the additive genetic predictors of genotype i for locus l, and α_{jl}^{add} are the associated effects. In models 2 and 3, x_{il}^{dom} are the dominance genetic predictors, and α_{jl}^{dom} are the associated effects. In model 3, x_{il}^{add} are the additive genetic predictors for maternal genotype i at locus l, x_{il}^{add2} are the additive genetic predictors for paternal genotype i, and α_{jl}^{add} and α_{jl}^{add2} are the associated effects. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). GE_{ij} is assumed to follow a multi-Normal distribution with mean vector 0, and a variance covariance matrix Σ, that can either be modelled explicitly (with an unstructured model) or by some parsimonious model (defined by option VCMODEL) as described in the VGESELECT procedure.

The backward selection procedure starts with the initial set of loci (defined by the QTLCANDIDATES parameter), and checks whether all loci are significant. If not, the locus with the lowest Wald test statistic is dropped from the model. This process is repeated until all loci in the model are significant. The procedure then switches to test whether the remaining QTLs show significant QTL-by-environment (or QTL-by-population) interaction, by breaking down the QTL effects into QTL main effects and QTL-by-environment (or QTL-by-population) interaction effects. If the QTL-by-environment (or QTL-by-population) interaction term is not significant, only a main effect is retained in the model for the corresponding QTL.

Action with RESTRICT

Restrictions are not allowed.

Reference

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QMESTIMATE, QMQTLSCAN, QMVAF, VGESELECT.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMESTIMATE

Calculates QTL effects in multi-environment trials or multiple populations (M.P Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

`PRINT` = *string tokens*	What to print (`summary`, `model`, `components`, `effects`, `means`, `stratumvariances`, `monitoring`, `vcovariance`, `deviance`, `Waldtests`, `missingvalues`, `covariancemodels`); default `summ`
`POPULATIONTYPE` = *string token*	Type of population (`BC1`, `DH1`, `F2`, `RIL`, `BCxSy`, `CP`); must be set
`NGENERATIONS` = *scalar*	Number of generations of selfing for a `RIL` population
`NBACKCROSSES` = *scalar*	Number of backcrosses for a `BCxSy` population
`NSELFINGS` = *scalar*	Number of selfings for a `BCxSy` population
`VCMODEL` = *string token*	Specifies the variance-covariance model for the set of environments or populations (`identity`, `diagonal`, `cs`, `hcs`, `outside`, `fa`, `fa2`, `unstructured`); default `cs` for multi-environment trials, and `diagonal` for multiple populations
`VCPARAMETERS` = *string token*	Whether to re-estimate the variance-covariance model parameters (`estimate`, `fix`); default `esti`
`VCSELECT` = *string token*	Whether to re-select the variance-covariance model (`no`, `yes`); default `no`
`CRITERION` = *string token*	Criterion to use for model selection (`aic`, `sic`); default `sic`
`FIXED` = *formula*	Defines extra fixed effects
`UNITFACTOR` = *factor*	Saves the units factor required to define the random model when `UNITERROR` is to be used
`MVINCLUDE` = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (`explanatory`, `yvariate`); default `expl`, `yvar`
`MAXCYCLE` = *scalar*	Limit on the number of iterations; default 100
`WORKSPACE` = *scalar*	Number of blocks of internal memory to be set up for use by the `REML` algorithm; default 100

Parameters

`TRAIT` = *variates*	Quantitative trait to be analysed; must be set
`GENOTYPES` = *factors*	Genotype factor; must be set
`ENVIRONMENTS` = *factors*	Environment factor; must be set for a multi-environment trial
`POPULATIONS` = *factors*	Population factor; must be set for a multiple-population analysis
`UNITERROR` = *variates*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default `*` i.e. omitted
`VCINITIAL` = *pointers*	Initial values for the parameters of the variance-covariance model
`SELECTEDMODEL` = *texts*	`VCMODEL` setting for the selected covariance structure
`ADDITIVEPREDICTORS` = *pointers*	Additive genetic predictors; must be set
`ADD2PREDICTORS` = *pointers*	Second (paternal) set of additive genetic predictors
`DOMINANCEPREDICTORS` = *pointers*	

	Dominance genetic predictors
CHROMOSOMES = *factors*	Chromosomes corresponding to the genetic predictors; must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
IDLOCI = *texts*	Labels for the loci; must be set
MKLOCI = *variates*	Logical variate containing the value 1 if the locus is a marker, otherwise 0; must be set
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the genetic predictors
IDPARENTS = *texts*	Labels to identify the parents
QTLSELECTED = *variates*	Index numbers of the selected QTLs; must be set
INTERACTIONS = *variates*	Logical variate indicating whether each selected QTL has a significant (1) or non-significant (0) QTL-by-environment or QTL-by-population interaction
DOMSELECTED = *variates*	Logical variate indicating whether the dominance predictor of each selected QTL must be present (1) or absent (0) in the model
DOMINTERACTIONS = *variates*	Logical variate indicating whether the dominance-by-environment or dominance-by-population interaction of each selected QTL must be present (1) or absent (0) in the model
RESIDUALS = *variates*	Residuals from the analysis
FITTEDVALUES = *variates*	Fitted values from the analysis
WALDSTATISTICS = *variates*	Saves the Wald test statistics
PRWALD = *variates*	Saves the associated Wald probabilities
DFWALD = *variates*	Saves the degrees of freedom for the Wald test
QEFFECTS = *pointers*	Saves the estimated QTL effects
QSE = *pointers*	Saves the standard errors of the QTL effects
OUTFILENAME = *texts*	Name of the Genstat workbook file (`*.gwb`) to be created
QSAVE = *pointers*	Saves a pointer with information and results for the significant effects
SAVE = *REML save structures*	Save the details of each REML analysis for use in subsequent VDISPLAY and VKEEP directives

Description

QMESTIMATES fits a final QTL model to estimate QTL effects in a multi-environment trial or for multiple populations. The procedure uses means per genotype-environment or genotype-population combinations as phenotypic data, but weights can be attached to the means (see the UNITERROR parameter and the UNITFACTOR option below). The response variable must be specified by the TRAIT parameter, and the corresponding environment and genotype factors must be specified by the ENVIRONMENTS and GENOTYPES parameters, respectively. The POPULATIONTYPE option must be set to specify the population from which the genotypes are derived. For recombinant inbred lines (POPULATIONTYPE = RIL), the NGENERATIONS option, must be set to supply the number of generations. For backcross inbred lines (POPULATIONTYPE = BCxSy), the NBACKCROSSES and NSELFINGS options must be set to define the number of backcrosses to the first parent and the number of selfings, respectively. For a multiple-population analysis, the POPULATIONS parameter should be set (to a factor) instead of ENVIRONMENTS.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive

effects can be included in the model by specifying dominance genetic predictors using the DOMINANCEPREDICTORS parameter (e.g. in a F2 population). In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified, the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci must be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked. The IDPARENTS parameter can supply labels to identify the parents.

The QTL model assumes ENVIRONMENTS (or POPULATIONS) and QTLs as fixed terms, and GENOTYPES as a random term. The QTLSELECTED parameter must specify the set of QTLs, in the form of a variate containing the index number of the positions where the QTLs are located. The INTERACTIONS parameter supplies a logical variate containing zero if a QTL effect is constrained to be constant across environments (or populations), and one if it is specific for each environment (or population). When the DOMINANCEPREDICTORS parameter is set, the DOMSELECTED parameter supplies a logical variate containing one if the dominance predictor of the corresponding marker must be present in the model, and zero if the dominance predictor of the corresponding marker must be absent in the model. If DOMINANCEPREDICTORS is set but DOMSELECTED is not set, all the dominance predictors are included. Similarly, the DOMINTERACTIONS parameter supplies a logical variate containing one if the dominance-by-environment (or dominance-by-population) interaction of the corresponding marker must be present in the model, and zero if it must be absent. If DOMINANCEPREDICTORS is set but DOMINTERACTIONS is not set, all the dominance predictors are included.

Extra fixed effects can be defined by the FIXED option. A multi-Normal distribution, with vector mean 0 and variance covariance matrix Σ is assumed for the random genetic effects in the different environments (or populations). The VCMODEL option defines the model to use for Σ. The default assumes compound symmetry, but the VGESELECT procedure can be used to assess what model would be most suitable. Initial values for the parameters in the variance-covariance model can be specified by the VCINITIAL parameter. The VCPARAMETERS option controls whether the variance-covariance parameters are re-estimated at each step of the backward selection (VCPARAMETERS=estimate), or whether they are fixed at the defined initial values (VCPARAMETERS=fix). The VCSELECT option defines whether an extra check is made at each step on the variance-covariance model, to assess whether a simpler model is more suitable than the current model (based on the criterion defined by the CRITERION option). The SELECTEDMODEL parameter stores the final variance-covariance model that is selected.

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the other settings correspond to those in the PRINT option of the REML directive.

The QTL effects and their standard errors can be saved, in pointers, by the QEFFECTS and QSE parameters, respectively. These pointers have 2 levels of suffixes: the first level has 1, 2 or 3 values depending on the setting of the 3 possible predictors ADDITIVEPREDICTORS, ADD2PREDICTORS and DOMINANCEPREDICTORS; the second level has as many levels as the number of levels of the ENVIRONMENTS (or POPULATIONS) factor. The fitted values and residuals can be saved by the FITTEDVALUES and RESIDUALS parameters. The Wald statistics,

degrees of freedom and probabilities can be saved by the parameters WALDSTATISTICS, DFWALD and PRWALD, respectively.

The OUTFILENAME parameter can be used to save the Wald statistics and the QEFFECTS and QSE structures in a Genstat work book file in a sheet named STATISTICS. This parameter should not contain an extension as the extension is defined automatically as .gwb.

The QSAVE parameter can be used to save a pointer containing information and results for the significant QTLs. The elements of the pointer are labelled as follows to simplify their subsequent use:

'procedure'	stores the string 'QMESTIMATE' to indicate the source of the results,
'trait'	trait,
'markernames'	marker names,
'chromosomes'	chromosomes,
'positions'	positions,
'envnames'	names of the environments (or populations),
'waldstatistics'	wald statistics,
'prwald'	probability values of wald statistics,
'dfwald'	degrees of freedom of the wald statistics,
'qeffects'	QTL effects,
'qse'	standard errors of the QTL effects,
'%vexplained'	percentage variance explained,
'lowerci'	lower bound of confidence interval of estimated QTL position,
'upperci'	upper bound of confidence interval of estimated QTL position,
'posmin'	position of left flanking marker,
'posmax'	position of right flanking marker,
'idlfm'	marker name of left flanking marker,
'idrfm'	marker name of right flanking marker,
'posminci'	position of left flanking marker outside confidence interval,
'posmaxci'	position of right flanking marker outside confidence interval,
'idlfmci'	marker name of left flanking marker outside confidence interval,
'idrfmci'	marker name of right flanking marker outside confidence interval,
'locus'	index numbers of the significant QTLs, and
'neff'	number of additive and dominance predictors in the model.

The elements 'procedure', 'trait', 'markernames', 'chromosomes', 'envnames', 'idlfm', 'idrfm', 'idlfmci' and 'idrfmci' are text structures; 'positions', 'waldstatistics', 'prwald' and 'dfwald' are variates; 'qeffects' and 'qse' are pointers (see parameters QEFFECTS and QSE), as similarly are 'lowerci', 'upperci', 'posmin', 'posmax', 'posminci', 'posmaxci', 'idlfmci' and 'idrfmci'; 'neff' is a scalar.

The SAVE parameter can be used to save the REML save structure from the analysis for use with subsequent VKEEP and VDISPLAY directives.

Options: PRINT, POPULATIONTYPE, NGENERATIONS, NBACKCROSSES, NSELFINGS, VCMODEL, VCPARAMETERS, VCSELECT, CRITERION, FIXED, UNITFACTOR,MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: TRAIT, GENOTYPES, ENVIRONMENTS, POPULATIONS, UNITERROR, VCINITIAL, SELECTEDMODEL, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, IDMGENOTYPES, IDPARENTS, QTLSELECTED, INTERACTIONS, DOMSELECTED, DOMINTERACTIONS, RESIDUALS, FITTEDVALUES, WALDSTATISTICS, PRWALD, DFWALD, QEFFECTS, QSE, OUTFILENAME, QSAVE, SAVE.

Method

QMESTIMATE fits the following models, which include a set L of QTLs:

1) $y_{ij} = \mu + E_j + \Sigma_{l \in L}\, x_{il}^{add}\, \alpha_{jl}^{add} + GE_{ij}$
   if only ADDITIVEPREDICTORS are specified

2) $y_{ij} = \mu + E_j + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_{jl}^{add} + x_{il}^{dom}\, \alpha_{jl}^{dom}) + GE_{ij}$
   if DOMINANCEPREDICTORS are also specified

3) $y_{ij} = \mu + E_j + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_{jl}^{add} + x_{il}^{add2}\, \alpha_{jl}^{add2} + x_{il}^{dom}\, \alpha_{jml}^{dom}) + GE_{ij}$
   if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_{ij} is the trait value of genotype i in environment (or population) j, E_j is the environment (or population) main effect, x_{il}^{add} are the additive genetic predictors of genotype i for locus l, and α_{jl}^{add} are the associated effects. In models 2 and 3, x_{il}^{dom} are the dominance genetic predictors, and α_{jl}^{dom} are the associated effects. In model 3, x_{il}^{add} are the additive genetic predictors for maternal genotype i at locus l, x_{il}^{add2} are the additive genetic predictors for paternal genotype i, and α_{jl}^{add} and α_{jl}^{add2} are the associated effects. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). GE_{ij} is assumed to follow a multi-Normal distribution with mean vector 0, and a variance covariance matrix Σ, that can either be modelled explicitly (with an unstructured model) or by some parsimonious model (defined by option VCMODEL) as described in the VGESELECT procedure.

Action with RESTRICT

Restrictions are not allowed.

Reference

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QMBACKSELECT, QMQTLSCAN, QMVAF, QFLAPJACK, QREPORT, VGESELECT.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMKDIAGNOSTICS

Generates descriptive statistics and diagnostic plots of molecular marker data (D.A. Murray, S.J. Welham, M. Malosetti, M.P. Boer, L.C.P. Keizer & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, missingvalues, frequencies); default summ, miss, freq
PLOT = *string tokens*	What to plot (missingvalues, frequencies, probabilities, genotypes, map); default miss, geno, map
GEN%MISSING = *scalar*	Threshold for printing genotypes with many missing values (i.e. genotypes with a higher percentage of missing values than the specified value); default 10
MK%MISSING = *scalar*	Threshold for printing markers with many missing values (i.e. markers with a higher percentage of missing values than the specified value); default 10
MK%EXTREME = *scalar*	Threshold for printing markers with rare alleles (i.e. alleles present with a lower percentage than the specified threshold); default 10
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP, AMP); must be set
NGENERATIONS = *scalar*	Number of generations for a RIL population; default 6
NBACKCROSSES = *scalar*	Number of backcrosses; must be set for a BCxSy population
NSELFINGS = *scalar*	Number of selfings; must be set for a BCxSy population
DCHROMOSOMES = *variate*, *text* or *scalar*	Specifies a subset of the linkage groups to be displayed
PDIRECTION = *string token*	How to sort the probabilities when PRINT=frequencies with BC1, DH1, F2, RIL and BCxSy populations (ascending, descending); default * i.e. no sorting

Parameters

MKSCORES = *pointers*	Genotype codes for each marker; must be set
CHROMOSOMES= *factors*	Linkage groups for the markers; must be set
POSITIONS = *variates*	Positions within the linkage groups of markers; must be set
MKNAMES = *texts*	Marker name; must be sets
IDMGENOTYPES = *texts*	Labels for genotypes corresponding to the marker scores
PARENTS = *pointers*	Parent information
IDPARENTS = *texts*	Labels to identify the parents
GENCHECK = *variates*	Logical variates containing the value one for genotypes with missing value problems, according to the setting of the GEN%MISSING option, and zero otherwise
MKCHECK = *variates*	Logical variates containing the value one for markers with missing or extreme value problems, as defined by the MK%MISSING and MK%EXTREME options, and zero otherwise
SUMMARY = *pointers*	Saves a summary of counts and probabilities for the chi-square tests for BC1, DH1, F2, RIL and BCxSy populations

Description

QMKDIAGNOSTICS generates descriptive statistics and diagnostic plots of molecular marker data. The marker scores data must be supplied in a pointer by the MKSCORES pointer. The length of the MKSCORES pointer must be equal to the number of markers, and each structure of the pointer must be a factor with labels. The population type must be specified by the POPULATIONTYPE option. For a RIL population, the number of generations is specified by the NGENERATIONS option; default 6. For a BCxSy population, the number of backcrosses and the number of selfings are supplied by the NBACKCROSSES and NSELFINGS options, respectively.The labels for the genotypes corresponding to the marker scores can be supplied by the IDMGENOTYPES parameter.

The corresponding map information for the markers must be supplied by the CHROMOSOMES and POSITIONS parameters, and the labels of the markers must be supplied by the MKNAMES parameter.

The parent information must be supplied using the PARENTS parameter in a pointer to a set of texts. The first text in the pointer defines the alleles for parent 1, the second text defines the allele for parent 2, and so on. The labels for the parents are supplied in a text using the IDPARENTS parameter.

The PRINT option controls printed output, with settings:

summary	to print the number of genotypes and markers, and summary statistics per chromosome,
missingvalues	to print the genotypes with percentages of missing values GEN%MISSING and the markers with percentages of missing values greater than MK%MISSING,
frequencies	to print the allele frequencies of all markers with allele frequencies greater than MK%EXTREME for for an AMP population, or the frequencies of genotype codes for markers for BC1, DH1, F2, RIL and BCxSy populations.

By default PRINT = summary, missingvalues, frequencies. If PRINT=frequencies or PLOT=probabilities, the output for BC1, DH1, F2, RIL and BCxSy populations includes the probabilities of the calculated chi-square tests of Mendelian segregation; the expected ratios are defined in the *Method* Section. The summary table of genotypic code frequencies can be sorted into ascending or descending order of probabilities by setting the PDIRECTION option.

The PLOT option controls graphical output, with settings:

missingvalues	to produce a trellis plot of percentages of missing values against the map position for each linkage group and a plot of missing marker scores using the DQMKSCORES procedure,
frequencies	to produce a trellis plot of the allele frequency percentages against the map position for each linkage group (for AMP population only),
probabilities	to produce a trellis plot of the chi-square probabilities, plotted on a −log10 scale against the map position for each linkage group (for BC1, DH1, F2, RIL and BCxSy populations only),
genotypes	to plot all graphical genotypes, and
map	to plot the linkage map.

By default PLOT = missingvalues, genotypes, map.

The DCHROMOSOMES option can be used to select a subset of the linkage groups to display. The setting can be either a variate or scalar to define a subset using the levels of the CHROMOSOMES factor, or a text to define a subset using its labels.

The GENCHECK parameter can save a logical variate identifying the genotypes that have less

(with values of zero) or more (with values of one) than the required number of missing values, based on the setting of the GEN%MISSING option. Similarly the MKCHECK parameter can save a logical variate identifying the markers that have problems of missing or extreme values, according to the settings of the MK%MISSING and MK%EXTREME options.

The SUMMARY parameter can save a pointer containing the structures that are printed when PRINT=frequencies for F2, BC1, DH1 and RIL populations. This contains the marker number, the marker name, the chromosome number, the position on the chromosome, percentage missing, the allele frequencies and the chi-square probability.

Options: PRINT, PLOT, GEN%MISSING, MK%MISSING, MK%EXTREME, POPULATIONTYPE, NGENERATIONS, NBACKCROSSES, NSELFINGS, DCHROMOSOMES, PDIRECTION.
Parameters: MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, IDMGENOTYPES, PARENTS, IDPARENTS, GENCHECK, MKCHECK, SUMMARY.

Method
For markers the segregation is evaluated against the expected allele frequencies using a chi-square test. The frequencies are as follows:

Population	Alleles	Expected ratio
BC1	1/1 : 1/2	1 : 1
DH1	1/1 : 2/2	1 : 1
F2	1/1 : 1/2 : 2/2	1 : 2 : 1
	1/1 : 2/–	1 : 3
	2/2 : 1/–	1 : 3
RILn	1/1 : 1/2 : 2/2	$2^{n-1}-1 : 2 : 2^{n-1}-1$
	1/1 : 2/–	$2^{n-1}-1 : 2^{n-1}+1$
	2/2 : 1/–	$2^{n-1}-1 : 2^{n-1}+1$
BCxSy	1/1 : 1/2 : 2/2	$2^{x+y+1}-2^{y}-1 : 2 : 2^{y}-1$
	1/1 : 2/–	$2^{x+y+1}-2^{y}-1 : 2^{y}+1$
	2/2 : 1/–	$2^{x+y+1}-2^{y}-1 : 2^{y}-1$

where 1 is the allele for parent 1, 2 is the allele for parent 2, n is the number of RIL generations, and x and y are the number of backcrosses and selfings, respectively, for a BCxSy population.

Action with RESTRICT
Restrictions are not allowed.

See also
Procedures: DQMAP, DQMKSCORES, DQMQTLSCAN, DQSQTLSCAN, QMKRECODE.
Genstat Reference Manual 1 Summary sections on: Statistical genetics and QTL estimation, Graphics.

QMKRECODE

Recodes marker scores into separate alleles (L.C.P. Keizer & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (alleles, summary); default alle
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP, AMP); must be set
MISSING = *text*	Character representing a missing genotype; default '-'
USEFIRSTGENOTYPE = *string token*	Makes all the first (and second) labels of the LABALLELES pointer from the first genotype of the population (yes, no); default no
ASEPARATOR = *text*	Character separating allele values; default '/'

Parameters

MKSCORES = *pointers*	Marker scores; must be set
MKALLELES = *pointers*	Saves the marker scores per allele
LABALLELES = *pointers*	Saves the allele labels
MKLABALLELES = *pointers*	Saves the allele labels per marker
NALLELES = *variates*	Saves the number of alleles per marker
MKNAMES = *texts*	Names of the markers

Description

QMKRECODE recodes the marker scores, specified by the MKSCORES parameter, into separate alleles. These separate alleles can be saved by the MKALLELES parameter, in a pointer with two levels of suffixes. The first level has an element for each marker names; the second level has as many elements as the number of alleles of the marker. The labels of the alleles per marker can be saved with the LABALLELES parameter.

If you set option USEFIRSTGENOTYPE=yes, all the first LABALLELES correspond to the first individual in the MKSCORES pointer (for association analysis). The number of the alleles per marker can be saved by the NALLELES, and the names of the markers by the MKNAMES parameter (from the labels of the MKSCORES pointer). MKLABALLELES is similar to the LABALLELELS but inverted. So it is not a pointer for alleles per marker (with elements of unequal length). Instead it is a pointer that contains the first up to maximum number of alleles for all markers, and missing positions where there were less alleles. This may be more convenient for printing and reporting.

The type of population must specified using the POPULATIONTYPE option. The MISSING option specifies a character to identify missing genotypes (default '-'). Genstat expects the alleles for each genotype to be separated using a '/' character, but an alternative can be supplied using the ASEPARATOR option.

The PRINT option controls the printed output, with settings:

alleles	for details of the alleles, and
summary	for a general summary.

By default PRINT=alleles.

Options: PRINT, POPULATIONTYPE, MISSING, ASEPARATOR.
Parameters: MKSCORES, MKALLELES, LABALLELES, MKLABALLELES, NALLELES, MKNAMES.

Action with RESTRICT

Restrictions are not allowed.

See also
Procedure: QMKDIAGNOSTICS.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMKSELECT

Obtains a representative selection of markers by means of genetic distance sampling or genetic distance optimization (J. Jansen & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, monitoring); default summ
NCLUSTERS = *scalar*	The number of markers to be selected; must be set
METHOD = *string token*	Method to be used (sampling, optimization); default samp

Parameters

MKNAMES = *texts*	Names of the markers; must be set
RECFREQUENCY = *symmetric matices*	Input recombination frequencies matrix for each selection; must be set
PRIORGROUPS = *factors*	Defines prior groupings of the markers
SELECTED = *variates*	Logical variate indicating whether a marker is selected (1) as cluster centre or not (0)
NEIGHBOURS = *variates*	Saves the nearest cluster centres of the markers
DISTANCES = *variates*	Saves the distances of the markers to the nearest cluster centre
SEED = *scalars*	Seed for randomization at the start; default 0

Description

QMKSELECT selects a representative subset of markers using a matrix of recombination frequencies, provided by the RECFREQUENCY parameter.

The METHOD option specifies whether to use genetic distance sampling or genetic distance optimization, by setting it to one of the following settings:

sampling	genetic distance sampling using the method of Jansen & Van Hintum (2006), or
optimization	genetic distance optimization based on K-medoids cluster analysis (Kaufman & Rouseeuw 1990).

The default is METHOD=sampling.

The marker names must be supplied by the MKNAMES parameter, and the number of markers to be selected must be specified by the NCLUSTERS option. Prior information about the grouping of the markers can be supplied using the PRIORGROUPS factor.

The SEED parameter specifies the seed to use to randomize the markers at the start. The default value of zero continues an existing sequence, or (if none) initializes the seed automatically.

The marker selection can be saved by the SELECTED parameter, in a logical variate containing one for each marker selected as a cluster centre, and zero for the markers that are not selected. The NEIGHBOURS parameter saves the nearest cluster centre for each marker, and the DISTANCES parameter saves the distances of each marker to the nearest cluster centre.

The PRINT option controls the printed output, with settings:

summary	for a summary of the selection, and
monitoring	for monitoring information.

Options: PRINT, NCLUSTERS, METHOD.

Parameters: MKNAMES, RECFREQUENCY, PRIORGROUPS, SELECTED, NEIGHBOURS, DISTANCES, SEED.

Action with `RESTRICT`
Restrictions are not allowed.

References
Jansen, J. & Th.J.L. van Hintum (2006). Genetic distance sampling: a novel sampling method for obtaining core collections using genetic distances with an application to cultivated lettuce. *Theor. Appl. Genet.*, **114**, 421-428.

Kaufman, P. & P.J. Rousseuw (1990). Finding Groups in Data: an Introduction to Cluster Analysis. Wiley, New York.

See also
Procedure: `QGSELECT`.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMQTLSCAN

Performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in multi-environment trials or multiple populations (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

`PRINT` = *string tokens*	What to print (`summary`, `progress`, `model`, `components`, `effects`, `means`, `stratumvariances`, `monitoring`, `vcovariance`, `deviance`, `Waldtests`, `missingvalues`, `covariancemodels`); default `summ`
`PLOT` = *string token*	Whether to plot the profile along the genome (`profile`); default `prof`
`POPULATIONTYPE` = *string token*	Type of population (`BC1`, `DH1`, `F2`, `RIL`, `BCxSy`, `CP`); must be set
`ALPHALEVEL` = *scalar*	Defines a genome-wide significance level to calculate the threshold; default 0.05
`VCMODEL` = *string token*	Specifies the variance-covariance model for the set of environments or populations (`identity`, `diagonal`, `cs`, `hcs`, `outside`, `fa`, `fa2`, `unstructured`); default `cs` for multi-environment trials, and `diagonal` for multiple populations
`VCPARAMETERS` = *string token*	Whether to re-estimate the variance-covariance model parameters (`estimate`, `fix`); default `esti`
`QTLMODEL` = *string token*	Type of QTL model (`q`, `qqe`); default `qqe`
`COFACTORS` = *variate*	Index numbers of loci to be used as cofactors for the genetic background
`COFWINDOW` = *scalar*	Specifies a window for cofactor exclusion from the model; default 10^6 which means that all cofactors on the same chromosomes are excluded
`THRMETHOD` = *string token*	Which method to use to calculate the threshold for QTL detection (`bonferroni`, `liji`, `given`); default `liji`
`THRESHOLD` = *scalar*	Threshold value for test statistic when `THRMETHOD=given`
`DISTANCE` = *scalar*	Distance between loci when `THRMETHOD=bonferroni`; default 4
`FIXED` = *formula*	Formula with extra fixed terms
`UNITFACTOR` = *factor*	Saves the units factor required to define the random model when `UNITERROR` is to be used
`STATISTICTYPE` = *string token*	Which test statistic to plot and save using the `STATISTICS` parameter (`wald`, `minlog10p`); default `minl`
`COLOURS` = *scalar*, *variate* or *text*	Colours to use for the chromosomes; default `*` uses the colours of pens 1, 2 up to the number of chromosomes
`TITLE` = *text*	General title for the plot
`YLOWERTITLE` = *text*	Title for the y-axis of the lower graph; default `'Environments'` for multi-environment trials, and `'Populations'` for multiple populations
`YUPPERTITLE` = *text*	Title for the y-axis of the upper graph; default uses the identifier of the `STATISTICS` variate or pointer
`XTITLE` = *string*	Title for the x-axis; default `'Chromosomes'`
`MVINCLUDE` = *string tokens*	Whether to include units with missing values in the

	explanatory factors and variates and/or the y-variates (`explanatory`, `yvariate`); default `expl`, `yvar`
`MAXCYCLE` = *scalar*	Limit on the number of iterations; default 100
`WORKSPACE` = *scalar*	Number of blocks of internal memory to be set up for use by the `REML` algorithm; default 100

Parameters

`TRAIT` = *variates*	Quantitative trait to be analysed; must be set
`GENOTYPES` = *factors*	Genotype factor; must be set
`ENVIRONMENTS` = *factors*	Environment factor; must be set for a multi-environment trial
`POPULATIONS` = factors	Population factor; must be set for a multiple-population analysis
`UNITERROR` = *variate*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default `*` i.e. omitted
`VCINITIAL` = *pointers*	Initial values for the parameters ofthe variance-covariance model
`ADDITIVEPREDICTORS` = *pointers*	Additive genetic predictors; must be set
`ADD2PREDICTORS` = *pointers*	Second (paternal) set of additive genetic predictors
`DOMINANCEPREDICTORS` = *pointers*	Dominance genetic predictors
`CHROMOSOMES` = *factors*	Chromosomes corresponding to the genetic predictors; must be set
`POSITIONS` = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
`IDLOCI` = *texts*	Labels for the loci
`IDMGENOTYPES` = *texts*	Labels for the genotypes corresponding to the genetic predictors
`IDEFFECTS` = *texts*	Labels for the effects along the y-axis, in the frame below the profile plot
`IDPARENTS` = *texts*	Labels to use to identify the parents
`QSTATISTICS` = *variates*	Saves test statistics for QTL effects along the genome
`QEFFECTS` = *pointers*	Saves QTL effects along the genome
`QSE` = *pointers*	Saves standard errors of the QTL effects
`OUTFILENAME` = *texts*	Name of the Genstat workbook file (`*.gwb`) to be created
`DFILENAME` = *texts*	Name of the graphics file for the plots

Description

`QMQTLSCAN` performs a genome-wide QTL scan in multi-environment trials as described by Malosetti *et al*. (2004) and Boer *et al*. (2007). Alternatively, it can analyse data from multiple populations. It uses means per genotype-environment or genotype-population combinations as phenotypic data, but weights can be attached to the means (see the `UNITERROR` parameter and the `UNITFACTOR` option below). The response variable must be specified by the `TRAIT` parameter, and the corresponding environment and genotype factors must be specified by the `ENVIRONMENTS` and `GENOTYPES` parameters, respectively. The `POPULATIONTYPE` option must be set to specify the population type. For a multiple-population analysis, the `POPULATIONS` parameter should be set (to a factor) instead of `ENVIRONMENTS`.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the `ADDITIVEPREDICTORS` parameter. Non-additive

effects can be included in the model by specifying dominance genetic predictors using the DOMINANCEPREDICTORS parameter (e.g. in a F2 population). In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified, the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked.

The QTL detection model assumes ENVIRONMENTS (or POPULATIONS) as a fixed term, and GENOTYPES as a random term. Extra fixed effects can be specified using the FIXED option. For the random genetic effects in the different environments (or populations) a multi-Normal distribution is assumed with mean vector 0 and variance-covariance matrix Σ. The VCMODEL option defines the model to use for Σ; the default for a multi-environment trial is to take compound symmetry, while for a multiple-population analysis the default is to take a diagonal variance matrix (the best model can be selected using the VGESELECT procedure). Initial values for the parameters in the variance-covariance model can be defined by the VCINITIAL parameter. The VCPARAMETERS option controls whether variance-covariance parameters are re-estimated at each iteration (VCPARAMETERS=estimate), or whether they are fixed at the initial values (VCPARAMETERS=fix). The fix setting can be useful to save computation time with large data sets or with more complex models.

By default the QTL model includes a separate QTL effect in every environment (or population), but it is possible to search for QTLs based only on QTL main effects by setting option QTLMODEL=q. The QTL search can be performed with cofactors to control for genetic background effects (*Composite Interval Mapping*) or without cofactors (*Simple Interval Mapping*). For Composite Interval Mapping, the COFACTORS option must be set to a variate containing the index numbers of the loci designated as cofactors. The COFWINDOW option defines a window around a tested position within which cofactors are temporarily excluded from the model.

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The method to define the threshold value is defined by the THRMETHOD option and uses a genome-wide error rate defined by the option ALPHALEVEL (default 0.05). If THRMETHOD=given, a user-defined threshold value must be specified using the THRESHOLD option. If THRMETHOD=bonferroni, an effective number of tests is calculated using the value specified by the DISTANCE option as the step size (default 4). Alternatively the liji setting uses the method described by Li & Ji (2005). See procedure QTHRESHOLD for details.

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the progress setting shows how the scan is progressing. The other settings correspond to those in the PRINT option of the REML directive.

By default QMQTLSCAN produces a pair of graphs: the upper one plots the test statistic associated with the effects of the genetic predictors against their position on the chromosomes, and the lower one is a heat plot showing how the statistic changes over the environments (or populations). You can suppress the plotting by setting option PLOT=*. The STATISTICTYPE option specifies what to plot along the y-axis of the upper plot, either the test statistic or the associated probability value (on a −log10 scale), and also defines what is saved in the variates

specified by the QSTATISTICS parameter. The IDEFFECTS parameter can be used to label the effects, and the IDPARENTS parameter can supply labels to identify the parents.

The effects of each genetic predictor and their standard errors can be saved, in pointers, by the QEFFECTS and QSE parameters, respectively. These pointers have 2 levels of suffixes: the first level has 1, 2 or 3 values depending on the setting of the 3 possible predictors ADDITIVEPREDICTORS, ADD2PREDICTORS and DOMINANCEPREDICTORS; the second level has as many levels as the number of levels of the ENVIRONMENTS (or POPULATIONS) factor.

The TITLE, YLOWERTITLE, YUPPERTITLE and XTITLE options can specify the general title of the graph, the title of the y-axis on the lower graph(s), the title of the y-axis on the upper graph, and the title of the x-axis, respectively. The colours to use for the chromosomes in the upper graph are specified by the COLOURS option using either a text of colour names or a variate of RGB values (see the PEN directive for details). If COLOURS is not set, the default is to use the default colours of the pens 1, 2, onwards, up to the number of chromosomes. By default, the plot is sent to the screen. However, you can supply a file for the plot, using the DFILENAME parameter. You can discover the types of graphics file that are supported by running the command.

```
DHELP possible
```

The OUTFILENAME parameter can be used to write the QSTATISTICS, QEFFECTS and QSE structures to a Genstat work book file in a sheet named STATISTICS. This parameter should not contain an extension as the extension is defined automatically given as .gwb.

Options: PRINT, PLOT, POPULATIONTYPE, ALPHALEVEL, VCMODEL, VCPARAMETERS, QTLMODEL, COFACTORS, COFWINDOW, THRMETHOD, THRESHOLD, DISTANCE, FIXED, UNITFACTOR, STATISTICTYPE, COLOURS, TITLE, YLOWERTITLE, YUPPERTITLE, XTITLE, YLABEL, MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: TRAIT, GENOTYPES, ENVIRONMENTS, POPULATIONS, UNITERROR, VCINITIAL, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, IDMGENOTYPES, IDEFFECTS, IDPARENTS, QSTATISTICS, QEFFECTS, QSE, OUTFILENAME, DFILENAME.

Method

QMQTLSCAN fits the following mixed models repeatedly along the genome:

1) $y_{ij} = \mu + E_j + \Sigma_{f \in F} x_{il}^{add} c_{jf}^{add} + x_i^{add} \alpha_j^{add} + GE_{ij}$

if only ADDITIVEPREDICTORS are specified

2) $y_{ij} = \mu + E_j + \Sigma_{f \in F} (x_{if}^{add} c_{jf}^{add} + x_{if}^{dom} c_{jf}^{dom}) + (x_i^{add} \alpha_j^{add} + x_i^{dom} \alpha_j^{dom}) + GE_{ij}$

if DOMINANCEPREDICTORS are also specified

3) $y_{ij} = \mu + E_j + \Sigma_{f \in F} (x_{if}^{add} c_{jf}^{add} + x_{if}^{add2} c_{jf}^{add2} + x_{if}^{dom} c_{jf}^{dom})$
$+ (x_i^{add} \alpha_j^{add} + x_i^{add2} \alpha_j^{add2} + x_i^{dom} \alpha_j^{dom}) + GE_{ij}$

if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_{ij} is the trait value of genotype i in environment (or population) j, E_j is the environmental (or population) main effect, F is a set of cofactors (if cofactors are included in the model), and x_{if}^{add} and x_i^{add} are the additive genetic predictors of genotype i at the cofactor positions and at the tested position, respectively. The associated effects are denoted by c_{jf}^{add} and α_j^{add} for cofactors and tested position respectively. In model 2 and 3, x_{if}^{dom} and x_i^{dom} are dominance genetic predictors of genotype i at the cofactor positions and at the tested position, respectively, with associated effects c_{jf}^{dom}, and α_j^{dom}. In model 3, x_{if}^{add} and x_i^{add} are the additive genetic predictors for the maternal genotype, for cofactors and tested position, respectively, and x_{if}^{add2} and x_i^{add2} are the equivalent additive genetic predictors for the paternal genotype. Finally x_{if}^{dom} and x_i^{dom} are the dominance genetic predictors for the cofactors and tested position, respectively. The associated

effects are given by c_{jf}^{add}, c_{jf}^{add2} and c_{jf}^{dom} for cofactors, and α_j^{add}, α_j^{add2} and α_j^{dom} for tested positions. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). The residual unexplained genetic and environmental (or population) effects are modelled by the GE_{ij} term, which is assumed to follow a multi-Normal distribution with mean vector 0, and a variance covariance matrix Σ. The matrix Σ can either be modelled explicitly (with an unstructured model) or by some parsimonious models (defined by option VCMODEL) as described in the VGESELECT procedure.

The procedure uses the REML directive iteratively to fit the model at each chromosome position, storing the Wald statistic for hypothesis testing. The resulting Wald statistic or the associated probability value (on a −log10 scale) can be plotted to produce the well-known profile plots along the chromosomes.

Action with RESTRICT

Restrictions are not allowed.

References

Boer, M.P., Wright, D., Feng, L., Podlich, D.W., Luo, L., Cooper, M. & van Eeuwijk, F.A. (2007). A mixed-model quantitative trait loci (QTL) analysis for multiple-environment trial data using environmental covariables for QTL-by-environment interactions, with an example in maize. *Genetics*, **177**, 1801-1813.

Malosetti, M., Voltas, J., Romagosa, I., Ullrich, S.E. & van Eeuwijk, F.A. (2004). Mixed models including environmental covariables for studying QTL by environment interaction. *Euphytica*, **137**, 139-145.

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QMBACKSELECT, QMESTIMATE, QMVAF, VGESELECT.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMTBACKSELECT

Performs a QTL backward selection for loci in multi-trait trials (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default summ
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
ALPHALEVEL = *scalar*	Defines a significance level; default 0.05
VCMODEL = *string token*	Defines the variance-covariance model for the set of traits (identity, diagonal, cs, hcs, outside, fa, fa2, unstructured); default cs
VCPARAMETERS = *string token*	Whether to re-estimate the variance-covariance model parameters (estimate, fix); default esti
VCSELECT = *string token*	Whether to re-select the variance-covariance model (no, yes); default no
STANDARDIZE = *string token*	How to standardize the traits (none, normalize) ; default norm
CRITERION = *string token*	Criterion to use for model selection (aic, sic); default sic
FIXED = *formula*	Defines extra fixed effects
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 100

Parameters

Y = *variates*	Quantitative traits to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
FTRAITS = *factors*	Factor indicating the trait of each y-value; must be set
UNITERROR = *variates*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted
VCINITIAL = *pointers*	Initial values for the parameters of the variance-covariance model
SELECTEDMODEL = *texts*	VCMODEL setting for the selected covariance structure
ADDITIVEPREDICTORS = *pointers*	Additive genetic predictors; must be set
ADD2PREDICTORS = *pointers*	Second (paternal) set of additive genetic predictors
DOMINANCEPREDICTORS = *pointers*	Dominance genetic predictors
CHROMOSOMES = *factors*	Chromosomes corresponding to the genetic predictors; must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set

IDLOCI = *texts*	Labels for the loci
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the genetic predictors
QTLCANDIDATES = *variates*	Specifies the locus index numbers from which to start the selection; must be set
QTLSELECTED = *variates*	Saves the index numbers of the selected QTLs
INTERACTIONS = *variates*	Saves a logical variate indicating whether each selected QTL showed a significant (1) or non-significant (0) QTL-by-trait interaction
DOMSELECTED = *variates*	Saves a logical variate indicating whether each selected QTL showed a significant (1) or non-significant (0) effect of the DOMINANCEPREDICTORS
DOMINTERACTIONS = *variates*	Saves a logical variate indicating whether each selected QTL showed a significant (1) or non-significant (0) dominance-by-trait interaction
WALDSTATISTICS = *variates*	Saves the Wald test statistics
PRWALD = *variates*	Saves the associated Wald probabilities

Description

QMTBACKSELECT selects QTLs by backward selection from a list of candidate QTLs (loci) in multi-trait trials. It uses means per genotype-trait combinations as phenotypic data, but weights can be attached to the means (see the UNITERROR parameter and the UNITFACTOR option below). The response variable must be specified by the Y parameter, and the corresponding trait and genotype factors must be specified by the FTRAITS and GENOTYPES parameters, respectively. The POPULATIONTYPE option must be set to specify the population from which the genotypes have been derived. By default, the values of each trait are standardized by dividing them by their standard deviation, but you can set option STANDARDIZE=none to suppress this.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive effects can be included in the model by specifying dominance genetic predictors using the DOMINANCEPREDICTORS parameter (e.g. in a F2 population). In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified, the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked.

The set of candidate QTLs must be supplied by the QTLCANDIDATES parameter. The model assumes FTRAITS as a fixed term, and GENOTYPES as a random term. Extra fixed effects can be defined using the FIXED option. A multi-Normal distribution is assumed for the random genetic effects, with mean vector 0 and variance-covariance matrix $\mathbf{\Sigma}$. The VCMODEL option defines the model to use for $\mathbf{\Sigma}$. See the VGESELECT procedure for details of the available models; the default is to use compound symmetry. Initial values for the parameters in the variance-covariance model can be specified by the VCINITIAL parameter. The VCPARAMETERS option controls whether the variance-covariance parameters are re-estimated at each step of the backward selection (VCPARAMETERS=estimate), or whether they are fixed at the defined initial values (VCPARAMETERS=fix). The VCSELECT option defines whether an extra check is made at each step on the variance-covariance model, to assess whether a simpler model is more suitable than the current model (based on the criterion defined by the CRITERION option). The SELECTEDMODEL parameter stores the final variance-covariance model that is selected. The

significance level to use at each step of the backward selection process is given by the ALPHALEVEL option (default 0.05).

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the other settings correspond to those in the PRINT option of the REML directive.

The list of selected QTLs can be saved by the QTLSELECTED parameter, and a logical variate that indicates whether the selected QTL showed a significant QTL-by-trait interaction can be saved by the INTERACTIONS parameter. This interaction is the combined effect of the ADDITIVEPREDICTORS, ADD2PREDICTORS and DOMINANCEREDICTORS pointers if specified. After the final step of the backward selection, extra tests are performed if the DOMINANCEPREDICTORS parameter is set. If the selected QTL has no interaction effect with trait, a test is performed of whether the dominance effect has a significant contribution in the combined QTL effect. If dominance is significant, the corresponding units of the logical variate saved by the DOMSELECTED parameter are set to one; the other units are set to zero. If the selected QTL has significant interaction with trait, a test is performed of whether the dominance-by-trait interaction has a significant contribution in the combined QTL-by-trait interaction. If the dominance-by-trait interaction is significant, the corresponding units of the logical variate saved by DOMINTERACTIONS parameter are set to one; the other units are set to zero. The Wald test and associated probability values for the combined effects (including the possible not-significant dominance and dominance-by-trait interactions) of the selected QTLs can be saved by the WALDSTATISTICS and PRWALD parameters, respectively.

Options: PRINT, POPULATIONTYPE, ALPHALEVEL, VCMODEL, VCPARAMETERS, VCSELECT, CRITERION, FIXED, UNITFACTOR, MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: Y, GENOTYPES, FTRAITS, UNITERROR, VCINITIAL, SELECTEDMODEL, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, IDMGENOTYPES, QTLCANDIDATES, QTLSELECTED, INTERACTIONS, DOMSELECTED, DOMINTERACTIONS, WALDSTATISTICS, PRWALD.

Method

QMTBACKSELECT starts with the following mixed models, which include a set L of candidate QTLs:

1) $y_{ij} = \mu + T_j + \Sigma_{l \in L}\, x_{il}^{add}\, \alpha_{jl}^{add} + GT_{ij}$
   if only ADDITIVEPREDICTORS are specified

2) $y_{ij} = \mu + T_j + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_{jl}^{add} + x_{il}^{dom}\, \alpha_{jl}^{dom}) + GT_{ij}$
   if DOMINANCEPREDICTORS are also specified

3) $y_{ij} = \mu + T_j + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_{jl}^{add} + x_{il}^{add2}\, \alpha_{jl}^{add2} + x_{il}^{dom}\, \alpha_{jl}^{dom}) + GT_{ij}$
   if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_{ij} is the value of trait j for genotype i, T_j is the trait main effect, x_{il}^{add} are the additive genetic predictors of genotype i for locus l, and α_{jl}^{add} are the associated effects. In models 2 and 3, x_{il}^{dom} are the dominance genetic predictors, and α_{jl}^{dom} are the associated effects. In model 3, x_{il}^{add} are the additive genetic predictors for maternal genotype i at locus l, x_{il}^{add2} are the additive genetic predictors for paternal genotype i, and α_{jl}^{add} and α_{jl}^{add2} are the associated effects. Genetic

predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). GT_{ij} is assumed to follow a multi-Normal distribution with mean vector 0, and a variance covariance matrix Σ, that can either be modelled explicitly (with an unstructured model) or by some parsimonious model (defined by option VCMODEL) as described in the VGESELECT procedure.

The backward selection procedure starts with the initial set of loci (defined by the QTLCANDIDATES parameter), and checks whether all loci are significant. If not, the locus with the lowest Wald test statistic is dropped from the model. This process is repeated until all loci in the model are significant. The procedure then switches to test whether the remaining QTLs show significant QTL-by-trait interaction, by breaking down the QTL effects into QTL main effects and QTL-by-trait interaction effects. If the QTL-by-trait interaction term is not significant, only a main effect is retained in the model for the corresponding QTL.

Action with RESTRICT

Restrictions are not allowed.

Reference

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QMTESTIMATE, QMTQTLSCAN, QMVAF, VGESELECT.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMTESTIMATE

Calculates QTL effects in multi-trait trials (M.P Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default summ
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
NGENERATIONS = *scalar*	Number of generations of selfing for a RIL population
NBACKCROSSES = *scalar*	Number of backcrosses for a BCxSy population
NSELFINGS = *scalar*	Number of selfings for a BCxSy population
VCMODEL = *string token*	Specifies the variance-covariance model for the set of traits (identity, diagonal, cs, hcs, outside, fa, fa2, unstructured); default cs
VCPARAMETERS = *string token*	Whether to re-estimate the variance-covariance model parameters (estimate, fix); default esti
VCSELECT = *string token*	Whether to re-select the variance-covariance model (no, yes); default no
STANDARDIZE = *string token*	How to standardize the traits (none, normalize) ; default norm
CRITERION = *string token*	Criterion to use for model selection (aic, sic); default sic
FIXED = *formula*	Defines extra fixed effects
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 100

Parameters

Y = *variates*	Quantitative traits to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
FTRAITS = *factors*	Factor indicating the trait of each y-value; must be set
UNITERROR = *variate*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted
VCINITIAL = *pointers*	Initial values for the parameters of the variance-covariance model
SELECTEDMODEL = *texts*	VCMODEL setting for the selected covariance structure
ADDITIVEPREDICTORS = *pointers*	Additive genetic predictors; must be set
ADD2PREDICTORS = *pointers*	Second (paternal) set of additive genetic predictors
DOMINANCEPREDICTORS = *pointers*	Dominance genetic predictors
CHROMOSOMES = *factors*	Chromosomes corresponding to the genetic predictors; must be set

POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
IDLOCI = *texts*	Labels for the loci; must be set
MKLOCI = *variates*	Logical variate containing the value 1 if the locus is a marker, otherwise 0; must be set
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the genetic predictors
IDPARENTS = *texts*	Labels to identify the parents
QTLSELECTED = *variates*	Index numbers of the selected QTLs; must be set
INTERACTIONS = *variates*	Logical variate indicating whether each selected QTL has a significant (1) or non-significant (0) QTL-by-trait interaction
DOMSELECTED = *variates*	Logical variate indicating whether the dominance predictor of each selected QTL must be present (1) or absent (0) in the model
DOMINTERACTIONS = *variates*	Logical variate indicating whether the dominance-by-trait interaction of each selected QTL must be present (1) or absent (0) in the model
RESIDUALS = *variates*	Residuals from the analysis
FITTEDVALUES = *variates*	Fitted values from the analysis
WALDSTATISTICS = *variates*	Saves the Wald test statistics
PRWALD = *variates*	Saves the associated Wald probabilities
DFWALD = *variates*	Saves the degrees of freedom for the Wald test
QEFFECTS = *pointers*	Saves the estimated QTL effects
QSE = *pointers*	Saves the standard errors of the QTL effects
OUTFILENAME = *texts*	Name of the Genstat workbook file (`*.gwb`) to be created
QSAVE = *pointers*	Saves a pointer with information and results for the significant effects
SAVE = *REML save structures*	Save the details of each REML analysis for use in subsequent VDISPLAY and VKEEP directives

Description

QMTESTIMATES fits a final QTL model to estimate QTL effects in a multi-trait trial. The procedure uses means per genotype-trait combinations as phenotypic data, but weights can be attached to the means (see the UNITERROR parameter and the UNITFACTOR option below). The response variable must be specified by the Y parameter, and the corresponding trait and genotype factors must be specified by the FTRAITS and GENOTYPES parameters, respectively. The POPULATIONTYPE option must be set to specify the population from which the genotypes are derived. For recombinant inbred lines (POPULATIONTYPE = RIL), the NGENERATIONS option, must be set to supply the number of generations. For backcross inbred lines (POPULATIONTYPE = BCxSy), the NBACKCROSSES and NSELFINGS options must be set to define the number of backcrosses to the first parent and the number of selfings, respectively. By default, the values of each trait are standardized by dividing them by their standard deviation, but you can set option STANDARDIZE=none to suppress this.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive effects can be included in the model by specifying dominance genetic predictors using the DOMINANCEPREDICTORS parameter (e.g. in a F2 population). In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified, the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS

parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci must be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked. The IDPARENTS parameter can supply labels to identify the parents.

The QTL model assumes FTRAITS and QTLs as fixed terms, and GENOTYPES as a random term. The QTLSELECTED parameter must specify the set of QTLs, in the form of a variate containing the index number of the positions where the QTLs are located. The INTERACTIONS parameter supplies a logical variate containing zero if a QTL effect is constrained to be constant across traits, and one if it is specific for each trait. When the DOMINANCEPREDICTORS parameter is set, the DOMSELECTED parameter supplies a logical variate containing one if the dominance predictor of the corresponding marker must be present in the model, and zero if the dominance predictor of the corresponding marker must be absent in the model. If DOMINANCEPREDICTORS is set but DOMSELECTED is not set, all the dominance predictors are included. Similarly, the DOMINTERACTIONS parameter supplies a logical variate containing one if the dominance-by-trait interaction of the corresponding marker must be present in the model, and zero if it must be absent. If DOMINANCEPREDICTORS is set but DOMINTERACTIONS is not set, all the dominance predictors are included.

Extra fixed effects can be defined by the FIXED option. A multi-Normal distribution, with vector mean 0 and variance covariance matrix Σ is assumed for the random genetic effects for the different traits. The VCMODEL option defines the model to use for Σ. The default assumes compound symmetry, but the VGESELECT procedure can be used to assess what model would be most suitable. Initial values for the parameters in the variance-covariance model can be specified by the VCINITIAL parameter. The VCPARAMETERS option controls whether the variance-covariance parameters are re-estimated at each step of the backward selection (VCPARAMETERS=estimate), or whether they are fixed at the defined initial values (VCPARAMETERS=fix). The VCSELECT option defines whether an extra check is made at each step on the variance-covariance model, to assess whether a simpler model is more suitable than the current model (based on the criterion defined by the CRITERION option). The SELECTEDMODEL parameter stores the final variance-covariance model that is selected.

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the other settings correspond to those in the PRINT option of the REML directive.

The QTL effects and their standard errors can be saved, in pointers, by the QEFFECTS and QSE parameters, respectively. These pointers have 2 levels of suffixes: the first level has 1, 2 or 3 values depending on the setting of the 3 possible predictors ADDITIVEPREDICTORS, ADD2PREDICTORS and DOMINANCEPREDICTORS; the second level has as many levels as the number of levels of the FTRAITS factor. The fitted values and residuals can be saved by the FITTEDVALUES and RESIDUALS parameters. The Wald statistics, degrees of freedom and probabilities can be saved by the parameters WALDSTATISTICS, DFWALD and PRWALD, respectively.

The OUTFILENAME parameter can be used to save the Wald statistics and the QEFFECTS and QSE structures in a Genstat work book file in a sheet named STATISTICS. This parameter should not contain an extension as the extension is defined automatically as .gwb.

The QSAVE parameter can be used to save a pointer containing information and results for the significant QTLs. The elements of the pointer are labelled as follows to simplify their subsequent use:

'procedure'	stores the string 'QMTESTIMATE' to indicate the source of the results,
'markernames'	marker names,
'chromosomes'	chromosomes,
'positions'	positions,
'traitnames'	names of the traits,
'waldstatistics'	wald statistics,
'prwald'	probability values of wald statistics,
'dfwald'	degrees of freedom of the wald statistics,
'qeffects'	QTL effects,
'qse'	standard errors of the QTL effects,
'%vexplained'	percentage variance explained,
'lowerci'	lower bound of confidence interval of estimated QTL position,
'upperci'	upper bound of confidence interval of estimated QTL position,
'posmin'	position of left flanking marker,
'posmax'	position of right flanking marker,
'idlfm'	marker name of left flanking marker,
'idrfm'	marker name of right flanking marker,
'posminci'	position of left flanking marker outside confidence interval,
'posmaxci'	position of right flanking marker outside confidence interval,
'idlfmci'	marker name of left flanking marker outside confidence interval,
'idrfmci'	marker name of right flanking marker outside confidence interval,
'locus'	index numbers of the significant QTLs, and
'neff'	number of additive and dominance predictors in the model.

The elements 'procedure','markernames','chromosomes','traitnames','idlfm', 'idrfm', 'idlfmci' and 'idrfmci' are text structures; 'positions', 'waldstatistics', 'prwald' and 'dfwald' are variates; 'qeffects' and 'qse' are pointers (see parameters QEFFECTS and QSE), as similarly are 'lowerci', 'upperci', 'posmin','posmax','posminci','posmaxci','idlfmci' and 'idrfmci'; 'neff' is a scalar.

The SAVE parameter can be used to save the REML save structure from the analysis for use with subsequent VKEEP and VDISPLAY directives.

Options: PRINT, POPULATIONTYPE, NGENERATIONS, NBACKCROSSES, NSELFINGS, VCMODEL, VCPARAMETERS, VCSELECT, STANDARDIZE, CRITERION, FIXED, UNITFACTOR,MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: Y, GENOTYPES, FTRAITS, UNITERROR, VCINITIAL, SELECTEDMODEL, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, IDMGENOTYPES, IDPARENTS, QTLSELECTED, INTERACTIONS, DOMSELECTED, DOMINTERACTIONS, RESIDUALS, FITTEDVALUES, WALDSTATISTICS, PRWALD, DFWALD, QEFFECTS, QSE, OUTFILENAME, QSAVE, SAVE.

Method

QMTESTIMATE fits the following models, which include a set L of QTLs:

1) $y_{ij} = \mu + T_j + \Sigma_{l \in L} x_{il}^{add} \alpha_{jl}^{add} + GT_{ij}$
   if only ADDITIVEPREDICTORS are specified

2) $y_{ij} = \mu + T_j + \Sigma_{l \in L} (x_{il}^{add} \alpha_{jl}^{add} + x_{il}^{dom} \alpha_{jl}^{dom}) + GT_{ij}$
   if DOMINANCEPREDICTORS are also specified

3) $y_{ij} = \mu + T_j + \Sigma_{l \in L} (x_{il}^{add} \alpha_{jl}^{add} + x_{il}^{add2} \alpha_{jl}^{add2} + x_{il}^{dom} \alpha_{jml}^{dom}) + GT_{ij}$
   if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_{ij} is the value of trait j for genotype i, T_j is the trait main effect, x_{il}^{add} are the additive genetic predictors of genotype i for locus l, and α_{jl}^{add} are the associated effects. In models 2 and 3, x_{il}^{dom} are the dominance genetic predictors, and α_{jl}^{dom} are the associated effects. In model 3, x_{il}^{add} are the additive genetic predictors for maternal genotype i at locus l, x_{il}^{add2} are the additive genetic predictors for paternal genotype i, and α_{jl}^{add} and α_{jl}^{add2} are the associated effects. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). GT_{ij} is assumed to follow a multi-Normal distribution with mean vector 0, and a variance covariance matrix Σ, that can either be modelled explicitly (with an unstructured model) or by some parsimonious model (defined by option VCMODEL) as described in the VGESELECT procedure.

Action with RESTRICT

Restrictions are not allowed.

Reference

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QMTBACKSELECT, QMTQTLSCAN, QMVAF, QFLAPJACK, QREPORT, VGESELECT.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMTQTLSCAN

Performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in multi-trait trials (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, progress, model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default summ
PLOT = *string token*	Whether to plot the profile along the genome (profile); default prof
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
ALPHALEVEL = *scalar*	Defines a genome-wide significance level to calculate the threshold; default 0.05
VCMODEL = *string token*	Specifies the variance-covariance model for the set of traits (identity, diagonal, cs, hcs, outside, fa, fa2, unstructured); default cs
VCPARAMETERS = *string token*	Whether to re-estimate the variance-covariance model parameters (estimate, fix); default esti
STANDARDIZE = *string token*	How to standardize the traits (none, normalize) ; default norm
COFACTORS = *variate*	Index numbers of loci to be used as cofactors for the genetic background
COFWINDOW = *scalar*	Specifies a window for cofactor exclusion from the model; default 10^6 which means that all cofactors on the same chromosomes are excluded
THRMETHOD = *string token*	Which method to use to calculate the threshold for QTL detection (bonferroni, liji, given); default liji
THRESHOLD = *scalar*	Threshold value for test statistic when THRMETHOD=given
DISTANCE = *scalar*	Distance between loci when THRMETHOD=bonferroni; default 4
FIXED = *formula*	Formula with extra fixed terms
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
STATISTICTYPE = *string token*	Which test statistic to plot and save using the STATISTICS parameter (wald, minlog10p); default minl
COLOURS = *scalar*, *variate* or *text*	Colours to use for the chromosomes; default * uses the colours of pens 1, 2 up to the number of chromosomes
TITLE = *text*	General title for the plot
YLOWERTITLE = *text*	Title for the y-axis of the lower graph(s); default 'Traits'
YUPPERTITLE = *text*	Title for the y-axis of the upper graph; default uses the identifier of the STATISTICS variate or pointer
XTITLE = *string*	Title for the x-axis; default 'Chromosomes'
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for

	use by the REML algorithm; default 100

Parameters

Y = *variates*	Quantitative traits to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
FTRAITS = *factors*	Factor indicating the trait of each y-value; must be set
UNITERROR = *variate*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted
VCINITIAL = *pointers*	Initial values for the parameters ofthe variance-covariance model
ADDITIVEPREDICTORS = *pointers*	Additive genetic predictors; must be set
ADD2PREDICTORS = *pointers*	Second (paternal) set of additive genetic predictors
DOMINANCEPREDICTORS = *pointers*	Dominance genetic predictors
CHROMOSOMES = *factors*	Chromosomes corresponding to the genetic predictors; must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
IDLOCI = *texts*	Labels for the loci
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the genetic predictors
IDEFFECTS = *texts*	Labels for the effects along the y-axis, in the frame below the profile plot
IDPARENTS = *texts*	Labels to use to identify the parents
QSTATISTICS = *variates*	Saves test statistics for QTL effects along the genome
QEFFECTS = *pointers*	Saves QTL effects along the genome
QSE = *pointers*	Saves standard errors of the QTL effects
OUTFILENAME = *texts*	Name of the Genstat workbook file (*.gwb) to be created
DFILENAME = *texts*	Name of the graphics file for the plots

Description

QMTQTLSCAN performs a genome-wide QTL scan in multi-trait trials as described by Malosetti *et al*. (2004) and Boer *et al*. (2007). It uses means per genotype-trait combinations as phenotypic data, but weights can be attached to the means (see the UNITERROR parameter and the UNITFACTOR option below). The response variable must be specified by the Y parameter, and the corresponding trait and genotype factors must be specified by the FTRAITS and GENOTYPES parameters, respectively. The POPULATIONTYPE option must be set to specify the population type. By default, the values of each trait are standardized by dividing them by their standard deviation, but you can set option STANDARDIZE=none to suppress this.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive effects can be included in the model by specifying dominance genetic predictors using the DOMINANCEPREDICTORS parameter (e.g. in a F2 population). In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified, the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the

phenotypic and in the marker data will be checked.

The QTL detection model assumes FTRAITS as a fixed term, and GENOTYPES as a random term. Extra fixed effects can be specified using the FIXED option. For the random genetic effects of the traits a multi-Normal distribution is assumed with mean vector 0 and variance-covariance matrix Σ. The VCMODEL option defines the model to use for Σ; the default is to take compound symmetry (the best model can be selected using the VGESELECT procedure). Initial values for the parameters in the variance-covariance model can be defined by the VCINITIAL parameter. The VCPARAMETERS option controls whether variance-covariance parameters are re-estimated at each iteration (VCPARAMETERS=estimate), or whether they are fixed at the initial values (VCPARAMETERS=fix). The fix setting can be useful to save computation time with large data sets or with more complex models.

The QTL search can be performed with cofactors to control for genetic background effects (*Composite Interval Mapping*) or without cofactors (*Simple Interval Mapping*). For Composite Interval Mapping, the COFACTORS option must be set to a variate containing the index numbers of the loci designated as cofactors. The COFWINDOW option defines a window around a tested position within which cofactors are temporarily excluded from the model.

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The method to define the threshold value is defined by the THRMETHOD option and uses a genome-wide error rate defined by the option ALPHALEVEL (default 0.05). If THRMETHOD=given, a user-defined threshold value must be specified using the THRESHOLD option. If THRMETHOD=bonferroni, an effective number of tests is calculated using the value specified by the DISTANCE option as the step size (default 4). Alternatively the liji setting uses the method described by Li & Ji (2005). See procedure QTHRESHOLD for details.

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the progress setting shows how the scan is progressing. The other settings correspond to those in the PRINT option of the REML directive.

By default QMTQTLSCAN produces a pair of graphs: the upper one plots the test statistic associated with the effects of the genetic predictors against their position on the chromosomes, and the lower one is a heat plot showing how the statistic changes over the traits. You can suppress the plotting by setting option PLOT=*. The STATISTICTYPE option specifies what to plot along the y-axis of the upper plot, either the test statistic or the associated probability value (on a −log10 scale), and also defines what is saved in the variates specified by the QSTATISTICS parameter. The IDEFFECTS parameter can be used to label the effects, and the IDPARENTS parameter can supply labels to identify the parents.

The effects of each genetic predictor and their standard errors can be saved, in pointers, by the QEFFECTS and QSE parameters, respectively. These pointers have 2 levels of suffixes: the first level has 1, 2 or 3 values depending on the setting of the 3 possible predictors ADDITIVEPREDICTORS, ADD2PREDICTORS and DOMINANCEPREDICTORS; the second level has as many levels as the number of levels of the TRAITS factor.

The TITLE, YLOWERTITLE, YUPPERTITLE and XTITLE options can specify the general title of the graph, the title of the y-axis on the lower graph(s), the title of the y-axis on the upper graph, and the title of the x-axis, respectively. The colours to use for the chromosomes in the upper graph are specified by the COLOURS option using either a text of colour names or a variate of RGB values (see the PEN directive for details). If COLOURS is not set, the default is to use the default colours of the pens 1, 2, onwards, up to the number of chromosomes. By default, the plot

is sent to the screen. However, you can supply a file for the plot, using the DFILENAME parameter. You can discover the types of graphics file that are supported by running the command.

```
DHELP possible
```

The OUTFILENAME parameter can be used to write the QSTATISTICS, QEFFECTS and QSE structures to a Genstat work book file in a sheet named STATISTICS. This parameter should not contain an extension as the extension is defined automatically given as .gwb.

Options: PRINT, PLOT, POPULATIONTYPE, ALPHALEVEL, VCMODEL, VCPARAMETERS, STANDARDIZE, COFACTORS, COFWINDOW, THRMETHOD, THRESHOLD, DISTANCE, FIXED, UNITFACTOR, STATISTICTYPE, COLOURS, TITLE, YLOWERTITLE, YUPPERTITLE, XTITLE, YLABEL, MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: Y, GENOTYPES, FTRAITS, UNITERROR, VCINITIAL, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, IDMGENOTYPES, IDEFFECTS, IDPARENTS, QSTATISTICS, QEFFECTS, QSE, OUTFILENAME, DFILENAME.

Method

QMTQTLSCAN fits the following mixed models repeatedly along the genome:

1) $y_{ij} = \mu + T_j + \Sigma_{f \in F} x_{il}^{add} c_{jf}^{add} + x_i^{add} \alpha_j^{add} + TE_{ij}$

if only ADDITIVEPREDICTORS are specified

2) $y_{ij} = \mu + T_j + \Sigma_{f \in F} (x_{if}^{add} c_{jf}^{add} + x_{if}^{dom} c_{jf}^{dom}) + (x_i^{add} \alpha_j^{add} + x_i^{dom} \alpha_j^{dom}) + TE_{ij}$

if DOMINANCEPREDICTORS are also specified

3) $y_{ij} = \mu + T_j + \Sigma_{f \in F} (x_{if}^{add} c_{jf}^{add} + x_{if}^{add2} c_{jf}^{add2} + x_{if}^{dom} c_{jf}^{dom})$
$+ (x_i^{add} \alpha_j^{add} + x_i^{add2} \alpha_j^{add2} + x_i^{dom} \alpha_j^{dom}) + TE_{ij}$

if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_{ij} is the value of trait j for genotype i, T_j is the trait main effect, F is a set of cofactors (if cofactors are included in the model), and x_{if}^{add} and x_i^{add} are the additive genetic predictors of genotype i at the cofactor positions and at the tested position, respectively. The associated effects are denoted by c_{jf}^{add} and α_j^{add} for cofactors and tested position respectively. In model 2 and 3, x_{if}^{dom} and x_i^{dom} are dominance genetic predictors of genotype i at the cofactor positions and at the tested position, respectively, with associated effects c_{jf}^{dom}, and α_j^{dom}. In model 3, x_{if}^{add} and x_i^{add} are the additive genetic predictors for the maternal genotype, for cofactors and tested position, respectively, and x_{if}^{add2} and x_i^{add2} are the equivalent additive genetic predictors for the paternal genotype. Finally x_{if}^{dom} and x_i^{dom} are the dominance genetic predictors for the cofactors and tested position, respectively. The associated effects are given by c_{jf}^{add}, c_{jf}^{add2} and c_{jf}^{dom} for cofactors, and α_j^{add}, α_j^{add2} and α_j^{dom} for tested positions. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). The residual unexplained genetic and trait effects are modelled by the GT_{ij} term, which is assumed to follow a multi-Normal distribution with mean vector 0, and a variance covariance matrix Σ. The matrix Σ can either be modelled explicitly (with an unstructured model) or by some parsimonious models (defined by option VCMODEL) as described in the VGESELECT procedure.

The procedure uses the REML directive iteratively to fit the model at each chromosome position, storing the Wald statistic for hypothesis testing. The resulting Wald statistic or the associated probability value (on a −log10 scale) can be plotted to produce the well-known profile plots along the chromosomes.

Action with RESTRICT
Restrictions are not allowed.

References
Boer, M.P., Wright, D., Feng, L., Podlich, D.W., Luo, L., Cooper, M. & van Eeuwijk, F.A. (2007). A mixed-model quantitative trait loci (QTL) analysis for multiple-environment trial data using environmental covariables for QTL-by-environment interactions, with an example in maize. *Genetics*, **177**, 1801-1813.

Malosetti, M., Voltas, J., Romagosa, I., Ullrich, S.E. & van Eeuwijk, F.A. (2004). Mixed models including environmental covariables for studying QTL by environment interaction. *Euphytica*, **137**, 139-145.

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also
Procedures: QMTBACKSELECT, QMTESTIMATE, QMVAF, VGESELECT.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMVAF

Calculates percentage variance accounted for by QTL effects in a multi-environment analysis (S.J. Welham, M.P. Boer, M.Malosetti & J.T.N.M. Thissen).

Options

PRINT = *string token*	What to print (summary); default summ
SELECTION = *string tokens*	What types of statistics to calculate (add, drop, cumulative); default add, drop, cumu
METHOD = *string tokens*	What methods to use to calculate the percentage variance accounted for (trace, determinant); default trac, dete
VCMODEL = *string token*	Specifies the variance-covariance model for the set of environments (identity, diagonal, cs, hcs, outside, fa, fa2, unstructured); default cs
FIXED = *formula*	Defines extra fixed effects
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 100

Parameters

TRAIT = *variates*	Quantitative trait to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
ENVIRONMENTS = *factors*	Environment factor; must be set
UNITERROR = *variate*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted
VCINITIAL = *pointers*	Initial values for the parameters of the variance-covariance model
ADDITIVEPREDICTORS = *pointers*	Additive genetic predictors; must be set
CHROMOSOMES = *factors*	Chromosomes corresponding to the genetic predictors; must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
IDLOCI = *texts*	Labels for the loci
QTLSELECTED = *variates*	Index numbers of the selected QTLs; must be set
INTERACTIONS = *variates*	Logical variate indicating whether each selected QTL has a significant (1) or non-significant (0) QTL-by-environment interaction
OUTFILENAME = *texts*	Name of the Genstat workbook file (*.gwb) to be created

Description

QMVAF calculates the percentage variance accounted for by estimated QTL effects in a multi-environment trial. The response variable must be specified by the TRAIT parameter, and the corresponding environment and genotype factors must be specified by the ENVIRONMENTS and GENOTYPES parameters, respectively. Molecular information used in the original analysis must

be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter.

The QTL model assumes ENVIRONMENTS and QTLs as fixed terms, and ENVIRONMENTS.GENOTYPES as random term. The QTLSELECTED parameter must specify the set of QTLs in the final model, in the form of a variate containing the index number of the positions where the QTLs are located. The INTERACTIONS parameter supplies a logical variate containing zero if a QTL effect is constrained to be constant across environments, and one if it is specific for each environment (QTL × environment interaction present). Extra fixed effects can be defined by the FIXED option. A multi-Normal distribution, with vector mean 0 and variance covariance matrix Σ is assumed for the random genetic effects in the different environments. The VCMODEL option defines the model to use for Σ, which should be the same as that used to identify the set of QTL effects. Initial values for the parameters in the variance-covariance model can be specified by the VCINITIAL parameter and the parameters will be re-estimated for each internal fit of the model.

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The PRINT option specifies the output to be displayed. The summary setting prints the information about the percentage variance accounted for by QTLs in the model.

The METHOD option specifies the method to use to calculate the percentage variance accounted for. This can be done by calculating the change in either the trace or determinant of the fitted covariance model. The trace and determinant correspond to the arithmentic and geometric means of the eigenvalues of the covariance matrix, respectively.

The SELECTION option specifies the statistics to be calculated, with the following settings:

add	the impact of adding a single QTL term is calculated by comparing the total variance (measured by trace or determinant) under the baseline model (which contains only the ENVIRONMENTS factor and any extra fixed terms specified using the FIXED option) with the total variance under a model containing a single QTL term, partitioned into main effect and (if present) interaction in addition to main effect; each of the QTL terms specified by the QTLSELECTED parameter is tested in turn;
drop	the comparison is between the full model (containing the ENVIRONMENTS factor, any extra FIXED terms, and all QTL terms specified by the QTLSELECTED parameter) and models excluding each one of the QTL terms, in turn; again this is partitioned into main effect and (if present) interaction; and
cumulative	a model is built up by adding in first all main effects and then all interaction terms, calculating the percentage variance accounted for at each step; the order in which the terms are added is determined by the percentage variance accounted for by individual terms.

The OUTFILENAME parameter can be used to save the summary statistics in a Genstat workbook. This workbook has one page for each type of statistic (determined by the settings of

SELECTION option) calculated using each method (determined by the settings of option METHOD). This parameter should not contain an extension as the extension is defined automatically as .gwb.

Options: PRINT, SELECTION, METHOD, VCMODEL, FIXED, UNITFACTOR,MVINCLUDE, MAXCYCLE, WORKSPACE.
Parameters: TRAIT, GENOTYPES, ENVIRONMENTS, UNITERROR, VCINITIAL, ADDITIVEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, QTLSELECTED, INTERACTIONS, OUTFILENAME.

Method
QMVAF works with the models fitted by QMESTIMATE, which include a set L of QTLs:

$$y_{ij} = \mu + E_j + \Sigma_{l \in L} x_{il} \alpha_{jl} + GE_{ij}$$

where y_{ij} is the trait value of genotype i in environment j, E_j is the environment main effect, x_{il} are the additive genetic predictors of genotype i for locus l, and α_{jl} are the associated effects. A variance matrix Σ (defined by option VCMODEL) is fitted within line across environments, with independence across lines. Additional fixed terms may be specified by using option FIXED.

QMVAF compares the fit of two models, that differ according to QTL effects that they contain, by looking at the change in the trace or the determinant of the across-environment variance matrix Σ. The trace considers the average change in within-environment variances, whilst the determinant also considers the impact on across-environment covariances. The variance matrix obtained by fitting the model without any QTL terms gives a measure of total variance, which is used as the denominator in all comparisons.

Action with RESTRICT
Restrictions are not allowed.

See also
Procedures: QMBACKSELECT, QMESTIMATE, QMQTLSCAN, VGESELECT.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMVESTIMATE

Replaces missing molecular marker scores using conditional genotypic probabilities (D.A. Murray, M. Malosetti & M.P. Boer).

Options

POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
NGENERATIONS = *scalar*	Number of generations of selfing for a RIL population
NBACKCROSSES = *scalar*	Number of backcrosses for a BCxSy population
NSELFINGS = *scalar*	Number of selfings for a BCxSy population

Parameters

MKSCORES = *pointers*	Genotype codes for each marker; must be set
CHROMOSOMES = *factors*	The chromosome where each marker is located; must be set
POSITIONS = *variates*	The position on the chromosome of each marker; must be set
MKNAMES = *texts*	Marker names; must be set
IDMGENOTYPES = *texts*	Labels for the genotypes
PARENTS = *pointers*	Parent information; must be set
IDPARENTS = *texts*	Labels used to identify the parents; must be set
NEWMKSCORES = *pointers*	Saves the imputed genotype codes for each marker; if this is not set, the imputed values overwrite those in MKSCORES

Description

QMVESTIMATE replaces missing marker scores using the conditional genotypic probabilities evaluated at specific chromosome positions. The marker scores containing the missing observations must be supplied in a pointer to a set of factors using the MKSCORES parameter. The linkage groups for each marker are supplied in a factor by the CHROMOSOMES parameter. The names of the markers are supplied in a text using the MKNAMES parameter, and the marker positions are supplied in a variate using the POSITIONS parameter. The IDMGENOTYPES parameter to label the genotypes should be supplied within a text. The parent information should be supplied in a pointer to a set of texts using the PARENTS parameter, and the labels for the parents should be supplied in a text using the IDPARENTS parameter. The marker scores containing the replaced missing observations can be saved within a pointer to a set of factors using the NEWMKSCORES parameter. If the NEWMKSCORES parameter is not set, then the missing marker scores are replaced in the MKSCORES.

The POPULATIONTYPE option must specify the population type. For recombinant inbred lines (POPULATIONTYPE = RIL), the NGENERATIONS option specifies the number of generations; default 3. For backcross inbred lines (POPULATIONTYPE = BCxSy), the NBACKCROSSES and NSELFINGS options must be set to define the number of backcrosses to the first parent and the number of selfings, respectively.

Options: POPULATIONTYPE, NGENERATIONS, NBACKCROSSES, NSELFINGS.
Parameters: MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, IDMGENOTYPES, PARENTS, IDPARENTS, NEWMKSCORES.

Method

QMVESTIMATE calls QIBDPROBABILITIES to calculate the conditional probabilities.

See also
Procedures: QIBDPROBABILITIES, QMVREPLACE.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QMVREPLACE

Replaces missing marker scores with the mode scores of the most similar genotypes (L.C.P. Keizer, J.T.N.M. Thissen & F.A. van Eeuwijk).

Options

PRINT = *string tokens*	What to print (summary, similarity, neighbours, details); default summ
NNEIGHBOURS = *scalar*	Number of nearest neighbours; default 5
MAXDISTANCE = *scalar*	Maximum similarity difference; default 0.1

Parameters

MKSCORES = *pointers*	Pointer with the original marker scores; must be set
MKNAMES = *texts*	Marker names
IDMGENOTYPES = *texts*	Labels for genotypes
NEWMKSCORES = *pointers*	Pointer to store the new marker scores; must be set

Description

QMVREPLACE replaces missing marker scores with the mode score of the most similar genotype(s). The marker scores with missing values are supplied by the MKSCORES pointer, which contains a factor for each marker. The length of factors is the number of genotypes. The new factors, in which the missing marker scores are replaced, can be saved by the NEWMKSCORES pointer. The MKNAMES and IDMGENOTYPES parameters can be set to obtain more readable output.

QMVREPLACE forms a similarity matrix from the marker scores using the FSIMILARITY directive with parameter TEST=simplematching. The NNEIGHBOURS option specifies the number of most-similar neighbouring genotypes to use when filling in the missing values for each genotype (default 5). To prevent the use of neighbours that are too different from the genotype, neighbours are selected only if their distances from the genotype are less than the value supplied by the MAXDISTANCE option (default 0.1). For each missing marker score of the genotype, a replacement value is obtained by taking the score that is most common amongst the selected neighbouring genotypes (i.e. the mode of their values). If the marker scores of the closest genotypes are all missing, the missing value is not replaced.

The PRINT option controls the printed output with settings:

summary	prints general information about the replaced marker missing scores,
similarity	prints the similarity matrix,
neigbours	prints the most-similar neighbours of the genotypes with missing marker scores, and
details	prints information about each replacement,

Options: PRINT, NNEIGHBOURS, MAXDISTANCE.
Parameters: MKSCORES, MKNAMES, IDMGENOTYPES, NEWMKSCORES.

Action with RESTRICT

Restrictions are not allowed.

See also

Procedures: MULTMISSING, QMVESTIMATE, SVHOTDECK.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QNORMALIZE

Performs quantile normalization (D.B. Baird).

Options

PRINT = *string token*	What to print (summary); default summ
PLOT = *string tokens*	What to plot (cdf, histogram, ncdf, nhistogram); default hist, nhis
METHOD = *string token*	Whether to use means, medians or geometric means for the averaged normalized distribution (means, medians, geometricmeans); default mean
ARRANGEMENT = *string token*	Whether to use trellis or single plots for PLOT=cdf or ncdf (single, trellis); default trel
DEVICE = *scalar*	Device number on which to plot the graphs
GRAPHICSFILE = *text*	What graphics filename template to use to save the graphs; default *

Parameters

DATA = *variates* or *pointers*	Data values
GROUPS = *factors* or *texts*	Groupings of the data values, or descriptions of the variates in the pointer
NEWDATA = *variates or pointers*	Saves the normalized values; if this is unset, they replace the original values in DATA

Description

QNORMALIZE performs quantile normalization. This transforms the data so that each group has a common cumulative density function. The data values are specified by the DATA parameter. They can be in a single variate, with groupings specified by the GROUPS parameter. Alternatively, they can be in a pointer to separate variates, one for each group. The GROUPS parameter can be set to a text to describe the variates. The normalized values can be saved using the NEWDATA parameter. If this is not set, they replace the values in the DATA variate(s).

The METHOD option selects the way in which the overall distribution is produced from the cumulative density functions within each group, with settings:

means	takes the means;
medians	takes the medians; and
geometricmeans	takes geometric means (i.e. the mean on the log scale, back-transformed to the natural scale).

The PLOT option controls what plots are produced: histograms or cumulative density plots of the original or normalized data. By default the plots for the groups are displayed in a trellis arrangement, but you can set option ARRANGEMENT=single to display them separately, in single plots. You can use the DEVICE option to plot to a device other than the screen. The GRAPHICSFILE option specifies then supplies a template for the file names.

By default a summary is produced, giving quantiles by groups. This can be suppressed by putting option PRINT=*.

Options: PRINT, METHOD, ARRANGEMENT, DEVICE, GRAPHICSFILE.
Parameters: DATA, SLIDES, NEWDATA.

Action with RESTRICT

Any restrictions on the DATA variates are removed.

See also

Procedure: MABGCORRECT.

Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Microarray data.

QRECOMBINATIONS

Calculates the expected numbers of recombinations and the recombination frequencies between markers (J. Jansen, J.T.N.M. Thissen & M.P. Boer).

Options

PRINT = *string tokens*	What to print (summary, positions); default summ
PLOT = *string token*	What to plot (frequencies); default freq
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, CP); must be set
METHOD = *string token*	Which method to use (twopoint, multipoint); default twop
USEPENALTY = *string token*	Whether to increase the number of recombinations when METHOD=twopoint by 0.5 recombination per informative meiosis for each missing marker score (yes, no); default no
TITLE = *text*	General title for the plot

Parameters

MKSCORES = *pointers*	Marker scores for each marker; must be set
CHROMOSOMES = *factors*	Factor defining the linkage groups
POSITIONS = *variates*	Saves the positions of the markers when METHOD=multipoint
MKNAMES = *texts*	Names of the markers; must be set
PARENTS = *pointers*	Marker scores of the parents; must be set
ORDER = *variates*	Order of the markers for METHOD=multipoint
NRECOMBINATIONS = *symmetric matrices* or *pointers*	Saves the number of recombinations
RECFREQUENCIES = *symmetric matrices* or *pointers*	Saves the recombination frequencies
PHASESWITCHES = *pointers*	Saves the phase switches for pairs of markers when POPULATIONTYPE=CP
INHERITANCEVECTORS = *pointers*	Saves the inheritance vectors when METHOD=multipoint
GENNRECOMBINATIONS = *variates*	Saves the numbers of recombinations of the genotypes when METHOD=multipoint

Description

QRECOMBINATIONS calculates the expected numbers of recombinations, and the recombination frequencies between markers, from the marker scores. The marker scores of the genotypes are supplied in a pointer by the MKSCORES parameter. This contains a set of factors (with levels all in the same order), each one with the data for one of the markers. The names of the markers must be supplied, in a text, using the MKNAMES parameter. The marker scores of the parents must be supplied using the PARENTS parameter. The CHROMOSOMES parameter can be set if the markers do not belong to the same linkage group. The POPULATIONTYPE option must be set to specify the type of population from which the marker scores have been obtained.

The METHOD option specifies whether the numbers of recombinations are calculated by the two-point or multi-point method. The default, METHOD=twopoint, must be used if the order of the markers is not available. The USEPENALTY option then controls whether the number of recombinations is increased by 0.5 recombination per informative meiosis for each missing marker score. For METHOD=multipoint the order of the markers must be supplied, using the ORDER parameter.

The numbers of recombinations and the recombination frequencies can be saved using the NRECOMBINATIONS and RECFREQUENCIES parameters, respectively. These usually save a

symmetric matrix. However, when POPULATIONTYPE=CP, the numbers of recombinations and recombination frequencies of the maternal and paternal meiosis are estimated separately, and so they each save a pointer containing two symmetric matrices. The PHASESWITCHES parameter saves the phase switches in the maternal and paternal meiosis for pairs of markers, in pointers of symmetric matrices. The value of the phase switch is set to one if the saved recombination frequency is equal to one minus the observed recombination frequency, and zero otherwise.

When METHOD=multipoint the positions of the markers are calculated, and can be saved in a variate using the POSITIONS parameter. The inheritance vectors and expected numbers of recombinations of the genotypes can then also be saved, using the INHERITANCEVECTORS and GENNRECOMBINATIONS parameters, respectively.

The PRINT option controls the printed output. The default setting, summary, prints the minimum, mean and maximum of the NRECOMBINATIONS values. When METHOD=multipoint, the positions setting can be used to print the minimum, mean and maximum of the POSITIONS values.

The default setting, frequencies, of the PLOT option plots the frequencies in a shaded diagram. The TITLE option can be used to provide a title for the plot.

Options: PRINT, PLOT, POPULATIONTYPE, METHOD, USEPENALTY, TITLE.
Parameters: MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, PARENTS, ORDER, NRECOMBINATIONS, RECFREQUENCIES, PHASESWITCHES, INHERITANCEVECTORS, GENNRECOMBINATIONS.

Method

For the two-point method, QRECOMBINATIONS estimates the expected numbers of recombinations and maximum likelihood estimates of the recombination frequencies of pairs of markers by the EM algorithm, using the formula

$$r_{New} = E(R|\text{marker data}, r_{Current}),$$

where R denotes the number of recombinations, and $r_{Current}$ and r_{New} denote the current and new values of the recombination frequency. The estimation requires iteration only when POPULATIONTYPE=F2 or in some cases when POPULATIONTYPE=CP; see Maliepaard, Jansen & van Ooijen (1997). In all other cases estimation only requires simple counting of recombinations.

For the multi-point method, QRECOMBINATIONS follows essentially the same procedure for estimating the recombination frequencies between adjacent markers in a sequence of markers, using hidden Markov models (HMM); see Lander & Green (1987).

Action with RESTRICT

Restrictions are not allowed.

References

Maliepaard, C., Jansen J. & van Ooijen J.W. (1997). Linkage analysis in a full-sib family of an outbreeding plant species: overview and consequences for applications. *Genet. Res., Camb,* **70**, 237-250.

Lander, E.S. & Green P. (1987). Construction of multilocus genetic linkage maps in humans. *Proc. Natl. Acad. Sci USA*, **84**, 2363-2367.

See also

Procedures: QLINKAGEGROUPS, QMAP.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QREPORT

Creates an HTML report from QTL linkage or association analysis results (D.A. Murray).

Options

OUTFILEPREFIX = *text*	Prefix to use for the files that are generated
WORKDIRECTORY = *text*	Working directory to use for files; default current Genstat working directory
CHROMOSOMES = *factor*	Factor defining linkage groups for the genetic map
POSITIONS = *variate*	Positions of markers within the linkage groups for the genetic map
HTMLHEAD = *text*	Text structure containing custom content for the header of the HTML report file

Parameter

QSAVE = *pointers*	Information and results saved from an earlier QTL analysis

Description

QREPORT creates an HTML report containing results from QTL linkage or association analyses. The QSAVE parameter of the QSESTIMATE, QMESTIMATE, QSASSOCIATION and QMASSOCIATION procedures can be used to save a pointer containing information and results for the significant QTLs. You can then produce an HTML report of the results, by using this pointer as the setting of the QSAVE parameter of QREPORT. If you specify a list of pointers, these will all be collated into a single report.

The OUTFILEPREFIX option specifies the name to use as the prefix for the file and associated graphs. The HTML report file is created using the prefix string with the extension .htm appended to the name. All graphs produced in the report are produced in PNG format using the prefix string in their file name. The files are saved within the working directory, which by default will be the current directory. However, an alternative directory can be supplied using the WORKDIRECTORY option.

A genetic map indicating the location of the significant QTLs can be included within the report, by supplying the linkage groupings and positions within the linkage groups, using the CHROMOSOMES and POSITIONS option, respectively.

The HTMLHEAD option allows you to supply additional markup content for the document header of the HTML file, which will be inserted between the <head> and </head> tags. It can be set either to a text containing all the HTML markup, or to the name of a file containing that information. It is intended primarily for inserting CSS style information; for more details see the OPEN directive. If HTMLHEAD is not set, QREPORT inserts the content of the file Genstat.css, which is supplied in the Source directory of the Genstat installation. This defines a number of classes that are used at various points in the Genstat output (for example to define styles used for output from the CAPTION directive). The file can be used as a template from which to derive a local variation, redefining basic elements of output.

Options: OUTFILEPREFIX, WORKDIRECTORY, CHROMOSOMES, POSITIONS, HTMLHEAD.
Parameter: QSAVE.

Method

The HTML report is produced by writing to an output file, opened by the OPEN directive, with parameter STYLE=html. The graphs are inserted in the report using the PLINK procedure.

Action with RESTRICT
Any data restrictions are ignored.

See also
Directive: OPEN.
Procedures: QFLAPJACK, QMASSOCIATION, QMESTIMATE, QSASSOCIATION, QSESTIMATE.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QSASSOCIATION

Performs marker-trait association analysis in a genetically diverse population using bi-allelic and multi-allelic markers (M. Malosetti & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, progress); default summ
PLOT = *string tokens*	What to plot (profile, qq, map); default prof, qq
RELATIONSHIPMODEL = *string token*	What model to use to account for genetic relatedness (eigenanalysis, kinship, subpopulations, null); default kins
SCORES = *pointer*	Provides the scores of significant principal components, obtained from an eigenvalue analysis
METHOD = *string token*	What model to use for GWAS (exact, fast); default fast
ALPHA = *scalar*	Defines a genome-wide significance level to calculate the threshold; default 0.05
THRMETHOD = *string token*	Method to define the threshold for significance (neffective, bonferroni, given); default neff
THRESHOLD = *scalar*	Threshold value for significant LD, on the −log10 scale; default 2
DISTANCE = *scalar*	Minimum distance gap between independent tests (i.e. distance beyond which loci are expected to be in linkeage equilibrium) when THRMETHOD=bonferroni; default *
MINORALLELE = *scalar*	Frequency of minor alleles; default 0.05
KMATRIX = *symmetric matrix*	Kinship matrix containing coefficients of coancestries
KMETHOD = *string token*	Method to use to estimate kinship matrix if not supplied by KMATRIX (correlation, dice); default dice
SUBPOPULATIONS = *factor*	Defines groupings of genotypes into subpopulations
MODELPART = *string token*	Defines which part of the model should include SUBPOPULATIONS if RELATIONSHIPMODEL is set to subpopulations, or the principal components scores if RELATIONSHIPMODEL is set to eigenanalysis (fixed, random); default rand
SCALING = *string token*	Whether to scale the scores by the square roots of their singular values (singularvalues, none); default none
STANDARDIZE = *string token*	Whether to standardize the marker scores according to their frequencies (frequency, none); default freq
COLOURS = *scalar*, *variate* or *text*	Colours to use for the chromosomes; default * uses the colours of pens 1, 2 up to the number of chromosomes
TITLE = *text*	General title for the plots
YTITLE = *text*	Title for the y-axis
XTITLE = *text*	Title for the x-axis

Parameters

TRAIT = *variates*	Phenotypic trait to analyse; must be set
GENOTYPES = *factors*	Genotype factor
MKSCORES = *pointers*	Genotype codes for each marker; must be set
CHROMOSOMES= *factors*	Linkage groups for the markers; must be set

POSITIONS = *variates*	Positions within the linkage groups of markers; must be set
MKNAMES = *texts*	Marker names
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the markers
GENFILENAME = *texts*	Name of a comma-delimited file (*.csv) containing marker scores (with markers in the rows and genotypes in the columns)
MAPFILENAME = *texts*	Name of a comma-delimited file (*.csv) with map information
WALDSTATISTICS = *variates*	Saves the Wald test statistics
NDF = *variates*	Saves the degrees of freedom associated with the Wald test
MINLOG10P = *variates*	Saves the associated probability values of the Wald test statistics, on a −log10 scale
LAMBDA = *scalars*	Saves the inflation factor i.e. slope of the QQ plot of −log10(P) values
QSAVE = *pointers*	Saves a pointer with information and results for the significant effects
DFILENAME = *texts*	Name of the graphics file for the plots

Description

QSASSOCIATION performs a mixed model marker-trait association analysis (also known as linkage disequilibrium mapping) with data from a single-environment trial. The trait data are supplied by the TRAIT parameter. The marker scores can be supplied as a pointer of factors by the MKSCORES parameter. The length of the pointer must be equal to the number of markers. Alternatively, if the fast method is requested by the METHOD option, they can be supplied in a file whose name is specified by the GENFILENAME parameter. The file must be comma-delimited (*.csv), with the markers in the rows and the genotypes in the columns. The first column of the file contains marker names, and the first row of the file contains the names of the genotypes.

The corresponding map information for the markers can be supplied by the CHROMOSOMES and POSITIONS parameters, and the labels for the markers can be supplied by the MKNAMES parameter. The IDMGENOTYPE parameter can be used to give the genotypes labels in the marker data. Alternatively, if the fast method is requested by the METHOD option, the map information can be supplied in a file, whose name is specified by the MAPFILE parameter. This file must also be comma-delimited (*.csv), and should contain three columns (without headings): marker name, linkage group (chromosome), and position within linkage group of each marker.

To avoid false positives in association mapping studies, some form of control is necessary for the genetic relatedness. The model to use is specified by the RELATIONSHIPMODEL option, with one of the following settings:

eigenanalysis	infers the underlying genetic substructure in the population by retaining the most significant principal components from the molecular marker matrix (Patterson *et al*. 2006) – the scores of the significant axes are used as covariables in the mixed model, which effectively is an approximation to the structuring of the genetic variance covariance matrix by a coefficient of coancestry matrix (kinship matrix);
kinship	is the default model, and includes a kinship matrix in the mixed model;
subpopulations	includes a factor supplied by the SUBPOPULATIONS option in the mixed model; and
null	makes no correction for genetic relatedness.

When RELATIONSHIPMODEL=kinship, the kinship matrix can be specified by the KMATRIX option. Alternatively, it can be calculated from the MKSCORES using the QKINSHIPMATRIX procedure with the method specified by the KMETHOD option (and can then be stored by KMATRIX).

When RELATIONSHIPMODEL=eigenanalysis, the scores of the significant axes can be supplied using the SCORES option. Otherwise they are calculated by the QEIGENANALYSIS procedure (and can then be stored by SCORES). The STANDARDIZE and SCALING options control whether the MKSCORES factors are standardized and scaled; see QEIGENANALYSIS for more details.

The MODELPART option controls whether the principal components scores (if RELATIONSHIPMODEL=eigenanalysis) or the subpopulations factor (if RELATIONSHIPMODEL=subpopulations) are included as random or fixed terms (default random).

The threshold for significant marker trait association (on a −log10 scale) is defined by the THRESHOLD option. The default value is 2.

The MINORALLELE option defines the frequency q below which alleles are considered rare. Rare alleles are automatically pooled together. Markers whose major frequency allele is greater than or equal to $1-q$ are considered close to fixation and are not used in the analysis.

The METHOD option defines the method to use to fit marker-trait association models, either exact or fast. For the exact method, the mixed models are solved for each marker separately. For the fast method, the mixed model is only solved for the genetic background model, without the markers in the model. The estimated variance-covariance matrix from this genetic background model is used to perform a generalized least squares scan for all the marker. The fast method is implemented only for bi-allelic markers, such as SNPs.

The THRMETHOD option controls how the threshold for significance is defined. The default THRMETHOD=neffective, first determines the effective number of columns (nC) in the marker matrix data using the estimator given by Patterson *et al*. (2006), and calculates the threshold as $-\log10(\alpha/nC)$. The parameter α is the genome-wide type I error rate, which is defined by the ALPHA option (default 0.05). Alternatively, THRMETHOD=bonferroni calculates the effective number of tests assuming one independent test within blocks of a size specified by the DISTANCE option. If DISTANCE is not set, the default is to take an independent test at every marker, which is very conservative in most cases. Finally, if THRMETHOD=given, a user-defined threshold value (on a log10 scale) must be specified using the THRESHOLD option. With the other setting of THRMETHOD, THRESHOLD can be used to save the estimated threshold.

The MINORALLELE option defines the frequency q below which alleles are considered rare. Rare alleles are automatically pooled together. Markers whose major frequency allele are greater than or equal to $1-q$ are considered close to fixation, and are not used in the analysis.

The PRINT option controls printed output, with settings:

summary	to print the list of markers with a significant association with the trait, and
progress	to monitor the progress of the analysis.

The default is PRINT=summary.

The PLOT option controls what graphs are produced, with settings:

profile	plots a genome wide profile of the -log10(P) of the test statistic,
map	plots a map with the location of the detected significant markers, highlighting whether or not the marker showed significant interaction with the environment, and
qq	makes a QQ plot of the −log10(P) values.

By default PLOT=profile,qq. The TITLE option can be used to provide a title for the graph, and the YTITLE and XTITLE options can supply titles for the y- and x-axis, respectively. The

colours to use for the chromosomes in the upper graph are specified by the COLOURS option using either a text of colour names or a variate of RGB values (see the PEN directive for details). If COLOURS is not set, the default is to use the default colours of the pens 1, 2, onwards, up to the number of chromosomes. By default, the plot is sent to the screen. However, you can supply a file for the plot, using the DFILENAME parameter. You can discover the types of graphics file that are supported by running the command.

```
DHELP possible
```

The Wald test statistics, their numbers of degrees of freedom and the associated probability values on a −log10 scale can be saved by the WALDSTATISTICS, NDF and MINLOG10P parameters, respectively. The LAMBDA parameter can save inflation factor, estimated as the slope of the QQ plot of the –log10(P) values. The QSAVE parameter can save a pointer containing information and results for the significant markers. The elements of the pointer are labelled as follows to simplify their subsequent use:

'procedure'	stores the string 'QSASSOCIATION' to indicate the source of the results,
'index'	index numbers of the significant markers,
'mkname'	marker names,
'chromosomes'	chromosomes,
'positions'	positions,
'minlog10p'	probability values on a −log10 scale,
'allele'	label of the relevant allele,
'frequency'	allele frequencies,
'effects'	effects and
'seeffects'	standard errors of the effects.

These are all pointers, with an element for each chromosome. The elements of the chromosome pointers are variates for all components except the standard errors of differences, which are scalars.

Options: PRINT, PLOT, RELATIONSHIPMODEL, SCORES, METHOD, ALPHA, THRMETHOD, THRESHOLD, DISTANCE, MINORALLELE, KMATRIX, KMETHOD, SUBPOPULATIONS, MODELPART, SCALING, STANDARDIZE, COLOURS, TITLE, YTITLE, XTITLE.

Parameters: TRAIT, GENOTYPES, MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, IDMGENOTYPES, GENFILENAME, MAPFILENAME, WALDSTATISTICS, NDF, MINLOG10P, LAMBDA, QSAVE, DFILENAME.

Method

QSASSOCIATION performs a mixed model marker-trait association analysis, or LD mapping. It takes account of the heterogeneous genetic relatedness between individuals in the population (sometimes referred as "population structure") using one of three possible models, specified by the RELATIONSHIPMODEL option, as defined below. The model for marker trait association may included the following terms: an intercept μ, the effects associated with k principal components $PCscore_{ki}$ (fixed or random), the effects of genotype groups $Group_k$ (fixed or random) and the effects of the tested markers MK (fixed).

The RELATIONSHIPMODEL option specifies which of the three possible models to use for the relatedness, and the MODELPART option controls whether these terms are treated as fixed or random.

Model	Fixed	Fixed or random	Fixed	Random
Eigenanalysis	$\mu +$	$\Sigma_i\, PCscore_{ki} +$	$MK +$	G_i
Kinship	$\mu +$		$MK +$	G_i with $G \sim N(0,2K\sigma_G)$
Subpopulations	$\mu +$	$Group_k +$	$MK +$	G_i
Null	$\mu +$		$MK +$	G_i

A Wald test is then used for each marker, individually, to test the null hypothesis that its effect is zero. The most frequent allele is set as the reference level. Marker allele frequencies, effects and standard errors are stored.

Action with `RESTRICT`
Restrictions are not allowed.

Reference
Patterson, N., Price, A.L., Reich, D. (2006). Population structure and eigenanalysis. *PLoS Genetics*, **2**, e190. doi:10.1371/journal.pgen.0020190

See also
Procedures: `QEIGENANALYSIS`, `QKINSHIPMATRIX`, `QLDDECAY`, `QMASSOCIATION`, `QREPORT`.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QSBACKSELECT

Performs a QTL backward selection for loci in single-environment trials (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default summ
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
ALPHALEVEL = *scalar*	Defines a significance level; default 0.05
FIXED = *formula*	Formula with extra fixed effects
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 100

Parameters

TRAIT = *variates*	Quantitative trait to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
UNITERROR = *variates*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted
ADDITIVEPREDICTORS = *pointers*	Additive genetic predictors; must be set
ADD2PREDICTORS = *pointers*	Second (paternal) set of additive genetic predictors
DOMINANCEPREDICTORS = *pointers*	Dominance genetic predictors
CHROMOSOMES = *factors*	Chromosomes corresponding to the genetic predictors; must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
IDLOCI = texts	Labels for the loci
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the genetic predictors
QTLCANDIDATES = *variates*	Specifies the locus index numbers from which to start the selection; must be set
QTLSELECTED = *variates*	Saves the index numbers of the selected QTLs; must be set
DOMSELECTED = *variates*	Logical indicator variable storing one where the selected QTLs show a significant effect of the dominance predictor, zero otherwise
WALDSTATISTICS = *variates*	Saves the Wald test statistics
PRWALD = *variates*	Saves the associated Wald probabilities

Description

QSBACKSELECT selects QTLs from a list of candidate QTLs (loci) in single-environment trials by backward selection. It uses single observation per genotype as phenotypic data. The response variable must be specified by the TRAIT parameter, and the genotypes by the GENOTYPES parameter. The POPULATIONTYPE option must be set to specify the population from which the genotypes are derived.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive effects can be included in the model by using the DOMINANCEPREDICTORS parameter to specify dominance genetic predictors (e.g. in a F2 population); again they are stored in variates and supplied in a pointer. In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified: the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked.

The set of candidate QTLs must be supplied by the QTLCANDIDATES parameter. The model assumes genotypes as random and QTLs as fixed effects. Extra fixed effects can be defined using the FIXED option. The significance level to use at each step of the backward selection process is given by the ALPHALEVEL option (default 0.05).

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the other settings correspond to those in the PRINT option of the REML directive.

The list of selected QTLs can be saved by the QTLSELECTED parameter. If the dominance predictors have been specified, the DOMSELECTED parameter can save a logical indicator variate storing one where the selected QTLs show a significant effect of the dominance predictor, and zero otherwise. The Wald test and associated probability values for the selected QTLs can be saved by the WALDSTATISTICS and PRWALD parameters, respectively.

Options: PRINT, POPULATIONTYPE, ALPHALEVEL, FIXED, UNITFACTOR, MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: TRAIT, GENOTYPES, UNITERROR, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, IDMGENOTYPES, QTLCANDIDATES, QTLSELECTED, DOMSELECTED, WALDSTATISTICS, PRWALD.

Method

QSBACKSELECT starts with one of the following models which includes a set L of candidate QTLs:

1) $y_i = \mu + \Sigma_{l \in L}\, x_{il}^{add}\, \alpha_l^{add} + G_i$

if only ADDITIVEPREDICTORS are specified

2) $y_i = \mu + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_l^{add} + x_{il}^{dom}\, \alpha_l^{dom}) + G_i$

if DOMINANCEPREDICTORS are also specified

3) $y_i = \mu + \Sigma_{l \in L} (x_{il}^{add} \alpha_l^{add} + x_{il}^{add2} \alpha_l^{add2} + x_{il}^{dom} \alpha_l^{dom}) + G_i$

if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_i is the trait value of genotype i, x_{il}^{add} are the additive genetic predictors of genotype i for locus l, and α^{add} are the associated effects. In models 2 and 3, x_{il}^{dom} are the dominance genetic predictors, and α_l^{add} are the associated effects. In model 3, x_{il}^{add} are the additive genetic predictors for maternal genotype i at locus l, x_{il}^{add2} are the additive genetic predictors for paternal genotype i, and α^{add} and α^{add2} are the associated effects. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). G_i is the residual unexplained genetic and environmental variation, which is assumed to follow a Normal distribution with mean 0 and variance σ^2.

The backward selection process starts with the initial set of loci L (defined by the QTLCANDIDATES parameter), and checks whether all the loci are significant. If not, the locus with the smallest Wald test statistic is dropped from the model. The process is repeated until all loci in the model are significant. If model 2 or 3 is specified, a further step of model reduction is performed by checking, for each of the remaining loci, whether the dominance effects can be dropped from the model.

Action with RESTRICT

Restrictions are not allowed.

Reference

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QSESTIMATE, QSQTLSCAN.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QSELECTIONINDEX

Calculates (molecular) selection indexes by using phenotypic information and/or molecular scores of multiple traits (M. Malosetti & F.A. van Eeuwijk).

Options

PRINT = *string tokens*	What to print (summary); default summ
METHOD = *string token*	Defines which index to calculate (simple, smithhazel, landethompson); default smit
INTENSITY = *scalar*	Specifies the selection intensity expressed as the percentage of individuals of the population to select; default 10

Parameters

TRAITS = *pointers*	Pointer with a variate for each trait, supplying the phenotypic values for the genotypes; must be set
MOLECULARSCORES = *pointers*	Pointer with a variate for each trait, supplying QTL-based predictions or genomic predictions
GENOTYPES = *factors*	Genotype factor; must be set
IDMGENOTYPES = *texts*	Labels of the genotypes
WEIGHTS = *variates*	Specifies economic weights for the traits; if unset, all traits have weight one
VCPHENOTYPIC = *symmetric matrices*	Specifies the phenotypic variance-covariance matrix of the traits
VCGENETIC = *symmetric matrices*	Specifies the genotypic variance-covariance matrix of the traits
HERITABILITY = *symmetric matrices*	Specifies the heritabilities and coheritabilities of the traits
SELECTIONINDEX = *variates*	Saves the selection index

Description

Selection indexes are a classical tool used in plant and animal breeding to select multiple traits simultaneously, conditional on given economic weights and specific selection targets. QSELECTIONINDEX allows several types of selection index to be calculated, that combine economic weights with additional selection constraints. It can produce standard selection indexes based on phenotypic information, or molecular selection indexes by incorporating molecular scores of genotypes derived from QTL or genomic prediction models.

The METHOD option defines which selection index to obtain, either a simple index, the Smith-Hazel index (default), or the Lande and Thompson index. See the Method Section below for details.

The INTENSITY option specifies the desired selection intensity, which is used to calculate the selection differential and the expected response to selection. It also determines the percentage of top performing genotypes to be to printed.

The TRAITS parameter must supply the phenotypes (observations) of the individuals whose selection indexes are to be calculated. Alternatively, you can use it to specify molecular scores (predictions from a QTL or genomic prediction model) if you want to construct an index based only on these. However, the Lande and Thompson index needs both phenotypes and molecular scores, and then the MOLECULARSCORES parameter must be used to supply the molecular scores of the individuals, while the TRAITS parameter provides the phentoypes.

The GENOTYPES parameter must specify a factor to identify the individuals, and the

IDMGENOTYPES parameter can supply a text to label the genotypes.

The WEIGHTS parameter specifies the economic weights to use for each of the traits entering the index. These must be given in the same order as in the pointer supplied by the TRAITS and MOLECULARSCORES parameters. The default is to use a weight of one for every trait.

The VCPHENOTYPIC and VCGENETIC parameters can be used to provide the phenotypic and genetic variance-covariance matrices between the traits. The rows of the matrices correspond to the traits, and must follow the same order as in the TRAITS and MOLECULARSCORES pointers. If VCPHENOTYPIC and VCGENETIC are not specified, the HERITABILITY parameter must be specified instead, to define the heritabilities and coheritabilities of the traits, is a symmetric matrix with the rows must be in the same order as in the TRAITS and MOLECULARSCORES pointers.

The SELECTIONINDEX parameter can be used to save the values of the selection index, in a variate.

By default, QSELECTIONINDEX prints a summary of the analysis, but you can set option PRINT=* to suppress this.

Options: PRINT, METHOD, INTENSITY.
Parameters: TRAITS, GENOTYPES, IDMGENOTYPES, MOLECULARSCORES, WEIGHTS, VCPHENOTYPIC, VCGENETIC, HERITABILITY, SELECTIONINDEX.

Method

The *simple* selection index uses either phenotypic information or molecular scores, and is defined as

$$SI = Y\,d$$

where Y is the $n \times t$ matrix containing the phenotypic data or the molecular scores for the n genotypes and t traits (with no missing values), and where d is the $t \times 1$ vector of trait-specific economic weights.

The *Smith-Hazel* index also uses either phenotypic information or molecular scores. Its definition is

$$SH = Y\,P^{-1}\,G\,d$$

where P and G are the $t \times t$ phenotypic and genotypic variance-covariance matrices.

The *Lande and Thompson* index uses both phenotypic and molecular scores, and is defined as

$$LT = Y^*\,P^{*-1}\,G^*\,d^*$$

where the matrix Y^* combines the matrix of phenotypic trait data Y and the matrix of predictions from the QTL or genomic prediction model (molecular scores) Y_m, appended one below the other i.e.

$$Y = \begin{pmatrix} Y \\ Y_m \end{pmatrix}$$

The corresponding variance-covariance matrices are

$$P^* = \begin{pmatrix} P & P_m \\ P_m & P_m \end{pmatrix}$$

and

$$G^* = \begin{pmatrix} G & G_m \\ G_m & G_m \end{pmatrix}$$

where P_m and G_m are the variance-covariance matrices for the molecular scores. The economic weights are

$$d^* = \begin{pmatrix} d \\ 0_m \end{pmatrix}$$

where 0_m is a vector of zero weights for the molecular scores.

Action with `RESTRICT`
Restrictions are not allowed.

See also
Procedure: `QBESTGENOTYPES`.
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QSESTIMATE

Calculates QTL effects in single-environment trials (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default summ
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
NGENERATIONS = *scalar*	Number of generations of selfing for a RIL population
NBACKCROSSES = *scalar*	Number of backcrosses for a BCxSy population
NSELFINGS = *scalar*	Number of selfings for a BCxSy population
FIXED = *formula*	Defines extra fixed effects
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 100

Parameters

TRAIT = *variates*	Quantitative trait to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
UNITERROR = *variates*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted
ADDITIVEPREDICTORS = *pointers*	Additive genetic predictors; must be set
ADD2PREDICTORS = *pointers*	Second (paternal) set of additive genetic predictors
DOMINANCEPREDICTORS = *pointers*	Dominance genetic predictors
CHROMOSOMES = *factors*	Chromosomes corresponding to the additive genetic predictors; must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the additive genetic predictors; must be set
IDLOCI = *texts*	Labels for the loci
MKLOCI = *variates*	Logical variate containing the value 1 if the locus is a marker, otherwise 0; must be set
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the the additive genetic predictors
IDPARENTS = *texts*	Labels to identify the parents
QTLSELECTED = *variates*	Index numbers of the selected QTLs; must be set
DOMSELECTED = *variates*	Logical variate indicating whether the dominance predictor of each selected QTL must be present (1) or absent (0) in the model
RESIDUALS = *variates*	Residuals from the analysis
FITTEDVALUES = *variates*	Fitted values from the analysis
WALDSTATISTICS = *variates*	Saves the Wald test statistics

PRWALD = *variates*	Saves the associated Wald probabilities
QEFFECTS = *pointers*	Saves the estimated QTL effects
QSE = *pointers*	Saves the standard errors of the QTL effects
OUTFILENAME = *texts*	Name of the Genstat workbook file (*.gwb) to be created
QSAVE = *pointers*	Saves a pointer with information and results for the significant effects
SAVE = *REML save structures*	Save the details of each REML analysis for use in subsequent VDISPLAY and VKEEP directives

Description

QSESTIMATE fits a final QTL model to estimate QTL effects in single-environment trials. It uses single observations per genotype as phenotypic data. The response variable must be specified by the TRAIT parameter, and the genotypes by the GENOTYPES parameter.The POPULATIONTYPE option must be set to specify the population from which the genotypes are derived. For recombinant inbred lines (POPULATIONTYPE = RIL), the NGENERATIONS option, must be set to supply the number of generations. For backcross inbred lines (POPULATIONTYPE = BCxSy), the NBACKCROSSES and NSELFINGS options must be set to define the number of backcrosses to the first parent and the number of selfings, respectively.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive effects can be included in the model by specifying dominance genetic predictors using the DOMINANCEPREDICTORS parameter (e.g. in a F2 population). In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified, the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked. The IDPARENTS parameter can supply labels to identify the parents.

The QTL model assumes genotypes as random and QTLs as fixed effects. Extra fixed effects can be specified using the FIXED option. The QTLSELECTED parameter must specify the set of QTLs, in the form of a variate containing the index number of the positions where the QTLs are located. When the DOMINANCEPREDICTORS parameter is set, the DOMSELECTED parameter supplies a logical variate containing one if the dominance predictor of the corresponding marker must be present in the model, and zero if the dominance predictor of the corresponding marker must be absent in the model. If DOMINANCEPREDICTORS is set but DOMSELECTED is not set, all the dominance predictors are included.

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the other settings correspond to those in the PRINT option of the REML directive. To be able to calculate the explained variance in the summary, the option POPULATIONTYPE must be set.

The QTL effects and their standard errors can be saved by the QEFFECTS and QSE parameters, respectively, and the fitted values and residuals can be saved by the FITTEDVALUES and

RESIDUALS parameters. These are saved in pointers that contain a single variate if only the ADDITIVEPREDICTORS parameter is specified, or two or three variates if the DOMINANCEPREDICTORS and/or ADD2PREDICTORS parameters are also specified. The Wald statistics, degrees of freedom and probabilities can be saved by the parameters WALDSTATISTICS, DFWALD and PRWALD, respectively.

The OUTFILENAME parameter can be used to save the Wald statistics and the QEFFECTS and QSE structures in a Genstat work book file in a sheet named STATISTICS. This parameter should not contain an extension as the extension is defined automatically as .gwb.

The QSAVE parameter can be used to save a pointer containing information and results for the significant QTLs. The elements of the pointer are labelled as follows to simplify their subsequent use:

'procedure'	stores the string 'QSESTIMATE' to indicate the source of the results,
'trait'	trait name,
'markernames'	marker names,
'chromosomes'	chromosomes,
'positions'	positions,
'envnames'	stores the string 'Experiment',
'waldstatistics'	wald statistics,
'prwald'	probability values of wald statistics,
'dfwald'	degrees of freedom of the wald statistics,
'qeffects'	QTL effects,
'qse'	standard errors of the QTL effects,
'%vexplained'	percentage explained variance,
'lowerci'	lower bound of confidence interval of estimated QTL position,
'upperci'	upper bound of confidence interval of estimated QTL position,
'posmin'	position of left flanking marker,
'posmax'	position of right flanking marker,
'idlfm'	marker name of left flanking marker,
'idrfm'	marker name of right flanking marker,
'posminci'	position of left flanking marker outside confidence interval,
'posmaxci'	position of right flanking marker outside confidence interval,
'idlfmci'	marker name of left flanking marker outside confidence interval,
'idrfmci'	marker name of right flanking marker outside confidence interval,
'locus'	index numbers of the significant QTLs, and
'neff'	number of additive and dominance predictors in the model.

The elements 'procedure', 'trait', 'markernames', 'chromosomes', 'envnames', 'idlfm', 'idrfm', 'idlfmci' and 'idrfmci' are text structures; 'positions', 'waldstatistics', 'prwald', 'dfwald', 'lowerci', 'upperci', 'posmin', 'posmax', 'posminci, 'posmaxci', 'vexplained' and 'locus' are variates; 'qeffects' and 'qse' are pointers (see parameters QEFFECTS and QSE); and 'neff' is a scalar.

The SAVE parameter can be used to save the REML save structure from the analysis for use with subsequent VKEEP and VDISPLAY directives.

Options: PRINT, POPULATIONTYPE, NGENERATIONS, NBACKCROSSES, NSELFINGS, FIXED, UNITFACTOR, MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: TRAIT, GENOTYPES, UNITERROR, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, MKLOCI, IDMGENOTYPES, IDPARENTS, QTLSELECTED, RESIDUALS, FITTEDVALUES, WALDSTATISTICS, PRWALD, QEFFECTS, QSE, OUTFILENAME, QSAVE, SAVE.

Method

QSESTIMATE fits the following models which include a set L of QTLs:

1) $y_i = \mu + \Sigma_{l \in L}\, x_{il}^{add}\, \alpha_l^{add} + G_i$

if only ADDITIVEPREDICTORS are specified

2) $y_i = \mu + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_l^{add} + x_{il}^{dom}\, \alpha_l^{dom}) + G_i$

if DOMINANCEPREDICTORS are also specified

3) $y_i = \mu + \Sigma_{l \in L}\, (x_{il}^{add}\, \alpha_l^{add} + x_{il}^{add2}\, \alpha_l^{add2} + x_{il}^{dom}\, \alpha_l^{dom}) + G_i$

if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_i is the trait value of genotype i, x_{il}^{add} are the additive genetic predictors of genotype i for locus l, and α^{add} are the associated effects. In models 2 and 3, x_{il}^{dom} are the dominance genetic predictors, and α_l^{add} are the associated effects. In model 3, x_{il}^{add} are the additive genetic predictors for maternal genotype i at locus l, x_{il}^{add2} are the additive genetic predictors for paternal genotype i, and α^{add} and α^{add2} are the associated effects. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). G_i is the residual unexplained genetic and environmental variation, which is assumed to follow a Normal distribution with mean 0 and variance σ^2.

Action with RESTRICT

Restrictions are not allowed.

Reference

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QSBACKSELECT, QSQTLSCAN, QFLAPJACK, QREPORT.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QSIMULATE

Simulates marker data and QTL effects for single and multiple environment trials (M.P. Boer & J.T.N.M. Thissen).

Options

PRINT = *string token*	What to print (summary); default summ
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
NGENERATIONS = *scalar*	Number of generations for a RIL population; default 3
NBACKCROSSES = *scalar*	Number of backcrosses for a BCxSy population; default 2
NSELFINGS = *scalar*	Number of selfings for a BCxSy population; default 3
GENOMELENGTH = *variate*	Length in cM for each chromosome
DISTANCE = *scalar*	Distance between the markers in cM; default 1 cM
COMPLETE = *string token*	Complete marker information, i.e. all parents have a different allele (yes, no); default no
FRACTIONMISSING = *scalar*	Fraction of the markers with missing values; default 0
NGENOTYPES = *scalar*	Number of genotypes; must be set
NCHROMOSOMES = *scalar*	Number of chromosomes
NPOSITIONS = *scalar*	Number of positions per chromosome
IDPARENTS = *texts*	Labels used to identify the parents
MEAN = *scalar* or *variate*	Mean of the trait for each environment; must be set if TRAIT is set
VARIANCE = *scalar* or *variate*	Variance of the trait for each environment; must be set if TRAIT is set
ADDITIVEEFFECTS = *variate* or *pointer*	Additive effects of each QTL for each environment; must be set if TRAIT is set
ADD2PREDICTORS = *pointers*	Second (paternal) set of additive genetic predictors of each QTL for each environment if POPULATIONTYPE is CP; must be set if TRAIT is set
DOMINANCEPREDICTORS = *pointers*	Dominance genetic predictors of each QTL for each environment if POPULATIONTYPE is F2 or CP; must be set if TRAIT is set
QTLCHROMOSOMES = *variate*	Chromosome number for each QTL; must be set if TRAIT is set
QTLPOSITIONS = *variate*	Position on the QTLCHROMOSOMES for each QTL; must be set if TRAIT is set

Parameters

TRAIT = *variates*	Saves the quantitative trait values
GENOTYPES = *factors*	Saves the genotype factor
ENVIRONMENTS = *factors*	Saves the environment factor
MKSCORES = *pointers*	Saves the marker scores for each marker
CHROMOSOMES = *factors*	Saves the linkage groups of the markers
POSITIONS = *variates*	Saves the position on the chromosome for each marker
MKNAMES = *texts*	Names of the markers
IDMGENOTYPES = *texts*	Labels of the genotypes
PARENTS = *pointers*	Saves the parent information
SEED = *scalars*	Specifies a seed to use for the random number generator;

default 0 continues from the previous generation or (if none) initializes the seed automatically

Description

QSIMULATE can be used to simulate marker data and/or QTL effects, for either single or multiple environment trials. The specification of the simulation is defined by the options, and the parameters save each set of simulated data.

The PRINT option controls printed output. There is a single setting, summary, which prints a summary of the simulations; this is the default.

The POPULATIONTYPE option must be set to specify the population type. For recombinant inbred lines (POPULATIONTYPE=RIL), the NGENERATIONS option specifies the number of generations; default 3. For backcross inbred lines (POPULATIONTYPE=BCxSy), the NBACKCROSSES option specifies the number of backcrosses, and the NSELFINGS option specifies the number of selfings.

The GENOMELENGTH option can be used to supply a variate specifying the length of the chromosomes, and DISTANCE option defines the distance between the markers in cM. Alternatively, instead supplying the GENOMELENGTH variate, you can define the genome using the NCHROMOSOMES and NPOSITIONS options. The NGENOTYPES option must be set to define the number of genotypes. If marker scores for the PARENTS are also to be simulated, the IDPARENTS option can be used to label the parents.

The MEAN option defines the overall mean of the trait, as scalar for a single environment or a variate for multi-environment trials. The VARIANCE option defines the error variance for the environments. The positions of the QTLs are defined by the QTLCHROMOSOMES and QTLPOSITIONS options. The ADDITIVEEFFECTS, ADD2EFFECTS and DOMINANCEEFFECTS options define the additive, second additive and dominance effects of the simulated QTLs.

Setting option COMPLETE=yes specifies that the parents all have a different allele for the markers. With the default setting, no the markers are assumed to be SNPs, and the marker score for a parent is either 1 or 2 with equal probabilities. For a bi-parental cross this means that around 50% of the markers will be polymorphic. The FRACTIONMISSING option can be used to define a fraction of missing marker scores. The default is that no scores are missing (FRACTIONMISSING=0).

The SEED parameter can be set to specify a seed for the random number generator for each set of simulated data. The MARKERSCORES parameter saves the simulated marker scores for all the markers and all the offspring. The MARKERNAMES parameter saves the names of the simulated markers. The IDMGENOTYPES parameter saves the labels of all the genotypes. The CHROMOSOMES and POSITIONS parameters save the chromosomes and positions of the simulated markers, and the PARENTS parameter saves the marker scores for the parents. The TRAIT parameter saves the simulated trait data, for all the genotypes and for all the environments. The ENVIRONMENTS and GENOTYPES factors save the corresponding environments and genotypes for the simulated TRAIT parameter.

Options: PRINT, POPULATIONTYPE, NGENERATIONS, NBACKCROSSES, NSELFINGS, GENOMELENGTH, DISTANCE, COMPLETE, FRACTIONMISSING, NGENOTYPES, NCHROMOSOMES, NPOSITIONS, IDPARENTS, MEAN, VARIANCE, ADDITIVEEFFECTS, ADD2EFFECTS, DOMINANCEEFFECTS, QTLCHROMOSOMES, QTLPOSITIONS.

Parameters: TRAIT, GENOTYPES, ENVIRONMENTS, MKSCORES, CHROMOSOMES, POSITIONS, MKNAMES, IDMGENOTYPES, PARENTS, SEED.

Method

QSIMULATE calls an external algorithm in the dynamic link library genetics.dll.

Action with `RESTRICT`
Restrictions are not allowed.

See also
Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QSQTLSCAN

Performs a genome-wide scan for QTL effects (Simple and Composite Interval Mapping) in single-environment trials (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, progress, model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default summ
PLOT = *string token*	Whether to plot the profile along the genome (profile); default prof
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
ALPHALEVEL = *scalar*	Defines a genome-wide significance level to calculate the threshold; default 0.05
COFACTORS = *variate*	Index numbers of loci to be used as cofactors for the genetic background
COFWINDOW = *scalar*	Specifies a window for cofactor exclusion from the model; default 10^6 which means that all cofactors on the same chromosomes are excluded
THRMETHOD = *string token*	Which method to use to calculate the threshold for QTL detection (bonferroni, liji, given); default liji
THRESHOLD = *scalar*	Threshold value for test statistic when THRMETHOD=given
DISTANCE = *scalar*	Distance between loci when THRMETHOD=bonferroni; default 4
FIXED = *formula*	Formula with extra fixed terms
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
STATISTICTYPE = *string token*	Which test statistic to plot and save using the STATISTICS parameter (wald, minlog10p); default minl
COLOURS = *scalar*, *variate* or *text*	Colours to use for the chromosomes; default * uses the colours of pens 1, 2 up to the number of chromosomes
TITLE = *text*	General title for plot
YTITLE = *text*	Title for the y-axis; default uses the identifier of the STATISTICS variate or pointer
XTITLE = *text*	Title for the x-axis; default 'Chromosomes'
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 100

Parameters

TRAIT = *variates*	Quantitative trait to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
UNITERROR = *variates*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted

ADDITIVEPREDICTORS = *pointers*	Additive genetic predictors; must be set
ADD2PREDICTORS = *pointers*	Second (paternal) set of additive genetic predictors
DOMINANCEPREDICTORS = *pointers*	Dominance genetic predictors
CHROMOSOMES = *factors*	Chromosomes corresponding to the genetic predictors; must be set
POSITIONS = *variates*	Positions on the chromosomes corresponding to the genetic predictors; must be set
IDLOCI = *texts*	Labels for the loci
IDMGENOTYPES = *texts*	Labels for the genotypes corresponding to the genetic predictors
IDEFFECTS = *texts*	Labels for the effects along the y-axis, in the frame below the profile plot
IDPARENTS = *texts*	Labels to use to identify the parents
QSTATISTICS = *variates*	Saves test statistics for QTL effects along the genome
QEFFECTS = *pointers*	Saves QTL effects along the genome (additive effects,and, if specified, also second additive and dominance effects)
QSE = *pointers*	Saves standard errors of the QTL effects
OUTFILENAME = *texts*	Name of the Genstat workbook file (*.gwb) to be created
DFILENAME = *texts*	Name of the graphics file for the plots

Description

QSQTLSCAN performs a genome-wide QTL scan in single-environment trials. It uses single observation per genotype as phenotypic data. The response variable must be specified by the TRAIT parameter, and the genotypes by the GENOTYPES parameter. The POPULATIONTYPE option must be set to specify the population type.

Molecular information must be provided in the form of additive genetic predictors stored in variates and supplied, in a pointer, by the ADDITIVEPREDICTORS parameter. Non-additive effects can be included in the model by using the DOMINANCEPREDICTORS parameter to specify dominance genetic predictors (e.g. in a F2 population); again they are stored in variates and supplied in a pointer. In the case of segregating F1 populations (outbreeders) two sets of additive genetic predictors must be specified: the maternal ones by the ADDITIVEPREDICTORS parameter, and the paternal ones by the ADD2PREDICTORS parameter. The corresponding map information for the genetic predictors must be given by the CHROMOSOMES and POSITIONS parameters. The labels for the loci can be supplied by the IDLOCI parameter, and the labels for the genotypes in the marker data can be supplied by the IDMGENOTYPES parameter. If IDMGENOTYPES is set, the match between the genotypes in the phenotypic and in the marker data will be checked.

The QTL detection model assumes genotypes as random and QTLs as fixed effects. Extra fixed effects can be specified using the FIXED option. The QTL search can be performed without cofactors (*Simple Interval Mapping*) or with cofactors that control for genetic background effects (*Composite Interval Mapping*). For Composite Interval Mapping, the COFACTORS option must specify a variate containing the index numbers of the loci designated as cofactors. The COFWINDOW option defines a window around a tested position within which cofactors are temporarily excluded from the model.

The MVINCLUDE, MAXCYCLE and WORKSPACE options operate in the same way as these options of the REML directive. The UNITERROR parameter allows uncertainty on the trait means (derived from individual unit or plot error) to be specified to include in the random model; by default this is omitted. The UNITFACTOR option allows the factor that is needed to define the

unit-error term to be saved (this would be needed, for example, to save information later about the term using VKEEP).

The method to define the threshold value is defined by the THRMETHOD option and uses a genome-wide error rate defined by the option ALPHALEVEL (default 0.05). If THRMETHOD=given, a user-defined threshold value must be specified using the THRESHOLD option. If THRMETHOD=bonferroni, an effective number of tests is calculated using the value specified by the DISTANCE option as the step size (default 4). Alternatively the liji setting uses the method described by Li & Ji (2005). See procedure QTHRESHOLD for details.

The PRINT option specifies the output to be displayed. The summary setting prints the information about the QTLs retained in the model, and the progress setting shows how the scan is progressing. The other settings correspond to those in the PRINT option of the REML directive.

By default QSQTLSCAN plots the test statistic associated with the effects of the genetic predictors against their position on the chromosomes, but you can set option PLOT=* to suppress this. The STATISTICTYPE option specifies what to plot along the y-axis of the upper plot, either the test statistic or the associated probability value (on a −log10 scale), and also defines what is saved in the variates specified by the QSTATISTICS parameter. The IDEFFECTS parameter can be used to label the effects, and the IDPARENTS parameter can supply labels to identify the parents.

The corresponding effects of each genetic predictor and their standard errors can be saved by the QEFFECTS and QSE parameters, respectively. These are saved in pointers that contain a single variate if only the ADDITIVEPREDICTORS parameter is specified, or two or three variates if the DOMINANCEPREDICTORS and/or ADD2PREDICTORS parameters are also specified. The TITLE, YTITLE and XTITLE options can specify the general title of the graph, the title of the y-axis and the title of the x-axis, respectively. The colours to use for the chromosomes in the upper graph are specified by the COLOURS option using either a text of colour names or a variate of RGB values (see the PEN directive for details). If COLOURS is not set, the default is to use the default colours of the pens 1, 2, onwards, up to the number of chromosomes. By default, the plot is sent to the screen. However, you can supply a file for the plot, using the DFILENAME parameter. You can discover the types of graphics file that are supported by running the command.

```
DHELP possible
```

The OUTFILENAME parameter can be used to write the QSTATISTICS, QEFFECTS and QSE structures to a Genstat work book file in a sheet named STATISTICS. This parameter should not contain an extension as the extension is defined automatically given as .gwb.

Options: PRINT, PLOT, POPULATIONTYPE, ALPHALEVEL, COFACTORS, COFWINDOW, THRMETHOD, THRESHOLD, DISTANCE, FIXED, UNITFACTOR, STATISTICTYPE, COLOURS, TITLE, YTITLE, XTITLE MVINCLUDE, MAXCYCLE, WORKSPACE.

Parameters: TRAIT, GENOTYPES, UNITERROR, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, CHROMOSOMES, POSITIONS, IDLOCI, IDMGENOTYPES, IDEFFECTS, IDPARENTS, QSTATISTICS, QEFFECTS, QSE, OUTFILENAME, DFILENAME.

Method

QSQTLSCAN fits the following mixed models repeatedly along the genome:

1) $y_i = \mu + \Sigma_{f \in F} x_{if} c_f + x_i \alpha_{lj} + G_i$

if only ADDITIVEPREDICTORS are specified

2) $y_i = \mu + \Sigma_{f \in F} (x_{if}^{add} c_f^{add} + x_{if}^{dom} c_f^{dom}) + (x_i^{add} \alpha^{add} + x_i^{dom} \alpha^{dom}) + G_i$

if DOMINANCEPREDICTORS are also specified

3) $y_i = \mu + \Sigma_{l \in L} (x_{if}^{add} c_l^{add} + x_{if}^{add2} c_l^{add2} + x_{if}^{dom} c_l^{dom})$

$+ (x_i^{add} \alpha^{add} + x_i^{add2} \alpha^{add2} + x_i^{dom} \alpha^{dom}) + G_i$

if both ADD2PREDICTORS and DOMINANCEPREDICTORS are specified (for population type CP)

where y_i is the trait value of individual i, F is a set of cofactors (if cofactors are included in the model), and x_{if}^{add} and x_i^{add} are the additive genetic predictors of genotype i at the cofactor positions and at the tested position, respectively. The associated effects are denoted by c_i^{add} and α^{add} for cofactors and tested position respectively. In model 2 and 3, x_{if}^{dom} and x_i^{dom} are dominance genetic predictors of genotype i at the cofactor positions and at the tested position, respectively, with associated effects c_f^{dom}, and α^{dom}. In model 3, x_{if}^{add} and x_i^{add} are the additive genetic predictors for the maternal genotype, for cofactors and tested position, respectively, and x_{if}^{add2} and x_i^{add2} are the equivalent additive genetic predictors for the paternal genotype. Finally x_{if}^{dom} and x_i^{dom} are the dominance genetic predictors for the cofactors and tested position, respectively. The associated effects are given by c_f^{add}, c_f^{add2} and c_f^{dom} for cofactors, and α^{add}, α^{add2} and α^{dom} for tested positions. Genetic predictors are genotypic covariables that reflect the genotypic composition of a genotype at a specific chromosome location (Lynch & Walsh 1998). The residual unexplained genetic and environmental effects are modelled by the G_i term, which is assumed to follow a Normal distribution with mean 0 and variance σ^2.

The procedure uses the REML directive iteratively to fit the model at each chromosome position, storing the Wald statistic for hypothesis testing. The resulting Wald statistic or the associated probability value (on a −log10 scale) can be plotted to produce the well-known profile plots used for interpretation.

Action with RESTRICT

Restrictions are not allowed.

Reference

Lynch, M. & Walsh, B. (1998). *Genetics and Analysis of Quantitative Traits*. Sinauer Associates, Sunderland, MA.

See also

Procedures: QSBACKSELECT, QSESTIMATE.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QTHRESHOLD

Calculates a threshold to identify a significant QTL (M.P. Boer & J.T.N.M. Thissen).

Options

PRINT = *string token*	What to print (summary); default summ
POPULATIONTYPE = *string token*	Type of population (BC1, DH1, F2, RIL, BCxSy, CP); must be set
THRMETHOD = *string token*	Which method to use (bonferroni, liji); default liji
STATISTICTYPE = *string token*	Which type of test statistic to use (wald, minlog10p); default minl
ALPHALEVEL = *scalar*	Defines the genome-wide significance level; default 0.05
DISTANCE = *scalar*	Distance between evaluation points for THRMETHOD=bonferroni; default 4
DF = *scalar*	Degrees of freedom for the Wald test; default 1

Parameters

CHROMOSOMES = *factors*	Chromosome for each locus; must be set
POSITIONS = *variates*	Position on the chromosome for each locus; must be set
ADDITIVEPREDICTORS = *pointers*	The additive genetic predictors
ADD2PREDICTORS = *pointers*	The second (paternal) additive genetic predictors if POPULATIONTYPE is CP
DOMINANCEPREDICTORS = *pointers*	The dominance genetic predictors if POPULATIONTYPE is F2 or CP
THRESHOLD = *scalars*	Saves the calculated threshold

Description

QTHRESHOLD calculates a genome wide significance threshold to use as a critical value to reject the null hypothesis of no QTL effect. The genome-wide type I error rate is defined by the option ALPHALEVEL. The threshold is based on a modified Bonferroni correction. The THRMETHOD option specifies the method for calculating the number of tests to used as the denominator. The default setting, liji, uses the effective number of independent tests, as described by Li & Ji (2005). Alternatively, the setting bonferroni assumes one independent test at every fixed distance on the genome, defined by the DISTANCE option (default 4 centiMorgans). By default, the threshold is expressed as the P value on a −log10 scale, but you can set option STATISTICTYPE=Wald to use the absolute Wald test statistic instead. Marker and map information must be supplied by the ADDITIVEPREDICTORS, CHROMOSOMES and POSITIONS parameters. The DOMINANCEPREDICTORS parameter can supply dominance genetic predictors for population types F2, RIL, BCxSy and CP, and the ADD2PREDICTORS parameter can supply the second (paternal) additive genetic predictors for population type CP. The corresponding degrees of freedom for the Wald test must be set by the DF parameter; this is equal to 1 in a single-environment QTL analysis, or to the number of environments in a multi-environment QTL analysis.

The calculated threshold can be saved using the THRESHOLD parameter. By default the threshold is printed, but you can suppress this by setting option PRINT=*.

Options: PRINT, POPULATIONTYPE, THRMETHOD, STATISTICTYPE, ALPHALEVEL, DISTANCE, DF.

Parameters: CHROMOSOMES, POSITIONS, ADDITIVEPREDICTORS, ADD2PREDICTORS, DOMINANCEPREDICTORS, THRESHOLD.

Method

QTHRESHOLD calculates a genome-wide significance threshold based on a modified Bonferonni correction, where the effective number of tests is used as the denominator instead of the total number of tests. By default the procedure estimates the effective number of independent tests by a singular value decomposition of the correlation matrix between all markers (see Li & Ji 2005 or Cheverud 2001). Alternatively, QTHRESHOLD assumes that the effective number of tests along the genome can be approximated by n independent tests:

n = ceiling(L/D)

where L is the total genome length (in cM), D is the distance between evaluation points (in cM) supplied by the DISTANCE option, and ceiling(x) gives the smallest integer not less than x. If the DISTANCE option is unset, n is set to the length of the CHROMOSOMES variate.

Using the Bonferonni correction, the genome wide significance threshold T is approximated by $X^2_{df}(1-\alpha/n)$, where df is the number of degrees of freedom.

Action with RESTRICT

Restrictions are not allowed.

References

Cheverud, J.M. (2001). A simple correction for multiple comparisons in interval mapping genome scans. *Heredity*, **87**, 52-58.

Li, J, & Ji, L. (2005). Adjusting multiple testing in multilocus analyses using the eigenvalues of a correlation matrix. *Heredity*, **95**, 221-227.

See also

Procedures: FDRBONFERRONI, FDRMIXTURE.

Genstat Reference Manual 1 Summary section on: Statistical genetics and QTL estimation.

QUANTILE

Calculates quantiles of the values in a variate (P.W. Lane).

Options

PRINT = *string token*	What to print (quantiles); default quan
METHOD = *string token*	Type of quantile to form (population, sample); default samp
PROPORTION = *variate*	or scalar Proportions at which to calculate quantiles; default !(0,0.25,0.5,0.75,1)

Parameters

DATA = *variates*	Values whose quantiles are required; this parameter must be specified
QUANTILES = *variates* or *scalars*	Identifiers of structures to store results, if required

Description

Quantiles are statistics that characterize a distribution. The DATA parameter supplies a sample of numbers $\{x_i, i=1...n\}$ from which the quantiles are to be calculated, and the METHOD option specifies the type of quantile to form.

By default QUANTILE calculates quantiles of the sample itself. For a proportion p in the range [0,1], the corresponding quantile q of the sample $\{x_i\}$ has the following properties:

1) at least the proportion p of $\{x_i\}$ are less than or equal to q;
2) at least the proportion $(1-p)$ of $\{x_i\}$ are greater than or equal to q;
3) if $q=x_i$ and $q=x_{i+1}$ satisfy 1) and 2), then take $q = (x_i+x_{i+1})/2$.

Thus the sample quantile for proportion 0.5 is the median, for 0.0 it is the minimum, and for 1.0 it is the maximum of the sample.

Alternatively, you can set METHOD=population to estimate quantiles of the underlying population from which data have been sampled. (This type of quantile is the one used most often elsewhere in Genstat.) The quantile is now an estimate of the value x such that a proportion p of the population has values less than or equal to x.

By default, QUANTILE produces the five quantiles called the "five-number summary" of a sample, corresponding to the proportions 0.0, 0.25, 0.5, 0.75, 1.0. The option PROPORTION can be set to a scalar or variate to request other single quantiles or sets of quantiles. By default, QUANTILE prints the statistics, but this can be suppressed by setting option PRINT=*. The quantiles can be stored in a variate using the parameter QUANTILES.

Options: PRINT, METHOD, PROPORTION.
Parameters: DATA, QUANTILES.

Method

With METHOD=sample, QUANTILE calculates the quantiles itself, using the SORT and CALCULATE directives. First, the values are sorted into ascending order. Then for each proportion, the two values that are candidates for the quantile are found, by counting from either end of the sorted list to leave the required number of values from that point in the list to the end. The quantiles are the averages of the two values found.

The alternative setting, METHOD=population, uses the Genstat QUANTILES function. QUANTILES assumes that the sorted data values are evenly distributed along the range of proportions, but with the lowest data value located at proportion $1/2n$, and the highest one located at proportion $1-1/2n$, where n is the size of the sample. (This recognises that sample is unlikely to contain the minumum and maximum values in the population.) If the required proportion p coincides with one of these sample proportions, QUANTILES estimates the quantile

as the corresponding data value. If not, QUANTILES finds the nearest sample point with a proportion below *p*, and the nearest one with a proportion above *p*. It then interpolates between these two points, i.e. it takes a weighted average of their data values, with weights given by the absolute difference between their proportions and *p*. However, if *p* lies outside (i.e. above or below) the sample proportions, QUANTILES does a linear extrapolation using the two nearest sample points.

Action with RESTRICT
If the DATA variate is restricted, the quantiles are formed only using the units that are not restricted out. The PROPORTION and QUANTILES variates must not be restricted.

See also
Directive: TABULATE.
Procedure: RQLINEAR.
Function: QUANTILES.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

QUESTION

Obtains a response using a Genstat menu (S.A. Harding & R.W. Payne).

Options

PREAMBLE = *text*	Text posing a question; (no default)
PROMPT = *text*	Text to be used as final prompt; the default prompt specifies the mode of response and lists the default values (if any), in brackets, followed by ">"
RESPONSE = *identifier*	Structure to store response; default * allows a menu to be saved without being executed
MODE = *string token*	Mode of response (p, t, v); default p
DEFAULT = *identifier*	Response to be assumed if just <RETURN> is given; default is to repeat the prompt until a response is obtained
LIST = *string token*	Whether a list of responses, rather than a single response, is valid (yes, no); default no
DECLARED = *string token*	Whether identifiers must already be declared (yes, no); default no
TYPE = *string tokens*	Allowed types for identifiers (ASAVE, datamatrix i.e. pointer to variates of equal lengths as required in multivariate analysis, diagonalmatrix, dummy, expression, factor, formula, LRV, matrix, pointer, RSAVE, scalar, SSPM, symmetricmatrix, table, text, tree, TSAVE, TSM, variate, VSAVE); default *, meaning no limitation
PRESENT = *string token*	Whether the identifier must have values (yes, no); default no
LOWER = *scalar*	Lower limit for numbers; default *, meaning no check
UPPER = *scalar*	Upper limit for numbers; default *, meaning no check
HELP = *text*	Text to be used in response to a general query for the question; default *
SAVE = *pointer*	Previously allowed you to save or reinput the specification of the menu, but is now no longer supported

Parameters

VALUES = *texts*	Possible codes for MODE t; (no default for MODE t; not relevant for others)
CHOICE = *texts*	Text giving explanation of each letter code; (no default for MODE t; not relevant for others)
HELP = *texts*	Text to be used in response to a specific query for a code; default *

Description

The QUESTION directive was replaced in the 16th Edition by the QDIALOG directive, with updated facilities appropriate to the more recent computing environments. This procedure uses QDIALOG to duplicate most of the facilities of the directive, so that existing programs can still run. The main difference is that the SAVE option and the option settings MODE=e and MODE=f are no longer supported.

So QUESTION displays a Genstat menu and obtains a response when in interactive mode. In batch, the procedure does nothing. Here is a simple example that asks the user to provide the

identifier of a variate structure.

```
QUESTION [PREAMBLE=!t('Y-VARIATE Menu (from ANOVA Menu)',*,\
  'What is the variate to be analysed ?'); RESPONSE=_yvar; \
  DECLARED=yes; TYPE=variate; PRESENT=yes]
```

The PREAMBLE option specifies a text structure, whose contents are printed at the beginning of the menu. Following this is the prompt: by default, this consists of a reminder of what type of answer is expected, followed by the greater-than symbol (>). However, there is a PROMPT option that allows any text to be printed instead, before the greater-than symbol.

The RESPONSE option specifies a dummy identifier that will point to the answer given by the user. Menus can request information in one of five modes. The default is Mode p (pointer), as here, and expects a response to consist of an identifier; but the MODE option can also be set to v (variate) or t (text). The earlier MODE settings of e (expression) and f (formula) are now faulted. When a correct answer has been received, an unnamed structure of the relevant type (pointer, variate, or whatever, but see later for text mode) is set up, and the dummy in the RESPONSE option is set to point at this unnamed structure.

Thus, if you give the identifier Y in response to the question above, the dummy _yvar will store the identifier of a pointer containing the single identifier Y. So the QUESTION statement could be followed by

```
ANOVA #_yvar
```

to do an analysis-of-variance of Y. The hash (#) is needed here to substitute the values of the unnamed pointer that is stored in the dummy structure _yvar.

By default, a question will expect to receive a single item of the specified mode: identifier, number, string, expression or formula. However, if the option LIST is set to yes for modes p, v or t, then a list of items is expected. The unnamed structure set up to store the answer will then contain as many values as there are items in the list.

The other three options in the example above specify restrictions on the answer that will be accepted. The DECLARED option specifies that the identifier must be of a structure that has already been declared. If a previously unused identifier is given, the QUESTION statement will print a warning, and issue the prompt again. Similarly, the TYPE option specifies what type of structure is acceptable; the setting may be a list of types if relevant. The PRESENT option specifies that the structure must already have values. Two further options, LOWER and UPPER, can be used to specify limits for numbers given in response to questions of mode v.

Menus of mode t resemble more closely what most people think of as a menu than the example above. These menus require extra information to be specified using parameters of the QUESTION procedure. The VALUES parameter should be set to a list of text structures, each of which stores a single string that is acceptable as an answer to the question. The CHOICE parameter should be set to another list of text structures, each storing a single string to be displayed by the side of the corresponding code in the menu to explain it. This example shows a question from procedure AGDESIGN.

```
QUESTION [PREAMBLE='Do you want to print the design?';\
         RESPONSE=pdes; MODE=t; DEFAULT='n'; LIST=no] \
         VALUES='n','y'; CHOICE='no','yes'
```

The codes must obey the rules for unquoted strings: that is, they must start with a letter and consist only of letters and digits. Only the first eight characters will be displayed, and only the first eight characters of the answer will be checked – all eight must match. Usually, of course, it is convenient to use single-letter codes.

Note that mode t cannot be used to ask the user for an arbitrary string, for example to provide a label for output. To request such information, you must use mode p, and set TYPE=text; the user must then supply the string in quotes, or supply the identifier of a text structure that already stores the string.

The response to a question of mode t is stored not as a text, but as a variate each value of

which is the number of the corresponding code as listed in the VALUES parameter. Usually, of course, a menu of mode t will be set with LIST=no, the default, and so the variate will contain only a single number. This can be used to control subsequent action in the menu system, for example with a CASE statement.

The DEFAULT option specifies a default answer to be used if the user just types RETURN, and can be set for any mode of question. The HELP option and parameter of the QUESTION procedure allow you to provide help text to guide the person answering the question.

The SAVE option, which allowed you to declare a menu without executing it or to execute a menu that has already been stored, is no longer supported.

Options: PREAMBLE, PROMPT, RESPONSE, MODE, DEFAULT, LIST, DECLARED, TYPE, PRESENT, LOWER, UPPER, HELP, SAVE.
Parameters: VALUES, CHOICE, HELP.

See also
Directive: QDIALOG.
Procedures: QFACTOR, QLIST.
Genstat Reference Manual 1 Summary sections on: Program control, Calculations and manipulation.

RADIALSPLINE

Calculates design matrices to fit a radial-spline surface as a linear mixed model (S.J. Welham & D.B. Baird).

Options

ORTHOGONALIZATION = *string token*	How to orthogonalize the random basis (`fixed`, `none`); default `fixe`
SCALING = *scalar*	Scaling of the `XRANDOM` terms (`automatic`, `none`); default `auto`

Parameters

X1 = *variates* or *factors*	Coordinates in the first dimension for which spline values are required
X2 = *variates* or *factors*	Coordinates in the second dimension for which spline values are required
XFIXED = *matrices*	Saves the design matrix to define the fixed terms (excluding the constant) for fitting the radial spline
XRANDOM = *matrices*	Saves the design matrix to define the random terms for fitting the radial spline
X1KNOTS = *variates*	Specifies the coordinates in the first dimension of the internal knots used to form the basis for the spline
X2KNOTS = *variates*	Specifies the coordinates in the second dimension of the internal knots used to form the basis for the spline
PX1 = *variates*	Specifies the coordinates in the first dimension at which to predict
PX2 = *variates*	Specifies the coordinates in the second dimension at which to predict
PFIXED = *matrices*	Saves the design matrix for the fixed terms (excluding the constant) for the radial spline at the prediction points
PRANDOM = *matrices*	Saves the design matrix for the random terms for the radial spline at the prediction points

Description

`RADIALSPLINE` generates the fixed and random terms required to fit a radial-spline surface as a linear mixed model, using `REML` estimation of the smoothing parameter. The coordinates at which the spline is to be calculated are specified in two variates using `X1` and `X2` parameters. The coordinates to be used as knots must be specified (in variates) using the `X1KNOTS` and `X2KNOTS` parameters.

The `ORTHOGONALIZATION` option specifies whether the components of the spline to be fitted as random terms should be made orthogonal to the components to be fitted as fixed. The default action (`ORTHOGONALIZATION=fixed`) is to perform the orthogonalization, and this means that all of the polynomial trend associated with the fixed terms will be captured in the fixed part of the model. When `ORTHOGONALIZATION=none`, some of this trend may be contained within the random terms.

The fixed and random components of the radial-spline terms are saved separately. The terms required to be fitted as fixed terms can be saved (in a matrix) using `XFIXED` parameter. This matrix does not include the constant term as this is added by default as part of a mixed model. The terms to be fitted as random can be saved (in a matrix) using the `XRANDOM` parameter.

The random terms can be scaled so that, for a random spline matrix `Z`,

```
TRACE(Z *+ T(Z)) = NROWS(Z)
```

This ensures that the average contribution of each component to the variance of an observation is equal to one. This improves interpretability of the spline variance components.

The radial-spline terms required for prediction can be saved using the PXFIXED and PXRANDOM parameters. The PX1 and PX2 parameters provide the coordinates at which predictions are to be made.

Options: ORTHOGONALIZATION, SCALING.
Parameters: X1, X2, XFIXED, XRANDOM, X1KNOTS, X2KNOTS, PX1, PX2, PFIXED, PRANDOM.

Method

This procedure calculates a low-rank thin-plate spline in two dimensions, following an approach equivalent to that of Section 13.5 in Ruppert, Wand & Carroll (2003). The fixed terms comprise the input variates, X1 and X2. For r knots at co-ordinates

$$t_j = (t_{j1} \ldots t_{jr})', \quad j=1,2$$

and input variates

$$x_i = (x_{i1} \ldots x_{in})', \quad i=1,2$$

the random basis functions are calculated via a function η (Green & Silverman, 1994), where

$$\eta(z) = z^2 \times \log(z^2) / (16 \times \pi) \quad \text{for } z > 0$$
$$= 0 \quad \text{for } z = 0.$$

The r random basis functions then take the form

$$b_l(x_1,x_2) = \eta([(x_1 - t_{1l})^2 + (x_2 - t_{2l})^2]^{1/2}) \quad \text{for } l = 1 \ldots r.$$

These columns can be concatenated into a $n \times r$ matrix E_t. The corresponding $r \times r$ penalty matrix K has entries

$$K[i,j] = \zeta([(t_{1i} - t_{1j})^2 + (t_{2i} - t_{2j})^2]^{1/2}) \quad \text{for } i,j = 1 \ldots r.$$

The matrix K can be transformed to full rank as

$$H = C' K C$$

where matrix C contains the eigenvectors of $X X'$ ($X = [1 \; x_1 \; x_2]$) corresponding to zero eigenvalues, with corresponding transformation of the matrix functions as

$$E_u = E_t C$$

This is translated to a set of independent random effects via post-multiplication by $H^{-1/2}$.

The design matrices for use in prediction are calculated by evaluating the same set of basis functions at the predict points.

Action with RESTRICT

Input structures must not be restricted.

References

Green, P.J. & Silvername, B.W. (1994). *Nonparametric Regression and Generalized Linear Models*. Chapman & Hall, London.

Ruppert, D., Wand, M.P. & Carroll, R.J. (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge.

See also

Directives: VCOMPONENTS, REML.
Procedures: NCSPLINE, PENSPLINE, PSPLINE, SPLINE, TENSORSPLINE.
Function: SSPLINE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

RANK

Produces ranks, from the values in a variate, allowing for ties (J.B. van Biezen & C.J.F. ter Braak).

Option

OMIT = *string token*	Whether units excluded by a restriction on the DATA variate should be omitted from the RANKS variate (restricted); default *, i.e. the units are not omitted, and their values are left unchanged

Parameters

DATA = *variates*	Variate containing values to be ranked
RANKS = *variates*	Variate to save vector of ranks
TIESIZE = *variates*	Variate to save the sizes of ties

Description

RANK calculates ranks of the values in a variate, allowing for ties. The variate must be specified by the DATA parameter, and the ranks saved using the RANKS parameter. The input variates in the parameter DATA must each have at least one non-missing value. Missing values in the DATA variates give corresponding missing values in the RANKS variates. The TIESIZE parameter can save the number of times each value occurs (starting with the lowest value).

The OMIT option controls whether the RANKS variate omits units that are excluded by a restriction on the DATA variate. By default, the values in these units are left unchanged. However, if OMIT=restricted, the RANKS variate is compressed to omit the excluded units; this setting is used particularly by the nonparametric procedures.

Option: OMIT. Parameters: DATA, RANKS, TIESIZE.

Method

The procedure uses the SORT directive to discover the number of distinct values in the input variate, and then uses TABULATE to obtain the number of times each value occurs – i.e. the size of ties. The tie-corrected rank numbers are then calculated. From these, the vector of ranks is obtained by modifying the levels of the factor that resulted from the first SORT.

Action with RESTRICT

The variates in DATA can be restricted, and in different ways. RANK operates on the restricted set only. If OMIT=restricted, the length of RANKS will be the size of the restricted set of the DATA variate. If OMIT is unset, the RANKS variate is of the same length as the DATA variate.

See also

Procedure: SORT.
Function: RANKS, SORT .
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

RAR1

Fits regressions with an AR1 or a power-distance correlation model (R.W. Payne).

Options

PRINT = *string tokens*	What to print (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, cparameter, cmonitoring, cplot); default mode, summ, esti, cpar
CALCULATION = *expression structures*	Calculation of explanatory variates involving nonlinear parameters
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalars*	Limit for expansion of model terms; default 3
POOL = *string token*	Whether to pool ss in accumulated summary between all terms fitted in a linear model (yes, no); default no
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default *
FPROBABILITY = *string token*	Printing of probabilities for variance and deviance ratios (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-statistics (yes, no); default no
SELECTION = *string tokens*	Statistics to be displayed in the summary of analysis produced by PRINT=summary, seobservations is relevant only for a Normally distributed response, and %cv only for a gamma-distributed response (%variance, %ss, adjustedr2, r2, seobservations, dispersion, %cv, %meandeviance, %deviance, aic, bic, sic); default %var, seob if DIST=normal, %cv if DIST=gamma, and disp for other distributions
SELINEAR = *string token*	Whether to calculate s.e.s for linear parameters when nonlinear parameters are also estimated (yes, no); default no
WEIGHTS = *variate*	Prior weights for the units
CMETHOD = *string token*	Estimation method (maximumlikelihood, reml); default maxi
CPARAMETER = *scalars*	Correlation parameter
CPOSITIONS = *variate*	Correlation positions
CGROUPS = *factor*	Groupings of correlation positions
MAXCYCLE = *scalars*	Maximum number of iterations; default 100
TOLERANCE = *scalars*	Convergence criterion; default 10^{-5}

Parameter

TERMS = *formula*	Terms to be fitted

Description

RAR1 allows you to fit regression and nonlinear models to data, such as repeated measurements, where the residuals may follow an AR1 or a power-distance correlation model. The CPOSITIONS option specifies the coordinates of the observations in the direction (e.g. time) along which the correlation model operates. You can also use the CGROUPS option to specify a factor to define groups of observations for the model – the correlation model is then defined only over the observations that belong to the same groups. The parameter *phi* of the AR1 or power-distance model is estimated within RAR1, and is assumed to be the same for every group. (Note that the model will be AR1 if the observations are each one unit apart within each group – the power-distance model is the natural extension of the AR1 model to unequally-spaced data; see Method.) You can save the estimated value of *phi*, in a scalar, using the CPARAMETER option.

Otherwise, RAR1 is used much like FIT. It must be preceded by a MODEL statement. You can also give an RCYCLE statement first if you want to estimate nonlinear parameters. The MODEL statement must have the WEIGHT option set to a symmetrix matrix, which need not have any values defined. RAR1 will set the values according to the distances (CPOSITIONS), groups (CGROUPS) and estimated parameter *phi*. These values remain set after RAR1. So you can display or save further output using RCHECK, RDISPLAY, RGRAPH or RKEEP, in the usual way. You could also, for example, use RAR1 to fit a full set of regression terms, and then use DROP to investigate smaller models while still using the *phi* estimate from the full model. RAR1 has a TERMS parameter to specify the terms to be fitted, like the parameter of FIT. It also has options CALCULATION, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION and SELINEAR which operate like those of FIT.

The PRINT option is also similar, except that it has three additional settings:

cparameter	prints the estimated value of the correlation *phi*, together with a test for *phi*=0,
cmonitoring	provides monitoring information for the estimation of *phi*,
cplot	plots the likelihood (or REML likelihood) for *phi*.

Note, the likelihood values omit some constant terms that depend only on the regression terms. The default is PRINT=model,summary,estimates,cparameter.

The other options control the estimation. The CMETHOD option controls whether *phi* is estimated for regression models by REML or by maximum likelihood (default maxi); with nonlinear models only maximum likelihood is available. The MAXCYCLE option defines the maximum number of iterations (default 100) used to estimate *phi*, and the TOLERANCE option specifies the convergence criterion i.e. the accurary to which *phi* is to be estimated (default 10^{-5}).

Options: PRINT, CALCULATION, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION, SELINEAR, WEIGHTS, CMETHOD, CPARAMETER, CPOSITIONS, CGROUPS, MAXCYCLE, TOLERANCE.
Parameter: TERMS.

Method

To estimate *phi* RAR1 uses procedure MIN1DIMENSION, which calls a procedure _MIN1DFUNCTION, which is loaded automatically with RAR1. _MIN1DFUNCTION uses the FIT directive to fit the regression model for a particular value of *phi*, and then evaluates the likelihood or REML likelihood (according to the setting of the CMETHOD option).

The total degrees of freedom for the regression are decreased by one, to take account of the estimation of the correlation parameter *phi*, by setting a variable in the regression save structure (rsave[1][3]$[47]) to one.

Action with RESTRICT

Restrictions are not allowed.

See also
Directive: `VSTRUCTURE`.
Procedure: `NLAR1`.
Genstat Reference Manual 1 Summary sections on: Repeated measurements, Regression analysis.

RBRADLEYTERRY

Fits the Bradley-Terry model for paired-comparison preference tests (R.W. Payne).

Options

PRINT = *string tokens*	What to print (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, confidence, preferenceprobabilities); default mode, summ, esti
GROUPS = *factor*	Factor representing different test circumstances
COVARIATE = *variates*	Other covariates to include in the model
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default *
FPROBABILITY = *string token*	Printing of probabilities for variance and deviance ratios (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-statistics (yes, no); default no
SELECTION = *string tokens*	Statistics to be displayed in the summary of analysis produced by PRINT=summary (%variance, %ss, adjustedr2, r2, dispersion, %meandeviance, %deviance, aic, bic, sic); default disp
DISPERSION = *scalar*	Dispersion parameter to be used as estimate for variability in s.e.s etc; default 1
PROBABILITY = *scalar*	Probability level for confidence intervals for parameter estimates; default 0.95

Parameters

WINNERS = *factors*	Specifies the winners in the tests
LOSERS = *factors*	Specifies the loser in the tests
NWINS = *variates* or *scalars*	Number of wins; default 1
NBINOMIAL = *variates* or *scalars*	Number of trials; default 1
PREFERENCEPROBABILITIES = *matrices* or *pointers*	Saves the estimated probability that each object is preferred to other objects
LOWERPREFERENCEPROBABILITIES = *matrices* or *pointers*	Saves the lower values of the confidence intervals for the preference probabilities
UPPERPREFERENCEPROBABILITIES = *matrices* or *pointers*	Saves the upper values of the confidence intervals for the preference probabilities
SAVE = *identifiers*	To save the regression save structure

Description

In a paired-comparison trial, assessers are given pairs of objects to assess and asked to indicate which of the two they prefer. They occur, for example, in sensory testing of food items, where the aim may be to establish preferred recipes or methods or cooking. Many other activities, including sports matches (where the items are teams that complete in pairs), can be analysed in the same way.

The results of the trial are specified by the WINNERS, LOSERS, NWINS and NBINOMIAL parameters. You can specify the comparisons individually, by setting the WINNERS and LOSERS

parameters to a pair of factors, with a unit for every competition. WINNERS specifies the object that was preferred, and LOSERS specifies the one with which it was compared.

Alternatively, it is more efficient to group the comparisons between each pair of objects together. You nominate one as winner and the other as loser, and record them in the corresponding element of the WINNERS and LOSERS factors. You define the number of times that they were compared in a variate to be specified by the NBINOMIAL parameter, and the number of wins in a variate to be specified by the NWINS parameter.

The data are analysed using the Bradley-Terry model (Bradley & Terry 1952), which is fitted as a generalized linear model with binomial distribution and logit link. The underlying assumption is that each item has an underlying "ability" score, which is estimated by the analysis on the log scale. The logit of the probability that one item is preferred to another is estimated by the difference in their estimated scores. For further details, see the *Methods* Section.

The COVARIATE option allows you to specify additional covariates to include in the model. The GROUPS option can specify a factor to define different trials; different ability scores are then estimated for each group. The other options (PRINT, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION, DISPERSION and PROBABILITY) all operate as in the standard regression directives like FIT etc, except that the PRINT option has an additional setting preferenceprobabilities to print a matrix showing the probability that each object is preferred to every other one. These can also be saved using the PREFERENCEPROBABILITIES parameter, and lower and upper values of their confidence intervals can be saved using the LOWERPREFERENCEPROBABILITIES and UPPERPREFERENCEPROBABILITIES parameters. If there are no groups, each of these saves a matrix, with losers on the rows and winners on the columns. If there are groups, they save pointers containing a matrix for each group.

After RBRADLEYTERRY you can use the standard regression output commands, RDISPLAY, RKEEP and so on, in the usual way. The SAVE parameter allows you to save the regression save structure.

Options: PRINT, GROUPS, COVARIATE, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION, DISPERSION, PROBABILITY.

Parameters: WINNERS, LOSERS, NWINS, NBINOMIAL, PREFERENCEPROBABILITIES, LOWERPREFERENCEPROBABILITIES, UPPERPREFERENCEPROBABILITIES, SAVE.

Method

The model assumes that each object *i* has an underlying "ability" score, τ_i say, and that the probability that object i is preferred to object j is given by

$$\begin{aligned} p_{ij} &= \tau_i / (\tau_i + \tau_j) \\ &= (\tau_i / \tau_j) / (1 + (\tau_i / \tau_j)) \\ &= \exp(\lambda_i - \lambda_j) / (1 + \exp(\lambda_i - \lambda_j)) \end{aligned}$$

where $\lambda_i = \log(\tau_j)$. So

$$p_{ij} / (1 - p_{ij}) = \exp(\lambda_i - \lambda_j)$$

and therefore

$$\text{logit}(p_{ij}) = \lambda_i - \lambda_j.$$

Action with RESTRICT

You can analyse a subset of the data by restricting any of the factors or variates in the data set.

Reference

Bradley, R.A., Terry, M.E. (1952). Rank analysis of incomplete block designs I: The method of paired comparisons. *Biometrika*, **39**, 324-45.

See also
Procedures: GENPROCRUSTES, SAGRAPES.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RCATENELSON

Performs a Cate-Nelson graphical analysis of bivariate data (V.M. Cave).

Options

PRINT = *string tokens*	Controls printed output (summary, quadrants, errorquadrants); default summ, quad
PLOT = *string tokens*	What graphs to plot (catenelson, criticalvalues); default cate
DIRECTION = *string token*	Direction of the association between the y and x values (ascending, descending); default asce i.e. a positive trend
YCRITICAL = *scalar*	Pre-specified critical value of y; default * i.e. the critical value of y is estimated)
XCRITICAL = *scalar*	Pre-specified critical value of x; default * i.e. the critical value of x is estimated
TITLE = *text*	Title for the Cate-Nelson plot; if unset, the title is generated automatically
YTITLE = *text*	Y-axis title for the Cate-Nelson plot; if unset, the title is generated automatically
XTITLE = *text*	X-axis title for the Cate-Nelson plot; if unset, the title is generated automatically
WINDOW = *scalar*	Window to use for the graphs; default 3
SAVE = *identifier*	Specifies the save structure of regression model holding the y-values, distribution, link function and weights; default * i.e. that from last regression fitted

Parameters

X = *variates*	Supplies the x-values for each analysis
RESULTS = *pointers*	Saves the critical value of x, the critical value of y and the quadrant allocations for each X variate

Description

The RCATENELSON procedure performs a graphical analysis of bivariate data (x,y) as defined by Cate & Nelson (1971). It also extends their analysis to y-variates with non-Normal distributions.

Before using RCATENELSON, you need to give a MODEL statement defining the y-variate. The distribution of the y-variate, a link function and weights can also be defined with the MODEL statement. (Note, however, that multinomial distributions, user-defined distributions and link functions and generalized least squares are not accommodated by RCATENELSON.) The variate containing the x-values is supplied using the X parameter.

The objective of the Cate-Nelson graphical analysis is to divide the data into two groups, based on the x-values, so that there is maximum statistical homogeneity within each group. The procedure finds the value of x that, in terms of predictive ability, best divides the data into two groups. This critical value of x is determined by iteratively dividing the data into two groups at each candidate critical x-value and selecting the one that minimizes the residual sum of squares, or the deviance for distributions other than the Normal. Alternatively, a pre-specified critical value of x may be supplied, as a scalar, using the XCRITICAL option.

After determining the critical value of x, the procedure then finds the critical value of y. (For the Binomial distribution, y is defined as the proportion of successes.) The critical values of x and y split the scatter plot of y on x into four quadrants: two of these contain data that follow the predictive model, and two (known as the *error quadrants*) contain data do not follow the model. The critical value of y is also determined iteratively, but here the critical value minimizes the

number of observations that fall into error quadrants, i.e. those that do not conform with the predictive model. Alternatively, a pre-specified critical value of y may be supplied, as a scalar, using the YCRITICAL option.

The DIRECTION option specifies whether the association between the y and x values is ascending (i.e. following a positive trend; the default) or descending (i.e. following a negative trend). This determines the error quadrants. For an ascending trend (i.e. where y increases with increasing x), observations in the top left (I) and in the bottom right (III) quadrants do not conform with the predictive model. Therefore, for data with an ascending trend, the critical y-value minimizes the number of observations that fall into Quadrants I and III. Conversely, for a descending trend (i.e. y where decreases with increasing x), the error quadrants are the top right (II) and bottom left (IV).

When there is more than one candidate critical x-value, or more than one candidate critical y-value, results are generated for each possibility.

Printed output is controlled by the PRINT option, with the following settings.

summary	prints a summary of the analysis, including the critical x-value, the critical y-value, the error rate (i.e. the percentage of observations falling into the two error quadrants) and the count and percentage of observations in each quadrant.
quadrants	prints the allocation of data to each quadrant.
errorquadrants	prints the data falling into the error quadrants.

The PLOT option controls the graphical output, with these settings.

catenelson	produces a Cate-Nelson plot. Here, a scatter plot of y on x is drawn, with a horizontal line superimposed through the critical value of y, and a vertical line superimposed through the critical value of x, splitting the data into four quadrants. Observations that fall into the error quadrants are drawn as red crosses, labelled by their unit number. Observations that followed the predictive model are drawn as black hollow circles.
criticalvalues	produces a plot of the residual sum of squares (or deviance for non-Normal distributions) against the candidate critical values of x, and a plot of the number of observations falling into the error quadrants against the candidate critical values of y. If XCRITICAL is supplied, no residual diagnostic plot will be produced for the residual sum of squares or deviance. If YCRITICAL is supplied, no diagnostic plot will be produced for the error quadrants.

By default, the Cate-Nelson plot is produced.

The TITLE, YTITLE and XTITLE options can supply an overall title, a y-axis title and a x-axis title for the Cate-Nelson plot, respectively. If these are not supplied, suitable titles are generated automatically. To omit a title, a blank string can be supplied, e.g.

```
XTITLE=' '
```

The WINDOW option defines the window to use for the plots; default 3.

Results can be saved using the RESULTS parameter. They are in a single pointer if there is only one critical x and critical y value. If there are several, they are in a pointer containing a pointer for each pair of critical x and critical y values. The first element of these pointers, indexed by 'Critical x-value', is a scalar storing the critical value of x. The second element, indexed by 'Critical y-value', is a scalar storing the critical value of y. The third element, indexed by 'Quadrant', stores the allocation of data to each quadrant, and is ordered by the unit number.

Options: PRINT, PLOT, DIRECTION, YCRITICAL, XCRITICAL, TITLE, YTITLE, XTITLE, WINDOW, SAVE.
Parameters: X, RESULTS.

Method

RCATENELSON uses the methods described in Cate & Nelson (1971) and Mangiafico (2013), but extended to accommodate y-variates with non-Normal distributions.

Candidate critical values of x are formed by ordering the unique values in X, and calculating the midpoint between each adjacent pair. Following Cate & Nelson (1971), the procedure ensures that at least two x-values fall to the left and to the right of each candidate value. The critical value of x minimizes the Residual Sum of Squares, or deviance for non-Normal distributions, which is obtained using the MODEL and FIT directives.

Candidate critical values of y are formed by ordering the unique values in Y, and calculating the midpoint between each adjacent pair. (For the Binomial distribution, the proportion of successes is used.) The critical value of y minimizes the number of observations in the error quadrants.

Action with RESTRICT

RCATENELSON will work with restricted X variates, and restricted Y, NBINOMIAL and WEIGHTS settings of MODEL. However, if more than one is restricted, they must be restricted in the same way.

References

Cate, R.B. & Nelson, L.A. (1971). A simple statistical procedure for partitioning soil test correlation data into two classes. *Soil Science Society of America Proceedings*, **35**, 658–660.

Mangiafico, S.S. (2013). Cate-Nelson analysis for bivariate data using R-project. *Journal of Extension*, **51**, 5TOT1.

See also

Genstat Reference Manual 1 Summary sections on: Graphics, Regression analysis.

RCHECK

Checks the fit of a linear, generalized linear or nonlinear regression (P.W. Lane, R. Cunningham & C. Donnelly).

Options

PRINT = *string tokens*	What to print (index, y, residuals, leverages, Cook); default *
RMETHOD = *string token*	Type of residual to use (deviance, Pearson, simple, deletion); default * i.e. as set in MODEL
INDEX = *variate* or *factor*	Which variable to use as index; default !(1...n)
ENVELOPE = *string token*	Type of envelope with Normal and half-Normal plots (none, rough, smooth, asymptotic); default none
PROBABILITY = *scalar*	Approximate probability level for envelope; default 0.95
NSIMULATIONS = *scalar*	How many simulations to generate for rough or smooth envelopes; default (1+PROB)/(1-PROB)
SHADE = *string token*	Whether to show shaded envelope rather than boundaries (no, yes); default no
RESIDUALS = *variate*	To store chosen type of residuals; default *
LEVERAGES = *variate*	To store leverages; default *
COOK = *variate*	To store modified Cook's statistics; default *
GRAPHICS = *string token*	Type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	Title for graph; default identifier of response
WINDOW = *numbers*	Window or series of windows in which to display graphs; default 4, or 5...8 for composite
SCREEN = *string token*	Treatment of previous graphics screen (clear, keep); default clea
SAVE = *regression save structure*	Specifies which model to check; default *

Parameters

YSTATISTIC = *string tokens*	What to display in the graph (residuals, Cook, leverages, absresiduals); default resi
XMETHOD = *string tokens*	What type of graph (fittedvalues, index, normal, halfnormal, histogram, composite); default comp

Description

Procedure RCHECK provides "diagnostic" information for checking the fit of regression models. Those directives make some checks, such as for large residuals and influential points, and give access to simple and standardized residuals and leverages through directive RKEEP. The RCHECK procedure automatically accesses these quantities via RKEEP and in addition can calculate deletion residuals and modified Cook's statistics. A range of graphs can then be drawn to help check the fit of the regression model. The defaults are intended to provide a sensible display from the simple command

```
RCHECK
```

following the fit of a regression model.

The procedure is controlled by the YSTATISTIC and XMETHOD parameters. These can be set to display various types of residuals, as specified by the RMETHOD option; the default is the setting of this option in the MODEL command in force when the model was fitted. In addition, the absolute residuals, the leverages, or the modified Cook's statistics can be displayed. Each of these sets of statistics can be plotted against the fitted values or against an index variable; by default, the index just orders the values in the order of the units. The statistics can also be shown

as Normal or half-Normal plots, or as a histogram (the Normal plot for absolute residuals being the same as the half-Normal plot). A set of four such plots is displayed as a composite picture: histogram, plot against fitted values, Normal plot and half-Normal plot (with an index plot replacing the Normal plot for absolute residuals). Graphs can be displayed in line-printer style by setting the GRAPHICS option, though some features are not then available.

The chosen type of residuals, the leverages and Cook's statistics can be printed, or stored in variates using the RESIDUALS option.

Plots of the residuals against fitted values or an index variable are displayed with a smoothed line fitted through the points, to indicate any potential trend.

Normal and half-Normal plots can be enhanced with an "envelope" by setting the ENVELOPE option. The rough setting produces an upper and lower bound for the values, and a median line, produced by simulation. The bounds correspond approximately to individual confidence intervals for each value, with probability as set by the PROBABILITY option (default 95%). The number of simulations by default is the minimum to allow estimation of the required limits: this is (1+PROBABILITY) / (1−PROBABILITY). A larger number of simulations can be requested with the NSIMULATIONS option, to give better estimates at the expense of more computing time. The smooth setting requests that the bounds are smoothed, using a cubic smooting spline with 4 d.f. The asymptotic setting produces bounds calculated from the asymptotic distribution of Normal order statistics. The envelope for all these settings can be displayed as a shaded region rather than as a set of three lines by setting the SHADE option to yes.

Envelopes cannot be calculated for nonlinear models or curves, nor for generalized linear models with inverse Normal, negative binomial, geometric, multinomial or calculated distributions. Nor can they be produced for deletion residuals or Cook's statistics; they are not appropriate for leverages, which have no associated distributional assumption.

The graphical displays can be controlled as usual using the TITLE and SCREEN options. The WINDOW option can be used to select a defined windows for high-resolution plots. Otherwise window 4 is used for a single plot or windows 5-8 for composite plots. These are redefined if necessary to fill the frame.

The colours and symbols used in the displays can be controlled by setting the attributes of the following pens with the PEN directive before calling the procedure:

pen 2	zero lines in fitted-value, Normal and index plots;
pen 3	points and histogram bars;
pen 4	shading of envelopes;
pen 5	smooth line in fitted-value and index plots of residuals, and envelope bounds if unshaded.

The procedure exits if there are fewer than four observations, or fewer than two non-missing standardized residuals.

Options: PRINT, RMETHOD, INDEX, ENVELOPE, NSIMULATIONS, PROBABILITY, SHADE, RESIDUALS, LEVERAGES, COOK, GRAPHICS, TITLE, WINDOW, SCREEN, SAVE.
Parameters: YSTATISTIC, XMETHOD.

Method

Standardized residuals and leverages are accessed using RKEEP from the latest fitted regression model, or from that specified by the SAVE option. Deletion residuals d_i are calculated for linear models as follows:

$$d_i = r_i / \sqrt{((n-p-r_i^2)/(n-p-1))}$$

where r_i are the standardized residuals, n is the number of observations, and p is the number of parameters in the model. For generalized linear models,

$$d_i = \mathrm{SIGN}(rd_i) \times \sqrt{((1-l_i) \times rd_i^2 + l_i) \times rp_i^2)}$$

where rd_i and rp_i are the standardized deviance and Pearson residuals respectively.

Modified Cook's statistics c_i are calculated as follows:

c_i = ABS(d_i) × √{ ($n-p$) × l_i / (p × (1−l_i)) }

where l_i are the leverages. In Normal plots, the Normal quantiles are calculated as follows:

q_i = NED((i−0.375) / (n+0.25))

while for a half-Normal plot they are given by

q_i = NED(0.5 + 0.5 × (i−0.375) / (n+0.25))

For generalized linear models, fitted values are transformed by an approximate variance-stabilizing transformation before use in graphs:

Poisson, multinomial, negative binomial and geometric	2 × SQRT(*fitted*)
binomial, Bernoulli	2 × ANG(100 × *fitted* / *nbinomial*)
gamma, exponential	LOG(*fitted*)
inverse Normal	1 / *fitted*

The smoothed line displayed for fitted-value or index plots is calculated as a straight line if the number n of distinct explanatory values is >3. Otherwise it is a cubic smoothing spline, with 2 d.f. for n>9, 3 for n>34 or 4 for n>59.

For Normal linear models, envelopes are calculated by default from *ns* sets of Normal random numbers, where

ns = (1 + PROBABILITY) / (1 − PROBABILITY).

If the number of observations is less than 100, the values are transformed using the projection matrix to induce the observed correlation pattern of the data; for larger datasets, no transformation is done. The values are then ordered and the minimum and maximum values determine the envelope boundaries. If *ns* is set by the NSIMULATIONS option, the boundaries are calculated with the QUANTILES function from the *ns* values generated for each ordered residual. For generalized linear models, *ns* sets of values of the response variate are generated from the distribution, with parameters estimated from the current fit. The model is refitted to each set, and the residuals extracted and dealt with as for the transformed Normal values above.

Action with RESTRICT

Restrictions applied to vectors used in the regression apply also to the RCHECK procedure. Values of diagnostic quantities are set to missing for all excluded units.

See also

Procedures: RDESTIMATES, RGRAPH, APLOT, DRESIDUALS, VPLOT.

Genstat Reference Manual 1 Summary section on: Regression analysis.

RCIRCULAR

Does circular regression of mean direction for an angular response (P.W. Goedhart).

Options

PRINT = *string tokens*	What to print (model, summary, estimates, fittedvalues, monitoring); default mode, summ, esti
FACTORIAL = *scalar*	Limit for expansion of model terms; default 3
RESIDUALS = *variate*	To save the residuals
FITTEDVALUES = *variate*	To save the fittedvalues, i.e. the fitted mean directions
LEVERAGES = *variate*	To save the leverages
ESTIMATES = *variate*	To save estimates of linear parameters
SE = *variate*	To save standard errors of the estimates
VCOVARIANCE = *symmetric matrix*	To save the variance-covariance matrix of the estimates
MU0 = *scalar*	To save the estimate of the mean parameter μ_0
SEMU0 = *scalar*	To save the standard error of the estimated mea parameter μ_0
KAPPA = *scalar*	To save the estimate of the concentration parameter κ of the von Mises distribution
SEKAPPA = *scalar*	To save the standard error of the estimated concentration parameter κ
_2LOGLIKELIHOOD = *scalar*	To save the value of minus twice the maximized log likelihood
DF = *scalar*	To save the residual degrees of freedom
ITERATIVEWEIGHTS = *variate*	To save the iterative weights
LINEARPREDICTOR = *variate*	To save the linear predictor
YADJUSTED = *variate*	To save the adjusted dependent variate
I_2LOGLIKELIHOOD = *variate*	To save the contribution of each unit to the value of minus twice the maximized log likelihood
MAXCYCLE = *scalar*	Maximum number of iterations for see-saw algorithm; default 30
TOLERANCE = *scalar*	Convergence criterion; default 10^{-5}

Parameter

TERMS = *formula*	List of explanatory variates and factors, or model formula

Description

Procedure RCIRCULAR can be used to fit a circular regression model to an angular response. A circular regression model is similar in spirit to a generalized linear model; it employs the von Mises distribution and the arctangent link function. More formally, it is assumed that the angular response follows a von Mises distribution with mean direction μ and concentration parameter κ. The mean direction μ is related to the linear predictor η by means of the link function

$$\mu = \mu_0 + 2 \arctan(\eta)$$

which maps the real line to the circle. The linear predictor η itself is a linear function of all the regressors in the usual way, except that it does not include a constant term. The circular regression model is fitted by means of an iterative algorithm which employs re-weighted least squares to estimate the linear parameters. A detailed account can be found in Fisher (1993) or Fisher & Lee (1992).

Note that the model is not invariant to linear shifts of explanatory variates. This is because the linear predictor η does not contain a constant term. This can be a serious drawback of the

circular regression model. An alternative model without the parameter μ_0 and including an intercept in the linear predictor is not invariant to rotations of the response, which is even worse. Also note that the estimates on page 161 of Fisher (1993) are for the centred distance explanatory variable.

A call to RCIRCULAR must be preceded by a MODEL statement which defines the angular response variate. Only the first response variate is analysed and options other than WEIGHTS should not be set in the MODEL statement. The TERMS parameter of RCIRCULAR specifies the model to be fitted. Cases with a missing response variate or with a zero weight are excluded from the analysis. The FACTORIAL option operates in the usual way. Printed output is controlled by the PRINT option with the usual settings. Setting PRINT=summary displays the value of minus twice the maximized log likelihood, both for the fitted model and for the null model with only the constant μ_0. The difference between the two log likelihood values is also printed with a corresponding probability based on the chi-square distribution using likelihood ratio testing. This tests whether the fitted model is an improvement over the null model. PRINT=monitoring displays monitoring information of the iterative algorithm. The iterative process itself is controlled by the MAXCYCLE option which determines the maximum number of cycles, and by the TOLERANCE option. The iterative process is stopped when the relative difference in minus twice the log likelihood is smaller than the specified tolerance.

Results of the circular regression can be saved by a number of options. The ESTIMATES, SE and VCOVARIANCE options save estimates of the linear parameters, their standard errors and variance-covariance matrix. This never includes the constant parameter. The estimate and standard error of the constant parameter μ_0 can be saved using options MU0 and SEMU0, and those for the concentration parameter κ of the von Mises distribution can be saved using options KAPPA and SEKAPPA. The _2LOGLIKELIHOOD option allows minus twice the maximized log likelihood to be saved, and the DF option saves the residual degrees of freedom. These may be useful for comparing a sequence of nested models fitted by RCIRCULAR using likelihood ratio testing. The RESIDUALS, FITTEDVALUES, LEVERAGES, ITERATIVEWEIGHTS, LINEARPREDICTOR and YADJUSTED options allow you to save the simple residuals, the fitted values (i.e. the fitted mean directions), the leverages of the iterative reweighted least squares algorithm, the linear predictor and an adjusted dependent variate. Finally the contribution of each unit to minus twice the maximized log likelihood can be saved by means of the I_2LOGLIKELIHOOD option.

Options: PRINT, FACTORIAL, RESIDUALS, FITTEDVALUES, LEVERAGES, ESTIMATES, SE, VCOVARIANCE, MU0, SEMU0, KAPPA, SEKAPPA, _2LOGLIKELIHOOD, DF, ITERATIVEWEIGHTS, LINEARPREDICTOR, YADJUSTED, I_2LOGLIKELIHOOD, MAXCYCLE, TOLERANCE.
Parameter: TERMS.

Method

The model is fitted using the algorithm of Fisher & Lee (1993) and Fisher (1993). The iterative fitting of the model is adapted by adding the linear predictor from the previous cycle to the adjusted y variate. For a weighted circular regression the estimated circular standard error of μ_0 is calculated using the sum of the weights instead of the degrees of freedom, see equation (6.64) in Fisher (1993). Note that the estimated standard errors for the linear parameters are conditional on the estimates of μ_0 and κ, and vice versa.

Action with RESTRICT

Only the angular response variate can be restricted. The analysis is restricted accordingly.

References

Fisher, N.I. & Lee, A.J. (1992). Regression models for an angular response. *Biometrics*, **48**, 665-677.

Fisher, N.I. (1993). *Statistical Analysis of Circular Data*. Cambridge University Press, Cambridge.

See also

Procedures: CASSOCIATION, CCOMPARE, CDESCRIBE, DCIRCULAR, WINDROSE.

Genstat Reference Manual 1 Summary section on: Regression analysis.

RCOMPARISONS

Calculates comparison contrasts amongst regression means (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (aov, contrasts); default aov, cont
COMBINATIONS = *string token*	Factor combinations for which to form the predicted means (full, present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when forming the predicted means (marginal, equal, observed); default marg
PSE = *string tokens*	Types of standard errors to be printed with the contrasts (contrasts, differences, lsd); default cont
WEIGHTS = *table*	Weights classified by some or all of the factors in the model; default *
OFFSET = *scalar*	Value of offset on which to base predictions; default mean of offset variate
METHOD = *string token*	Method of forming margin (mean, total); default mean
ALIASING = *string token*	How to deal with aliased parameters (fault, ignore); default faul
BACKTRANSFORM = *string token*	What back-transformation to apply to the values on the linear scale, before calculating the predicted means (link, none); default link
SCOPE = *string token*	Controls whether the variance of predictions is calculated on the basis of forecasting new observations rather than summarizing the data to which the model has been fitted (data, new); default data
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, nonlinear); default *
DISPERSION = *scalar*	Value of dispersion parameter in calculation of s.e.s; default is as set in the MODEL statement
DMETHOD = *string token*	Basis of estimate of dispersion, if not fixed by DISPERSION option (deviance, Pearson); default is as set in the MODEL statement
NBINOMIAL = *scalar*	Supplies the total number of trials to be used for prediction with a binomial distribution (providing a value *n* greater than one allows predictions to be made of the number of "successes" out of *n*, whereas the value one predicts the proportion of successes); default 1
LSDLEVEL = *scalar*	Significance level (%) for least significant differences; default 5
SAVE = *identifier*	Regression save structure for the analysis from which the comparison contrasts are to be calculated

Parameters

FACTOR = *factors*	Factor whose levels are compared
CONTRASTS = *matrices*	Defines the comparisons to be estimated
ORDER = *scalars*	Number of comparisons to estimate; default is the number of rows of the CONTRASTS matrix
GROUPS = *factors* or *pointers*	Set if comparisons are to be made at different

	combinations of another factor or factors
ESTIMATES = *variates* or *pointers*	Saves the estimated contrasts in a variate if GROUPS is unset, or in a pointer to a set of tables
SE = *variates* or *pointers*	Saves standard errors of the contrasts in a variate if GROUPS is unset, or in a pointer to a set of tables
SED = *pointers*	Pointer to a set of symmetric matrices to save standard errors for differences between the contrasts estimated for different levels of the GROUPS factor(s)
LSD = *pointers*	Pointer to a set of symmetric matrices to save least significant differences for the contrasts estimated for different levels of the GROUPS factor(s)
DEVIANCES = *variates*	Saves sums of squares or deviances of the contrasts
DF = *variates*	Saves degrees of freedom for the contrasts

Description

RCOMPARISONS allows you to make comparisons between predicted means from a linear or generalized linear regression. The model should previously have been fitted by the FIT directive in the usual way. The SAVE option can be used to specify the regression save structure from the analysis for which the comparisons are to be calculated (see the SAVE option of the MODEL directive). If SAVE is not specified, the comparisons are calculated from the most recent regression analysis.

The factor amongst whose levels the comparisons are to be calculated is specified by the FACTOR parameter. The CONTRASTS parameter supplies a matrix to specify the comparisons to be calculated. This works in the same way as the matrix supplied as the third parameter of the COMPARISON function, with a column for each level of the FACTOR, and a row for each comparison. You can set the ORDER parameter to a scalar, *n* say, to indicate that only the comparisons in the first *n* rows of the CONTRASTS matrix are to be calculated (otherwise they are all calculated).

By default the comparisons are calculated between the means in the one-way table classified by FACTOR. However, you can set the GROUPS parameter to some other factor to indicate that the comparisons are to be made for each level of that factor, or you can set it to a pointer of factors to make the comparisons for every combination of the levels of those factors.

RCOMPARISONS calculates the means using the PREDICT directive. The first step (A) of the calculation forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data, or COMBINATIONS=full uses all the cells. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data. Alternatively, you can supply your own table of weights, using the WEIGHTS option. There are also options OFFSET, METHOD, ALIASING, BACKTRANSFORM, SCOPE, NOMESSAGE, DISPERSION, DMETHOD and NBINOMIAL to control further aspects of the calculations; these operate exactly as in the PREDICT directive.

The PRINT option controls printed output, with settings:

aov	to print an analysis of variance (for an ordinary linear

regression) or an analysis of deviance (for a generalized linear model), giving the sums of squares (or deviances) and so on for the comparisons;

`contrasts` to print the contrasts.

By default these are both printed. The `PSE` option controls the types of standard errors that are produced to accompany the contrasts, with settings:

`contrasts` for standard errors of the contrasts;

`differences` for standard errors for differences between pairs of contrasts calculated for the different GROUPS;

`lsd` for least significant differences for contrasts calculated for the `GROUPS`.

The default is `contrasts`. The `LSDLEVEL` option sets the significance level (as a percentage) for the least significant differences.

The `ESTIMATES` parameter allows you to save the estimated contrasts. These are in a variate if `GROUPS` is unset, or in a pointer containing a table classified by `GROUPS` for each comparison otherwise. The `SE` parameter saves the standard errors of the contrasts, in a variate or pointer similarly to `ESTIMATES`. If `GROUPS` is set, you can also save standard errors for differences between the contrasts estimated for different levels of the `GROUPS` factor(s). This is again a pointer, with a symmetric matrix for each comparison. Finally, the `DF` parameter can save a variate containing the degrees of freedom of the contrasts, and the `DEVIANCES` parameter can save a variate with their deviances (for a generalized linear model) or sums of squares (for an ordinary linear regression).

Options: `PRINT`, `COMBINATIONS`, `ADJUSTMENT`, `PSE`, `WEIGHTS`, `OFFSET`, `METHOD`, `ALIASING`, `BACKTRANSFORM`, `SCOPE`, `NOMESSAGE`, `DISPERSION`, `DMETHOD`, `NBINOMIAL`, `LSDLEVEL`, `SAVE`.

Parameters: `FACTOR`, `CONTRASTS`, `ORDER`, `GROUPS`, `ESTIMATES`, `SE`, `SED`, `LSD`, `DEVIANCES`, `DF`.

Method

The predicted means and their variances and covariances are calculated using the `PREDICT` directive. The comparisons, their standard errors and sums of squares are then calculated using Genstat's table and matrix calculation facilities.

See also

Directive: `PREDICT`.

Procedures: `FCONTRASTS`, `RTCOMPARISONS`, `VTCOMPARISONS`.

Genstat Reference Manual 1 Summary section on: Regression analysis.

†RCURVECOMMONNONLINEAR

Refits a standard curve with common nonlinear parameters across groups to provide s.e.'s for linear parameters (R.W. Payne).

Options

PRINT = *string tokens*	Printed output from the analysis (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring); default mode, summ, esti
MAXCYCLE = *variate*	Maximum number of iterations; default 30
METHOD = *string token*	Algorithm for fitting nonlinear model (gaussnewton, newtonraphson, fletcherpowell); default newt
STEPLENGTHS = *scalar* or *variate*	Initial step lengths for the parameters
SAVE = *regression save structure*	Save structure from this analysis
INSAVE = *regression save structure*	Save structure for the curve fitted by FITCURVE, default takes the most recent regression analysis

No parameters

Description

RCURVECOMMONNONLINEAR can be used after a FITCURVE analysis to refit a standard curve that has common nonlinear parameters across groups. It uses the CALCULATION option of FIT, which provides standard errors for the linear parameters. These are unavailable with FITCURVE.

The INSAVE option can provide the regression save structure from the FITCURVE analysis. If this is not set, the save structure from the most recent regression analysis is used. A fault is given if the save structure is not from a FITCURVE analysis with groups and common nonlinear parameters. The SAVE option saves the regression save structure from this analysis.

The PRINT option controls printed output, with the same settings as FIT. The other options control aspects of the optimization. MAXCYCLE specifies the maximum number of iterations to be used to estimate the nonlinear parameters; default 30. METHOD specifies the algorithm to be used. The default is Newton Raphson, which is the same method as FITCURVE. STEPLENGTHS defines step lengths for the estimation of the nonlinear parameters. FITCURVE uses a different strategy from FIT. It includes nonlinear parameters for all the groups in the model, but constrains them to be equal when they are common across groups. Consequently RCURVECOMMONNONLINEAR may obtain slightly different parameter estimates from the original FITCURVE analysis. Modifying these options may enable you to obtain closer results.

Options: PRINT, MAXCYCLE, METHOD, STEPLENGTHS, SAVE, INSAVE.
Parameters: none.

Action with RESTRICT

Any restriction applied to vectors used in the regression model applies also to the results from RCURVECOMMONNONLINEAR.

See also

Directives: FIT, FITCURVE.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RDA

Performs redundancy analysis (A.I. Glaser).

Options

PRINT = *string tokens*	What to print (variance, loadings, roots, evalues, evectors, speciesscores, sitescores, fitsitescores, correlations, fitcorrelations, weights); default vari, root
NROOTS = *scalar*	Number of eigenvalues and eigenvectors to include in output; default * takes all the non-zero eigenvalues
NORMALIZE = *string tokens*	Whether to normalize the Y, X and/or Z variates to have unit sums-of-squares before the analysis (x, y, z); default x, z
SCALING = *string token*	Scaling for species and site scores (none, both); default none
TOLERANCE = *scalar*	Tolerance for detecting non-zero eigenvalues; default 10^{-5}

Parameters

Y = *pointers*	Each pointer defines a set of response variates to be modelled
X = *pointers*	Explanatory variates or factors to use for each pointer of y-variates
Z = *pointers*	Conditioning variates or factors to remove ("partial out") before the analysis
LRV = *LRVs*	LRV structure from each analysis, storing the eigenvectors, eigenvalues and total variance
SPECIESSCORES = *matrices*	Saves the "species scores" from each analysis
SITESCORES = *matrices*	Save the "site scores" from each analysis
FITSITESCORES = *matrices*	Save the fitted "site scores" from each analysis
CORRELATIONS = *matrices*	Saves the correlations between the site scores and the x-variates
FITCORRELATIONS = *matrices*	Saves the correlations between the fitted site scores and the x-variates
WEIGHTS = *matrices*	Save the weights of the x-variates in the formation of the site scores
SAVE = *pointers*	Save structure which provides information for use in CRBIPLOT and CRTRIPLOT

Description

Redundancy analysis is the direct extension of multiple regression to the modelling of multivariate response data (see e.g. Legendre & Legendre 1998). The response data are a set of y-variates, specified in a pointer using the Y parameter. The explanatory variables, which may be either variates or factors, are specified in a pointer by the X parameter. Similarly, the Z parameter can be used to specify conditioning variables, which again may be either variates or factors; this gives partial RDA, in which the effect of the z-variables is removed before performing RDA. This may be useful in cases where the effects of the elements of Z on Y are well known, or we may wish to isolate the effect of an individual explanatory variable (in which case we would place all but one of the explanatory variables in Z). When all elements of a variable are equal to zero, CCA removes the variable.

The PRINT option controls printed output, with settings:

`roots`	the eigenvalues of the fitted values;
`evalues`	synonym of roots;
`loadings`	the eigenvectors associated with each eigenvalue, also known as the "species scores";
`evectors`	synonym of loadings;
`speciesscores`	the "species scores" from the analysis (synonym of `loadings` and `evectors`);
`variance`	the fraction of the variance of the y-variates associated with each eigenvalue;
`sitescores`	the "site scores" of the y-variates (i.e. the ordination of the units in the y-variate space);
`fitsitescores`	the fitted "site scores" of the fitted values of the y-variates (i.e. the ordination of the units in the y-variate space);
`correlations`	the correlation between the site scores and the x-variables;
`fitcorrelations`	the correlation between the fitted site scores and the x-variables;
`weights`	the weights of the x-variables in the formation of the site scores.

By default `PRINT=roots,variance`. The `LRV`, `SPECIESSCORES`, `SITESCORES`, `FITSITESCORES`, `CORRELATIONS`, `FITCORRELATIONS` and `WEIGHTS` parameters allow this information to be saved.

The `NROOTS` option specifies the number of eigenvalues and eigenvectors to include in the output. By default all the non-zero eigenvalues are included. The `NORMALIZE` option controls whether to normalize the `Y` variates, or `X` or `Z` variables to have unit sums-of-squares before the analysis. The default is to normalize the x- and z-variables but not the y-variates. (Note: this normalization of the x's and z's does not affect the variances accounted for in the y-variates.) The `SCALING` option controls scaling for species and site scores. If `both` is selected, both species and site scores are multiplied by the square root of their corresponding eigenvalues. For RDA choosing `none` is equivalent to Scaling type 1 in Legendre & Legendre (1998), whilst `both` is equivalent to Scaling type 2 in the same book. The `TOLERANCE` option specifies a threshold for the detection of non-zero eigenvalues (default 10^{-5}). An eigenvalue is taken to be non zero if is it greater than `TOLERANCE` multiplied by the total variance.

The `SAVE` parameter allows you to save a pointer containing full details of the analysis. This can then be used to generate plots using the `CRBIPLOT` or `CRTRIPLOT` procedures. The most recent save structure is kept automatically inside Genstat to use as a default for the `SAVE` options of `CRBIPLOT` and `CRTRIPLOT`. So, you need save the pointer explicitly only if you want to display output from more than one analysis at a time.

Options: `PRINT`, `NROOTS`, `NORMALIZE`, `SCALING`, `TOLERANCE`.
Parameters: `Y`, `X`, `Z`, `LRV`, `SPECIESSCORES`, `SITESCORES`, `FITSITESCORES`, `CORRELATIONS`, `FITCORRELATIONS`, `WEIGHTS`, `SAVE`.

Method
`RDA` and partial RDA are explained in Sections 11.1 and 11.3 of Legendre & Legendre (1998).

Action with `RESTRICT`
If any of the variate or factors in the `Y`, `X` or `Z` pointers are restricted, only the defined subset of the units will be used in the analysis.

Reference
Legendre, P. & Legendre, L. (1998). *Numerical Ecology, Second English Edition*. Elsevier,

Amsterdam.

See also
Procedures: CRBIPLOT, CRTRIPLOT, CANCORRELATION, CCA, PLS.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

RDESTIMATES

Plots one- or two-way tables of regression estimates (R.W. Payne).

Options

GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
METHOD = *string token*	What to plot (estimates, lines); default esti
XFREPRESENTATION = *string token*	How to label the x-axis (levels, labels); default labels uses the XFACTOR labels, if available
PSE = *string token*	What s.e. to plot to represent variation (average, individual); default aver
SAVE = *regression save structure*	Save structure of the analysis to display; default * shows the most recently fitted regression

Parameters

XFACTOR = *factors*	Factor providing the x-values for each plot
GROUPS = *factors*	Factor identifying the different sets of points from a two-way table of estimates
XVARIATES = *variates*	X-variates for regression coefficients or pointer
NEWXLEVELS = *variates*	Values to be used for XFACTOR instead of its existing levels
TITLE = *texts*	Title for the graph; default defines a title automatically
YTITLE = *texts*	Title for the y-axis; default ''
XTITLE = *texts*	Title for the x-axis; default is to use the identifier of the XFACTOR

Description

RDESTIMATES plots tables of estimates from a regression analysis. By default the estimates are from the most recent regression, but you use the SAVE option to specify the save structure (from a MODEL statement) from some other analysis.

The XFACTOR parameter indicates the factor against whose levels the estimates are plotted. You can also specify a second factor, using the GROUPS parameter, to plot a two-way table of estimates. A separate set of points is then plotted for every level of GROUPS.

By default, the estimates will be for the model term XFACTOR (if GROUPS is not set) or XFACTOR.GROUPS (if GROUPS is set). You can also specify one, or more, variates for the term, using the XVARIATES parameter. If XVARIATES is set to a single variate, xvar say, the term will be XFACTOR.xvar or XFACTOR.GROUPS.xvar (representing regression coefficients for xvar). Alternatively, it can be set to a pointer containing several variates, for example x1var and x2var. The term will be then be XFACTOR.x1var.x2var or XFACTOR.GROUPS.x1var.x2var (representing regression coefficients for the product of the variates x1var and x2var).

The NEWXLEVELS parameter enables different levels to be supplied for XFACTOR if the existing levels are unsuitable. If XFACTOR has labels, these are used to label the x-axis unless you set option XFREPRESENTATION=levels.

Usually, each estimate is represented by a point (using pens 1, 2, and so on for each level in turn of the GROUPS factor). However, with high-resolution plots, the METHOD option can be set to lines to draw lines between the points. The GRAPHICS option controls whether a high-resolution or a line-printer graph is plotted; by default GRAPHICS=high.

The PSE option specifies how to represent the variability of the estimates, as follows:

average	plots an error bar showing the average standard error of the

	estimates;
`individual`	plots a bar around each estimate showing plus and minus its standard error.

The `TITLE`, `YTITLE` and `XTITLE` parameters allow you to supply titles for the graph, the y-axis and the x-axis respectively.

Options: `GRAPHICS`, `METHOD`, `XFREPRESENTATION`, `PSE`, `SAVE`.
Parameters: `XFACTOR`, `GROUPS`, `XVARIATES`, `NEWXLEVELS`, `TITLE`, `YTITLE`, `XTITLE`.

Method
`RDESTIMATES` uses the `GET` directive, if necessary, to obtain the regression save structure, and `RKESTIMATES` to obtain the tables of estimates.

See also
Procedures: `RCHECK`, `RGRAPH`, `AGRAPH`, `DTABLE`, `VDEFFECTS`, `VGRAPH`.
Genstat Reference Manual 1 Summary section on: Regression analysis.

REPPERIODOGRAM

Gives periodogram-based analyses for replicated time series (R.P. Littlejohn).

Options

PRINT = *string token*	What to print (pair, randomization, glm); default * i.e. none
PLOT = *string token*	What graphs to plot (group, mean, logmean, cumulative, cv, pair); default mean, logm
TITLE = *text*	Title for each page of graphs
REPRESENTATION = *string token*	Form of data in SERIES (timeseries, meanperiodogram); default time
LENGTH = *scalar* or *variate*	Scalar specifying that the first N units of the series are to be used, or a variate specifying the first and last units of the series to be used
SEED = *scalar*	Seed for randomization; default 0
NRANDOMIZATIONS = *scalar*	Number of randomizations; default 99
TREATMENTS = *factor*	Contains ordered classification of SERIES
PAIR = *variates*	Treatment pair levels for pairwise comparisons
COLOUR = *text* or *variate*	Colours for each level of TREATMENTS; default * sets suitable colours automatically
MEANPERIODOGRAM = *pointer*	Saves mean periodograms according if REPRESENTATION=timeseries
REPLICATION = *scalar* or *variate*	Inputs or saves number of replicate series if REPRESENTATION=timeseries; scalar can be used for equal replication

Parameter

SERIES = *variates*	Specify the time series to be analysed

Description

REPPERIODOGRAM gives periodogram-based analyses of replicated time series. The data are supplied in a list of variates using the SERIES parameter, either as the original time series (option REPRESENTATION set to timeseries) with the level for each series given by the factor specified by the TREATMENTS option, or as the mean periodograms for each treatment level (option REPRESENTATION set to meanperiodogram), with levels and labels optionally given by the TREATMENTS factor and the multiplicity of each treatment defined by the REPLICATION option. In the former case the LENGTH option can specify that only part of each series is to be used, using either a scalar N to indicate that the first N values are to be used, or a variate of length two, holding the values of the first and last units of the required subseries. This may be used to eliminate missing values, which are otherwise not permitted. Further, when REPRESENTATION=timeseries, periodogram means and the replication variate can be saved using the MEANPERIODOGRAM and REPLICATION options, respectively.

Graphical output is controlled by the PLOT option. For the group (REPRESENTATION=timeseries only), mean, logmean and cumulative periodogram and cv graphs, the COLOUR option can be used to code for treatments; by default, the standard colours are used in the same order as for pens 2, 3... (see PEN). The cv plot (REPRESENTATION=timeseries only) gives a scatterplot of coefficients of variation for each treatment group at each frequency, together with lines for the means of these cvs at each frequency for those treatments with replication greater than one, and cv=1, the theoretical value if there is no subject-specific variation. For these graphs a title can be supplied using the TITLE option. Graphs are also given for the differences between pairs of log periodograms as defined

by PAIR (see below), with 95% confidence intervals on the sample and null (equal periodograms) distribution.

Output of various test statistics for pairwise comparison of treatment levels described by Diggle (1990) and Diggle & Fisher (1991) is controlled by the PRINT and PAIR options. PAIR is a list of 2-unit variates representing treatment levels, e.g.

```
PAIR=!(1,2),!(3,4)
```

gives tests comparing treatment levels 1 and 2, followed by tests for levels 3 and 4. With PRINT=pair, the maximum absolute value and range of the difference of log periodograms give (weak) tests against the null hypotheses of equal and proportional spectra, respectively. With PRINT=random, a randomization test is given for the equality of cumulative spectra, which is insensitive to the alternative of proportional spectra. The seed for the ramdomizations can be set using the SEED option, and the number of randomizations is specified by NRANDOMIZATIONS (default 99). This is available only if the treatments in the pair have equal replication.

When PRINT=glm, a generalized linear model is fitted to the mean periodograms for all treatments, adjusting for frequency, and testing for differences with treatment in constant (proportional spectra), linear (power shift) and quadratic (power spread) contrasts with frequency (Diggle 1990). Results are presented in the accumulated analysis of deviance table and tables of parameter estimates, within which the Intercept-Difference, Slope-Difference and Curve-Difference estimates relate to the above hyptheses.

Options: PRINT, PLOT, REPRESENTATION, LENGTH, TREATMENTS, PAIR, SEED, NRANDOMIZATIONS, COLOUR, TITLE, MEANPERIODOGRAM, REPLICATION.
Parameter: SERIES.

Method

The series are mean-corrected, but not trend corrected, before transformation, and are not smoothed. Critical values for the Range test are obtained from tables in Potscher & Reschenhofer (1988) and Coates & Diggle (1986). Random numbers are generated using URAND. The analysis for PRINT=glm is obtained from fitting a generalized linear model with DISTRIBUTION=gamma, LINK=log and DISPERSION=1/*nr*, where *nr* is number of replicates of the treatments.

Action with RESTRICT

The SERIES may not be restricted; restriction of the input series to a contiguous set of units may be achieved by use of the LENGTH parameter.

References

Coates, D.S. & Diggle, P.J. (1986) Tests for comparing two estimated spectral densities. *Journal of Time Series Analysis*, **7**, 7-20.

Diggle, P.J. (1990). *Time Series: A Biostatistical Introduction*. Oxford, Clarendon Press.

Diggle, P.J. & Fisher, N.I. (1991). Nonparametric comparisons of cumulative periodograms. *Applied Statistics*, **40**, 423-434.

Potscher, B.M. & Reschenhofer, E. (1988). Discriminating between two spectral densities in case of replicated observations. *Journal of Time Series Analysis*, **9**, 221-224.

See also

Directive: FOURIER.
Procedures: DFOURIER, MCROSSPECTRUM, PERIODTEST, SMOOTHSPECTRUM.
Genstat Reference Manual 1 Summary section on: Time series.

†RESHAPE

Reshapes a data set with classifying factors for rows and columns, into a reorganized data set with new identifying factors (D.B. Baird).

Options

PRINT = *string token*	What to print (results); default *, i.e. none
ROWCLASSIFICATION = *factors*, *texts*, *variates* or *pointer*	Factors classifying the rows in the data; default a factor called Rows with a level for each row
COLCLASSIFICATION = *factors*, *texts*, *variates* or *pointer*	Factors or texts classifying the columns in the data; default a factor called Columns with labels formed from the column identifiers in DATA
MEANFACTORS = *factors*, *texts*, *variates* or *pointer*	Row or column factors whose groups are averaged in the output data set
TOTALFACTORS = *factors*, *texts*, *variates* or *pointer*	Row or column factors whose groups are totalled in the output data set
FIRSTSUMMARY = *string token*	Which summaries to form first (means, totals) default means
NEWROWFACTORS = *factors*	Factors to index the new rows
NEWCOLUMNFACTORS = *factors*, *texts* or *variates*	Factors to indexing the columns in the new data set
REDEFINE = *string token*	Whether to redefine the NEWROWFACTORS factors and DATA columns, if NEWROWFACTORS or NEWDATA are not set or use names used in the input data (yes, no); default no
MVINCLUDE = *string token*	Whether to include factor combinations with no observations in the output data set (*,rows, columns); default *; i.e. remove missing rows and columns

Parameters

DATA = *pointers*	Pointer containing data to be reshaped
NEWDATA = *pointers*	Pointer containing the reshaped data columns

Description

RESHAPE reshapes data matrices. This is useful when rows and columns in a data set are to be swapped to another dimension (rows to columns or vice-versa). It combines the functionality of STACK and UNSTACK in a single procedure. The data columns are stacked into a single columns, with factors indexing the resulting rows created from the original row and column factors. It is then unstacked by one or more of these factors to reshape the data. RESHAPE goes beyond STACK, in that more than one column factor can be defined. For example, if you have variates containing different measurements taken at several different times, one column factor could index the measurements and another the times. Also, the data can be collapsed across some of the factors by taking totals or means.

The ROWCLASSIFICATION and COLCLASSIFICATION options classify the rows and columns, respectively, and together these provide the input factors. If texts or variates are specified with these options, they are converted internally to factors. If any of these options is not specified, a default factor (Rows or Columns) is created, and this can be used in the NEWCOLUMNFACTORS, TOTALFACTORS or MEANFACTORS options, described below. However, the Columns factor is formed only if there is more than one vector in the pointer specified by the DATA parameter. This

pointer specifies the variates, text or factors to be reorganized. The vectors in DATA must be compatible with the first vector, and this determines the type of the resulting column. Any type of vector can be combined with a factor or a text, but texts cannot be combined with a variate. If the COLCLASSIFICATION factors have no values, a warning is given, and their values are formed using GENERATE. A fault is given if the product of the numbers of levels is not equal to the number of vectors in DATA.

The TOTALFACTORS and MEANFACTORS options specify factors over which totals or means, respectively, are to be formed. The FIRSTSUMMARY option controls whether the means or the totals are formed first. Suppose, for example, we have a 3 × 2 classification of (3,5,6,3,7,*) where the final cell is missing. Totalling (3,5,6) and (3,7,*) and then averaging would give (14 + 10)/2 = 12. However, averaging (3,3), (5,7) and (6,*) and then totalling would give (3+6+6) = 15. If the missing value is replaced by 8, both orders of operation would give the same result, as either (14+18)/2 or (3+6+7) i.e. 16.

The NEWDATA parameter saves a pointer containing the reshaped data columns. If the NEWDATA pointer is undefined, it will be created with labels formed from the input factors labels (if present) or levels. The NEWCOLUMNFACTORS option lists factors to index the columns in the new data pointer. A column in the output data is created for each combination of these factors (but some of these columns may be dropped according to the setting of the MVINCLUDE option as explained below). The NEWROWFACTORS option specifies new row factors, which are formed from the input factors that have not been used in the NEWCOLUMNFACTORS, MEANS or TOTALS options. The factors specified by NEWROWFACTORS are allocated to the original factors in the order in which they were specified, first by the ROWCLASSIFICATION option, and then by the COLCLASSIFICATION option. These input factors can be reused if you set option REDEFINE = yes.

The MVINCLUDE option controls whether empty rows or columns are included in the new data set. The default setting, *, removes both empty rows and columns, rows includes empty rows, columns includes empty columns, and rows,columns includes both empty rows and columns.

You can set option PRINT=results to print the new data set. By default, nothing is printed.

Options: PRINT, ROWCLASSIFICATION, COLCLASSIFICATION, TOTALFACTORS, MEANFACTORS, FIRSTSUMMARY, NEWROWFACTORS, NEWCOLUMNFACTORS, REDEFINE, MVINCLUDE.
Parameters: DATA, NEWDATA.

Method

RESHAPE uses APPEND to form the data set into a single column indexed by all the input factors. TABULATE is used to form means and totals, and then VTABLE is used to extract the data from the summary tables. UNSTACK is used to extract the reshaped columns. SUBSET is used to remove missing rows.

Action with RESTRICT

RESHAPE ignores any restrictions on the input factors or the DATA structures.

See also

Directives: EQUATE, TABULATE.
Procedures: APPEND, STACK, SUBSET, UNSTACK, VSUMMARY, VTABLE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation.

†RFFAMOUNT

Fits harmonic models to mean rainfall amounts for a Markov model (J.O. Ong'ala & D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output for each fitted model (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, confidence); default mode, summ, esti, accu
PLOT = *string token*	What plots to display (results); default resu
NHARMONICS = *scalar*	Defines the number of harmonics to fit (1...4); default 2
SPREADSHEET = *string tokens*	What to save in a spreadsheet (results); default *

Parameters

COUNTS = *table*	Supplies the table of counts by Markov class and day number within the year (1...366)
AMOUNTS = *tables*	Supplies the table of mean rainfall by wet Markov class and day
WINDOW = *scalars*	Window for the graph; default 3 for a single class and 1 otherwise
TITLE = *texts*	Title for the graph; default forms an automatic description
RESULTS = *pointers*	Saves a pointer to the variates of fitted rainfall means by day for each wet class
OUTFILE = *texts*	File(with extension .gwb, or .xlsx) to save the spreadsheet of results

Description

RFFAMOUNT fits harmonic (Fourier) models with a period of 366 days to rainfall summaries produced by RFSUMMARY. The Markov model fitted by RFSUMMARY splits the days into different classes based on the history of the preceding days. The daily states, order and type of the Markov model can be formed by RFSUMMARY, but only models with two states are handled. The harmonic model is a linear combination of sine and cosine terms with periods of 366/*n*.. The number of harmonic terms (*n*) is specified by the NHARMONICS option and can be 1, 2, 3 or 4.

The COUNTS and AMOUNTS parameters give the table of rainfall counts and mean amounts for each Markov state by day within the year (1...366). The RESULTS parameter can save variates of fitted amounts for the wet (e.g. ww and wd) Markov classes for each day within a year.

Printed output of the summaries is controlled by the PRINT option, with the same settings as the FIT directive. The fitted amounts can be displayed in a spreadsheet using by setting option SPREADSHEET =results. This creates a sheet containing the variates giving the fitted amounts of rainfall for each day in the year by the wet Markov classes. The spreadsheet can be saved to a file by setting the OUTFILE parameter to a Genstat or Excel spreadsheet filename (.gwb or .xlsx).

You can set option PLOT=results to plot the fitted amounts. The TITLE parameter can supply a title for the graph; if this not set, a descriptive title will be created from the Markov-chain options. The WINDOW parameter specifies the window to use for the graph.

Options: PRINT, PLOT, NHARMONICS, SPREADSHEET.
Parameters: COUNTS, AMOUNTS, WINDOW, TITLE, RESULTS, OUTFILE.

Method
The procedure calculates sine and cosine terms for the number of harmonics and fits a gamma generalized linear model to the rainfall means weighted by the counts of the number of wet days.

Reference
Ong'ala, J.O. (2011). Simplifying the Markov chain analysis of rainfall data using Genstat. *MSc Thesis*, Maseno University.

See also
Directive: FIT.
Procedures: RFFPROBABILITY, RFSUMMARY.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

†RFFPROBABILITY

Fits harmonic models to rainfall probabilities for a Markov model (J.O. Ong'ala & D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output for each fitted model (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, confidence); default mode, summ, esti, accu
PLOT = *string token*	What plots to display (results); default resu
NHARMONICS = *scalar*	Defines the number of harmonics to fit (1...4); default 2
SPREADSHEET = *string tokens*	What to save in a spreadsheet (results); default *

Parameters

COUNTS = *table*	Supplies the table of counts by Markov class and day within the year (1...366)
WINDOW = *scalars*	Window to plot the graph; default 3 for a single class and 1 otherwise
TITLE = *texts*	The title for the plot; default forms an automatic description
RESULTS = *pointers*	Saves a pointer to variates of fitted rainfall probabilities by day for each wet state
OUTFILE = *texts*	File (with extension .gwb, or .xlsx) to save the selected spreadsheet components

Description

RFFPROBABILITY fits harmonic (Fourier) models with a period of 366 days to rainfall counts produced by RFSUMMARY. The Markov model fitted by RFSUMMARY splits the days into different classes based on the history of the preceding days. The daily states, order and type of the Markov model can be formed by RFSUMMARY. The harmonic model is a linear combination of sine and cosine terms with periods of 366/*n*.. The number of harmonic terms (*n*) is specified by the NHARMONICS option, and can be 1, 2, 3 or 4.

The COUNTS parameter supplies the table of counts for each Markov class by day within the year (1...366). The RESULTS parameter can save fitted probabilities by wet class for each day.

Printed output of the summaries is controlled by the PRINT option, with the same settings as the FIT directive. The probabilities can be displayed in a spreadsheet by setting option SPREADSHEET=results. This creates a sheet containing variates giving the fitted probabilities for each day in the year by the wet Markov classes. The spreadsheet can be saved to a file by setting the OUTFILE parameter to a Genstat or Excel spreadsheet filename (.gwb or .xlsx).

You can set option PLOT=results to plot the fitted probabilities. The TITLE parameter can supply a title for the graph; if this not set, a descriptive title will be created from the Markov-chain options. The WINDOW parameter specifies the window to use for the graph.

Options: PRINT, PLOT, NHARMONICS, SPREADSHEET.
Parameters: COUNTS, WINDOW, TITLE, RESULTS, OUTFILE.

Method

The procedure calculates sine and cosine terms for the number of harmonics and fits a binomial generalized linear model to the counts of wet days vs dry days for each history from the preceding days.

Reference

Ong'ala, J.O. (2011). Simplifying the Markov chain analysis of rainfall data using Genstat. *MSc Thesis*, Maseno University.

See also

Directive: FIT.
Procedures: RFFAMOUNT, RFSUMMARY.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

RFINLAYWILKINSON

Performs Finlay and Wilkinson's joint regression analysis of genotype-by-environment data (P.W. Lane & K. Ryder).

Options

PRINT = *string tokens*	What to print (model, summary, estimates, sortedsensitivities, monitoring); default mode, summ, esti, sort
PLOT = *string tokens*	What graphs to plot (lines, trellislines, sensitivities); default *
NBEST = *scalar*	Number of best genotypes to print in table of sorted sensitivities; default * i.e. print all of them
DIRECTION = *string token*	Direction to sort table of sorted sensitivities (ascending, descending); default asce
TOLERANCE = *scalar*	Convergence criterion; default 0.001
MAXCYCLE = *scalar*	Maximum number of cycles; default 15
SAVE = *regression save structure*	Save structure from MODEL statement defining the model; default is to use the structure from the latest MODEL statement

Parameters

GENOTYPES = *factors*	The genotype factor; no default
ENVIRONMENTS = *factors*	The environment factor; no default
SENSITIVITIES = *tables*	Saves the estimates of sensitivities; default *
GENMEANS = *tables*	Saves the estimates of genotype means; default *
ENVMEANS = *tables*	Saves the estimates of environment means; default *
ENVEFFECTS = *tables*	Saves the estimates of environment effects; default *
SESENSITIVITIES = *tables*	Saves the s.e.s of sensitivities; default *
SEGENMEANS = *tables*	Saves the s.e.s of genotype means; default *
SEENVEFFECTS = *tables*	Saves the s.e.s of environment effects; default *
MSDEVIATIONS = *tables*	Saves the mean square deviations about the line fitted to each genotype; default *
DEVIANCE = *scalar*	Saves the residual deviance
DF = *scalar*	Saves the residual d.f
TITLE = *text*	Overall title for the graphs
YTITLE = *text*	Y-axis title for the graph of the lines
XTITLE = *text*	X-axis title for the graph of the lines
EXIT = *scalar*	Exit status: set to 0 if the analysis converged, 1 otherwise

Description

Procedure RFINLAYWILKINSON performs the analysis proposed by Finlay & Wilkinson (1963) and Yates & Cochran (1938) to investigate the interaction between two factors. It is an update of the procedure RJOINT, with syntax and output conventions revised for compatibility with the new QTL procedures. RJOINT, however, is retained to allow existing programs to continue to run.

The analysis is motivated by the study of genotype-by-environment interactions in agriculture. The two factors are then genotypes of a particular crop and environments in which some experiments have been carried out. The factors are specified using the parameters GENOTYPES and ENVIRONMENTS.

The environments may be different sites within the same year, different years for the same

site, or a combination of the two with little interest in individual year and site contributions. The intention is to characterize the *sensitivity* of each genotype to environmental effects by fitting a regression of the environment means for each genotype on the average environment means. Sensitivity provides a way of assessing the *stability* of the genotypes. The responses of genotypes with low sensitivity values are more stable with respect to changes of environment. Eberhart & Russell (1966) suggested that it is also interesting to consider the means of the squared deviations of the observations about the line fitted for each genotype. The genotypes with smaller mean square deviations are giving more predictable responses.

The model to fit is nonlinear, with the form

$$y_{ij} = g_i + b_i \times e_j + \text{error}$$

where g_i are genotype means, e_j are environment effects (with $\sum e_j=0$) and b_i are the sensitivity parameters (with mean(b_i)=1). Usually, the aim is to find genotypes with large means and small sensitivities, to ensure a reliable crop under variable conditions.

The data may consist of one value of the response (e.g. yield) for each combination of genotype and environment. More often, however, the data are incomplete because not all genotypes are tested at each environment. Also, there may be multiple measurements of genotypes at some environments. If the response is a count or a proportion, as for example when investigating disease resistance, it will be more appropriate to use a generalized linear model based on a Poisson or binomial distribution and a log or logit link function.

The model and response variate must be specified by giving a `MODEL` statement before calling `RFINLAYWILKINSON`. For example,

```
MODEL yield
```

You can choose to fit any generalized linear model by setting the `DISTRIBUTION` and `LINK` options of `MODEL`: thus, to model proportions, you could give a statement like

```
MODEL [DISTRIBUTION=binomial; LINK=logit; DISPERSION=*] \
      prop; NBINOMIAL=100
```

The iterative process used in the procedure is controlled by the options `TOLERANCE` and `MAXCYCLE`. At each iteration, the maximum difference between estimates of the sensitivity parameters in successive iterations is compared to the tolerance: the process ends when the differences are small enough, or when the maximum number of iterations is reached. The progress of the search can be followed by including the `monitoring` setting of the `PRINT` option, and the `EXIT` parameter can save a scalar with the value zero if the analysis converged and one otherwise.

Output is controlled by the `PRINT` option. The `model` setting prints a description of the model. The `summary` setting displays an analysis of variance (or deviance for non-Normal distributions) showing the effects of Varieties, Environments and Sensitivities (i.e. the effect of allowing different sensitivities for each genotype). The `estimates` setting displays two tables. The first table is classified by genotypes and contains the unadjusted means, estimated means (on the scale of the link function, if relevant), standard errors of the estimated means, back-transformed means (if relevant), sensitivities, standard errors of sensitivities, mean square deviations and the ranks of the genotypes according to their sensitivities. (The genotype with rank 1 is the one that is least sensitive.) The second table is classified by environments, and contains estimates of effects (on the scale of the link function, if relevant), standard errors of estimates, means (formed from the effects and the mean of the genotype means), back-transformed means (if relevant), the ranks of the environments according to their means. The `sortedsensitivities` setting displays a table classified by genotypes, containing sensitivities and estimated means with their standard errors, mean square deviations and back-transformed means (if relevant). The rows of the table are sorted into either ascending or descending order of sensitivities, according to the setting of the `DIRECTION` option (default `descending`). The `NBEST` option can be set to control the number of genotypes that are included; by default they are all printed. The `monitoring` setting produces monitoring information during the fit.

The PLOT option controls the graphs that are plotted, with settings:

lines	plots the fitted lines, all on the same graph,
trellislines	plots the fitted lines in a trellis plot, classified by genotypes, and
sensitivities	produces a scatter-plot matrix displaying the sensitivities, the mean square deviations and the estimated means.

The TITLE parameter defines the overall title for plots of the fitted lines; the default is "Finlay & Wilkinson analysis". The YTITLE and XTITLE parameters define titles for the y- and x-axes, respectively for the plots of the fitted lines; the default for the y-axis is the name of the y-variate, and the default for the x-axis is the name of the ENVIRONMENTS factor.

The remaining parameters allow the results from the analysis to be saved: sensitivities, genotype means, environment effects, environment means, and standard errors of sensitivities, genotype means, environment effects, mean square deviations, residual deviance and degrees of freedom. After calling the procedure, you can use the RKEEP directive to access fitted values and residuals. Other results from the fit, that can be accessed via RKEEP or RDISPLAY, may not be correct: for example, the number of residual d.f. shown by

```
RDISPLAY [PRINT=summary]
```

does not allow for the estimation of sensitivities.

Options: PRINT, PLOT, NBEST, DIRECTION, TOLERANCE, MAXCYCLE, SAVE.

Parameters: ENVIRONMENTS, GENOTYPES, SENSITIVITIES, GENMEANS, ENVMEANS, ENVEFFECTS, SESENSITIVITIES, SEGENMEANS, SEENVEFFECTS, MSDEVIATIONS, DEVIANCE, DF, TITLE, YTITLE, XTITLE, EXIT.

Method

The procedure uses iterative scheme (A) referred to in Digby (1979). The scheme has been generalized to deal with alternative distributions and link functions. First the environment effects are estimated with the sensitivity parameters set to 1, and then the procedure alternates between estimating the sensitivities with given environment effects and estimating environment effects with given sensitivities. Convergence is tested by comparing the maximum difference between old and new sensitivities against the criterion (default 0.001), but the maximum number of cycles (default 15) will not be exceeded. If the MAXCYCLE option is set to 1, the result is an unmodified joint regression analysis; see Finlay & Wilkinson (1963).

Action with RESTRICT

A restriction applied to the response variate will be taken into account. Residuals and fitted values will be formed only for the restricted subset of values. If levels of the factors are not represented in the restricted subset, then no results will be shown for those genotypes and/or environments. Do not restrict the environment or genotype factor differently to the response variate: results may then be incorrect.

References

Digby, P.G.N. (1979). Modified joint regression analysis for incomplete variety × environment data. *Journal of Agricultural Science, Cambridge*, **93**, 81-86.

Eberhart, S.A. & Russell, W.A. (1966). Stability Parameters for Comparing Varieties. *Crop Science*, **6**, 36-40.

Finlay, K.W. & Wilkinson, G.N. (1963). The analysis of adaptation in a plant-breeding programme. *Australian Journal of Agricultural Research*, **14**, 742-754.

Yates, F. & Cochran, W.G. (1938). The analysis of groups of experiments. *Journal of Agricultural Science, Cambridge*, **28**,556-580.

See also

Procedures: AMMI, GESTABILITY, GGEBIPLOT, RJOINT.

Genstat Reference Manual 1 Summary sections on: Regression analysis, REML analysis of linear mixed models.

†RFSUMMARY

Forms summaries for a Markov model from rainfall data (J.O. Ong'ala & D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output (counts, amounts, probabilities); default *
PLOT = *string token*	What plots to display (probabilities); default prob
DAY = *variate* or *factor*	Day as a date or a day number within the year
LIMITS = *scalar* or *variate*	Values to define the daily rainfall states; default 0.85
ORDER = *scalar*	Defines the order of the Markov chain (0...5); default 1
HIGHORDER = *scalar*	Whether to use a high-order Markov chain; (no, yes); default no
INITIAL = *scalar* or *variate*	The amounts of rainfall prior to the first day; default *
SPREADSHEET = *string tokens*	What to save in a spreadsheet (counts, amounts, probabilities); default *

Parameters

DATA = *variates*	The daily rainfall amounts
WINDOW = *scalars*	Window to plot the graph; default 3 for ORDER=0 and 1 otherwise
TITLE = *texts*	The title for the plot; default uses an automatic description
COUNTS = *tables*	Saves the counts by Markov state and day
AMOUNTS = *tables*	Saves the mean rainfall by Markov wet states and day
PROBABILITIES = *pointers*	Saves a pointer to variates of probabilities of a wet day by class
CATEGORIES = *factors*	Saves the Markov class for each day
STATECOUNTS = *pointers*	Saves a pointer to tables of counts for each state
OUTFILE = *texts*	File (with extension .gwb, or .xlsx) to save selected spreadsheet components

Description

RFSUMMARY creates summaries from rainfall data for a Markov chain model analysis. The Markov model splits the days into different classes based on the history of the preceding days. This is to allow for different probabilities and amounts of rainfall on a day according to what happened previously: for example, in most climates, it is more likely to rain on a day following previous rain.

The daily states, order and type of Markov model are specified by the LIMITS, ORDER and HIGHORDER options, respectively. If the LIMITS option is set to a scalar or variate of length one, this defines the breakpoint between dry and wet days. A small positive value treats days with less than this amount of rainfall as dry days (these are also removed from the rainfall for wet days). If LIMITS is set to a variate of length of two or more, the rainfall states are defined as the days with rainfall less than or equal to these limits, with an extra group for rainfall greater than the top limit. The ORDER option specifies the number of previous days to use when forming the Markov classes. The classes are the combination of the daily states over the history length defined by ORDER. (So there will be (NVALUES(LIMITS)+1)**(ORDER+1) classes.) If there are two rainfall states, these are labelled w and d for wet and dry on each day. Otherwise they are labelled by the integers from 0 upwards. When there are two states, the default HIGHORDER=no gives all the unique combinations of wet and dry days over these days. Setting HIGHORDER=yes collapses the states to just the number of dry days preceding a wet day. For example, with ORDER=2 and HIGHORDER=no, the 8 states are ddd, ddw, dwd, dww, wdd, wdw,

wwd and www (where d = dry day and w = wet day); with ORDER=2 and HIGHORDER=yes, the 6 states are ddd, ddw, dw, wd, wdd, and ww, as dwd and dww are combined into dw and wwd and www are combined into ww. ORDER must be at between 0 and 3 for HIGHORDER=no and between 2 and 5 for HIGHORDER=yes.

The DAY option gives the dates or the day number within a year (1...366), and the DATA parameter gives the amount of rainfall on these dates. The data should be sorted into chronological order with no missing days. (Missing values should be entered for any days with no observations.) The INITIAL option can specify the amount of rain on the days preceding the first day in DATA; this should have ORDER values. If INITIAL is not set, the first ORDER days will not contribute to the counts and amounts.

You can save the summaries with the COUNTS, AMOUNTS, PROBABILITIES, CATEGORIES and STATECOUNTS parameters:

COUNTS	saves a table of counts classified by day number within the year (1...366) and Markov class (e.g. dd, wd, dw and ww);
AMOUNTS	saves a table of the sum of rainfall amounts classified by day and Markov wet classes (e.g. wd and ww);
PROBABILITIES	seves a pointer to a set of variates for each wet class giving probability of a wet day vs. a dry day for the days;
CATEGORIES	saves a factor giving the Markov class for each date; and
STATECOUNTS	saves a pointer to tables for each state defined by LIMITS, giving the counts by Markov class and day.

Printed output is controlled by the PRINT option, with settings:

counts	counts by day and Markov class;
amounts	amounts by day and wet Markov class; and
probabilities	probabilities by day and wet Markov class.

The summaries can be displayed in a spreadsheet by setting the SPREADSHEET option to the following settings:

counts	creates a sheet containing the counts for each day by the Markov classes;
amounts	shows the amounts of rainfall in the wet classes; and
probabilities	shows the probability of rainfall in the wet classes.

The spreadsheet can be saved to a file by setting the OUTFILE parameter to a Genstat or Excel spreadsheet filename (.gwb or .xlsx).

You can set option PLOT=probabilities to plot the probabilities. The TITLE parameter can supply a title for the graph; if this not set, a descriptive title will be created from the Markov chain options. The WINDOW parameter specifies the window to use for the graph.

Options: PRINT, PLOT, DAY, LIMITS, ORDER, HIGHORDER, INITIAL, SPREADSHEET.
Parameters: DATA, WINDOW, TITLE, COUNTS, AMOUNTS, PROBABILITIES, CATEGORIES, STATECOUNTS, OUTFILE.

Method

The procedure calculates the class of each day, and then tabulates these to create summaries. If dates are provided in DAY, these are converted to days in the year by the NDAYINYEAR function. Note: the 29 of February (which is only present in leap years) is day 60. The 1st March is always day 61.

Action with RESTRICT

The DATA or DAY variates can be restricted to analyse a subset of the data. If both DATA and DAY are restricted, the restrictions must be consistent.

Reference

Ong'ala, J.O. (2011). Simplifying the Markov chain analysis of rainfall data using Genstat. *MSc Thesis*, Maseno University.

See also

Procedures: `RFFAMOUNT`, `RFFPROBABILITY`.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

RGRAPH

Draws a graph to display the fit of a regression model (P.W. Lane).

Options

GRAPHICS = *string token*	Type of graphics to produce (lineprinter, highresolution); default high
TITLE = *text*	Title for the graph; default 'Fitted and observed relationship'
WINDOW = *number*	Which high-resolution graphics window to use; default 4 (redefined if necessary to fill the frame)
SCREEN = *string token*	Whether to clear the graphics screen before plotting (clear, keep); default clea
CIPLOT = *string token*	Whether to plot confidence intervals (no, yes); default no
CIPROBABILITY = *scalar*	Probability for confidence interval; default 0.95
BACKTRANSFORM = *string token*	What back-transformation to make (link, none, axis); default link
SAVE = *regression save structure*	Save structure of the model to display; default * uses the most recently fitted regression model

Parameters

INDEX = *variate*	Which explanatory variate to display; default * if GROUPS is set, otherwise INDEX is set to the first variate in the fitted model (must be set for nonlinear models other than standard curves)
GROUPS = *factor*	Which explanatory factor to display; default * if INDEX is set, otherwise GROUPS is set to the first factor in the fitted model (ignored for nonlinear models)

Description

Procedure RGRAPH displays the fit of either a linear regression, a generalized linear model, a generalized additive model, a standard curve or a nonlinear model.

For models other than the nonlinear models fitted by FITNONLINEAR or FIT with the CALCULATION option set, the graph shows the relationship between the response variate and either one explanatory variate or one explanatory factor or one of each. If no parameters are set, RGRAPH takes the first explanatory variate and the first factor in the model, and the predicted relationship is represented by a line for each level of the factor. The display represents the observed relationship as points, plotting the response (adjusted for further explanatory terms in the model, if any) against the chosen explanatory variate, with each point labelled according to the corresponding factor level. If no factor has been fitted, a single line is drawn, while if no variate has been fitted the graph simply shows the predicted mean for each level of the factor.

If a linear, generalized linear, or generalized additive model has been fitted, the INDEX and GROUPS parameters can be used to specify which explanatory variate and factor, respectively, should be used. If INDEX is set and GROUPS is not, a single line is drawn even if there are factors in the model; similarly if GROUPS is set and INDEX is not, the effect of the factor alone is shown. With generalized linear models, the relationship is usually plotted on the original scale, but you can set option BACKTRANSFORM to either none or axis, to plot on the scale of the linear predictor. These settings are useful, for example, if you want to check for potential non-linearity in the response. They differ in that the axis setting includes axis markings, back-transformed onto the natural scale, on the right-hand side of the y-axis. However, this is not available for log-ratio, power, reciprocal or calculated links.

For nonlinear models fitted by the FITNONLINEAR directive, a single line is drawn by joining the fitted values, and the response values are shown as points. Any setting of the GROUPS parameter is ignored. For curves fitted by the FITCURVE directive, settings of the INDEX and GROUPS parameters are ignored.

No graph can be drawn if the REG function has been used for any variate in the model. If the SSPLINE function has been used for any variate whose relationship with the response is not actually displayed, then the only adjustment for its effect will be the linear component of the fitted smooth curve. If the displayed variate itself is smoothed, then the curve is formed by interpolation between adjusted fitted values. The POL function is dealt with correctly.

The TITLE option can be used to supply a title for the graph. By default the graph is plotted on the current high-resolution device, but the GRAPHICS option can be set to line for a line printer plot. The WINDOW option can be used to select a pre-defined window for high-resolution plots; otherwise window 4 is used, and is redefined if necessary to fill the frame. The SCREEN option allows the graph to be added to an existing high-resolution plot. The colours and symbols used in the displays can be controlled by setting the attributes of the following pens with the PEN directive before calling the procedure:

pen 1	labels for lines when drawn for each level of a factor,
pen 2	fitted lines and means,
pen 3	points, and
pen 4	back-transformed axis marks and labels.

By default the current regression model is displayed, but option SAVE can be set to specify the save structure (from a MODEL statement) of some other model.

When there are no groups with a linear or generalized linear model, you can set option CIPLOT=yes to include confidence intervals for the fitted relationship. The CIPROBABILITY option sets the size of the interval. The default is 0.95 (i.e. 95%).

Options: GRAPHICS, TITLE, WINDOW, SCREEN, CIPLOT, CIPROBABILITY, BACKTRANSFORM, SAVE.
Parameters: INDEX, GROUPS.

Method

For a linear or generalized linear model, fitted lines are drawn by joining predicted values calculated by a PREDICT statement at 21 equally spaced values spanning the range of the explanatory variate. Alternatively, PREDICT provides adjusted means at each level of the factor, if the effect of the factor is to be displayed alone. If all the effects in the current model are displayed, the response is plotted against the explanatory variable (or against the factor level) as points. But if adjustment has to be made for some effects, adjusted response values (known as "partial residuals") are calculated by adding the simple residuals to predictions produced for all observations.

For generalized additive models, no predictions are used if the explanatory variate is smoothed: the nonlinear component of the smooth is extracted using RKEEP. For curves and general nonlinear models no predictions are made, and fitted values are used.

If a linear or generalized linear model is constrained to pass through the origin, the display will extend the range of the explanatory if necessary to include the origin.

The back-transformed axis markings, given by BACKTRANSFORM=axis, are added by using AXIS to including an additional (oblique) axis alongside the y-axis.

Action with RESTRICT

Any restriction that was in force when the model was fitted will apply also to the graphs. Problems may occur, however, if the response variate is not restricted and an explanatory variate or factor is restricted.

See also
Procedures: RCHECK, RDESTIMATES, AGRAPH, DTABLE, VGRAPH.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RIDGE

Produces ridge regression and principal component regression analyses (A.J. Rook & M.S. Dhanoa).

Options

PRINT = *string token*	What to print (correlation, pcp, ridge); default corr
PLOT = *string token*	Graphical output required (ridgetrace); default *

Parameters

Y = *variates*	Response variate in regression model
X = *pointers*	Containing explanatory variates in regression model

Description

Procedure RIDGE produces analyses for identifying and overcoming collinearity among the independent variates in a multiple regression analysis. The correlation matrix, variance inflation factors (the diagonal elements of the inverse of the correlation matrix) and the ratio of the squared error in the least squares regression coefficients to the expected squared error in orthogonal data are calculated. Principal component regressions excluding 1, 2 or 3 minor principal axes are calculated and transformed back to the original variables on either the original or standardized scale. The "Positive correlation spread association" (PCSA) (Vinod 1976) is also calculated. This is an overall measure of the suitability of the data for the application of principal component regression and ridge regression. Ridge regressions (Hoerl & Kennard 1970) are calculated and the ridge coefficients are printed together with 2 indices of stability proposed by Vinod (1976): the index of stability of relative magnitudes (ISRM) and the numerical largeness of more significant regression coefficients (NLMS). These are 0 and 1 respectively in orthogonal data. High-resolution graphs of the ridge trace can be plotted against Hoerl & Kennard's *k* scale and Vinod's *m* scale.

The parameters of the procedure are used to input the data: the Y parameter supplies the y-variate, and the X parameter specifies a pointer containing the x-variates. None of these variates must be restricted nor contain missing values.

Printed output is controlled by the PRINT option: correlation prints the correlation matrix, variance inflation factors and ratio of squared error to that in orthogonal data, pcp prints principal component analysis and principal component regression, and ridge prints ridge coefficients and stability parameters.

Graphical output is controlled by the PLOT option: ridgetrace produces ridge traces.

Options: PRINT, PLOT. Parameters: Y, X.

Method

The correlation matrix is produced using the CORRELATE directive. This is then used to calculate the variance inflation factors (VIF) and the ratio of the squared error to that in orthogonal data (RL).

Principal component analysis is carried out using the PCP directive. The standardized response variable is regressed on the principal component scores using MODEL, FIT and TERMS directives. The coefficients are then transformed back to the original variables on either the standardized or original scale with up to three principal components excluded. The correlations of the standardized variable with each of the principal components are also printed.

Ridge regression is carried out as described by Hoerl & Kennard (1970). Ridge coefficients on both the standardized and original scales are printed for values of the biasing parameter *k* between 0 and 1 together with the standard errors of the coefficients on the standardized scale.

Residual sums of squares (RSS) R-squared and total variance of the ridge coefficients (TVARB) are printed for each value of *k*. In addition Vinod's (1976) *m* scale is printed together with the index of stability of relative magnitudes (ISRM) and the numerical largeness of the more significant regression coefficients (NLMS).

High-resolution graphs of the ridge traces are produced. These are graphs of the ridge coefficients against *k* or against *m* and are plotted to the device set up prior to calling the procedure.

Action with RESTRICT

None of the input variates must be restricted.

References

Chatterjee, S. & Price, B. (1991). *Regression Analysis by Example (second edition)*. New York, Wiley.

Hoerl, A.E. & Kennard, R.W. (1970). Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55-67.

Vinod, H.D. (1976). Application of new ridge regression methods to a study of Bell system scale economies. *Journal of the American Statistical Association*, **71**, 835-841.

See also

Directive: PCP.

Procedure: LRIDGE, RLASSO.

Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

RJOINT

Does modified joint regression analysis for variety-by-environment data (P.W. Lane & K. Ryder).

Options

PRINT = *string tokens*	What to print (model, summary, estimates, monitoring, graph); default mode, summ, esti
TITLE = *text*	Overall title for graph
YTITLE = *text*	Y-axis title for graph
XTITLE = *text*	X-axis title for graph
TOLERANCE = *scalar*	Convergence criterion; default 0.001
MAXCYCLE = *scalar*	Maximum number of cycles; default 15
SAVE = *regression save structure*	Save structure from MODEL statement defining the model; default is to use the structure from the latest MODEL statement

Parameters

ENVIRONMENT = *factors*	The environment factor; no default
VARIETY = *factors*	The variety factor; no default
SENSITIVITIES = *variates*	To store estimates of sensitivities; default *
VARMEANS = *variates*	To store estimates of variety means; default *
ENVEFFECTS = *variates*	To store estimates of environment effects; default *
ENVMEANS = *variates*	To store estimates of environment means; default *
SESENSITIVITIES = *variates*	To store s.e.s of sensitivities; default *
SEVARMEANS = *variates*	To store s.e.s of variety means; default *
SEENVEFFECTS = *variates*	To store s.e.s of environment effects; default *
DEVIANCE = *scalar*	To store the residual deviance
DF = *scalar*	To store the residual d.f
EXIT = *scalar*	Exit status – set to 0 if the analysis converged, 1 otherwise

Description

Procedure RJOINT performs a modified joint regression analysis of data classified by two factors. This analysis is motivated by the study of variety-by-environment interactions in agriculture, where the two factors are varieties of some crop and environments at which experiments were carried out. The environments may be different sites within the same year, different years for the same site, or, as is more common, a combination of the two with little interest in individual year and site contributions. The intention is to characterize the sensitivity (or, inversely, the stability) of each variety to environmental effects by fitting a regression of the environment means for a variety on the average environment means. The model is thus nonlinear, of the form

$$y_{ij} = v_i + b_i \times e_j + \text{error}$$

where v_i are variety means, e_j are environment effects (with $\sum e_j = 0$) and b_i are the sensitivity parameters (with mean(b_i)=1). Usually, an experimenter is looking for varieties with large means and small sensitivities, to ensure a reliable crop under variable conditions. In RJOINT the factors are specified using the parameters VARIETY and ENVIRONMENT.

The data may consist of one value of the response, such as yield, for each combination of variety and yield. More often, the data are incomplete because not all varieties are tested at each environment; also, there may be multiple measurements of varieties at some environments. If the response is a count or a proportion, such as when investigating disease resistance, it will be more appropriate to use a generalized linear model based on a Poisson or binomial distribution and a

log or logit link function. The model should be specified by giving a MODEL statement before calling RJOINT; for example,

```
MODEL yield
```

You can choose to fit any generalized linear model by setting the DISTRIBUTION and LINK options of MODEL: thus, to model proportions, you could give a statement like

```
MODEL [DISTRIBUTION=binomial; LINK=logit; DISPERSION=*] \
      prop; NBINOMIAL=100
```

The iterative process used in the procedure is controlled by the options TOLERANCE and MAXCYCLE. At each iteration, the maximum difference between estimates of the sensitivity parameters in successive iterations is compared to the tolerance: the process ends when the differences are small enough, or when the maximum number of iterations is reached. The progress of the search can be followed by including the monitoring setting of the PRINT option, and the EXIT parameter can save a scalar with the value zero if the analysis converger and one otherwise.

Output is controlled by the option PRINT. The setting model prints a description of the model. The setting summary displays an analysis of variance (or deviance for non-Normal distributions) showing the effects of Varieties, Environments and Sensitivities: the last is the effect of allowing different sensitivities for each variety. The setting estimates displays two tables, one classified by varieties and the other by environments. For varieties the columns are: unadjusted means, final estimates of means (on the scale of the link function, if relevant), standard errors of estimates, back-transformed means (using the inverse of the link function), sensitivities, and standard errors of sensitivities. For environments the columns are: estimates of effects (on the scale of the link function, if relevant), standard errors of estimates, means (formed from the effects and the mean of the variety means), and back-transformed means (using the inverse of the link function). Finally, the setting graph plots the model. The TITLE option defines the onerall title for the graph; the default is Joint regression analysis. The YTITLE and XTITLE options define titles for the y- and x-axes, respectively; the default for the y-axix is the name of the y-variate, and the default for the x-axis is the name of the ENVIRONMENT factor.

The remaining parameters allow the following results to be saved: sensitivities, variety means, environment effects, environment means, and standard errors of sensitivities, variety means, environment effects, residual deviance and degrees of freedom. After calling the procedure, you can use the RKEEP directive to access fitted values and residuals. Other results from the fit that can be accessed via RKEEP or RDISPLAY may not be correct: for example, the number of residual d.f. shown by

```
RDISPLAY [PRINT=summary]
```

does not allow for the estimation of sensitivities.

Options: PRINT, TITLE, YTITLE, XTITLE, TOLERANCE, MAXCYCLE, SAVE.
Parameters: ENVIRONMENT, VARIETY, SENSITIVITIES, VARMEANS, ENVEFFECTS, ENVMEANS, SESENSITIVIT, SEVARMEANS, SEENVEFFECTS, DEVIANCE, DF, EXIT.

Method

The procedure uses iterative scheme (A) referred to in Digby (1979). The scheme has been generalized to deal with alternative distributions and link functions. First the environment effects are estimated with the sensitivity parameters set to 1, and then the procedure alternates between estimating the sensitivities with given environment effects and estimating environment effects with given sensitivities. Convergence is tested by comparing the maximum difference between old and new sensitivities against the criterion (default 0.001), but the maximum number of cycles (default 15) will not be exceeded. If the MAXCYCLE option is set to 1, the result is an unmodified joint regression analysis; see Finlay & Wilkinson (1963).

Action with `RESTRICT`

A restriction applied to the response variate will be taken into account. Residuals and fitted values will be formed only for the restricted subset of values. If levels of the factors are not represented in the restricted subset, then no results will be shown for those varieties and/or environments. Do not restrict the environment or variety factor differently to the response variate: results may then be incorrect.

References

Digby, P.G.N. (1979). Modified joint regression analysis for incomplete variety × environment data. *Journal of Agricultural Science, Cambridge*, **93**, 81-86.

Finlay, K.W. & Wilkinson, G.N. (1963). The analysis of adaptation in a plant-breeding programme. *Australian Journal of Agricultural Research*, **14**, 742-754.

See also

Procedures: `AMMI`, `GESTABILITY`, `GGEBIPLOT`. `RFINLAYWILKINSON`.

Genstat Reference Manual 1 Summary sections on: Regression analysis, REML analysis of linear mixed models.

RLASSO

Performs lasso using iteratively reweighted least-squares (D.A. Murray & P.H.C. Eilers).

Options

`PRINT` = *string token*	What output to print (`estimates`, `best`, `crossvalidation`, `progress`, `correlation`, `fitted`, `monitoring`); default `best`
`PLOT` = *string tokens*	What graphs to plot (`correlation`, `coefficients`); default `*` i.e. none
`TERMS` = *formula*	Explanatory model
`FACTORIAL` = *scalar*	Limit on number of factors/covariates in a model term; default 3
`LAMBDA` = *variate* or *scalar*	Values for the parameter lambda; must be set
`VALIDATIONMETHOD` = *string token*	Which cross-validation method to use (`crossvalidation`, `gcv`); default `gcv`
`NCROSSVALIDATIONGROUPS` = *scalar*	Number of groups for k-fold cross-validation; default 10
`NBOOT` = *scalar*	Number of times to bootstrap data to estimate standard errors and confidence limits for fitted values; default 100
`SEED` = *scalar*	Seed for random numbers to use in cross-validation and then in bootstrapping; default 0
`CIPROBABILITY` = *scalar*	Probability level for confidence interval for fitted values; default 0.95
`MAXCYCLE` = *scalar*	Maximum number of iterations for the iterative process
`TOLERANCE` = *variate*	Contains two values to define the convergence criterion for iterative least-squares and the adjustment to avoid division by zero in the penalty term; default `!(0.0001,1e-08)`

Parameters

`Y` = *variates*	Response variate
`BESTLAMBDA` = *scalars*	Saves the optimal lambda value from cross-validation
`CVSTATISTICS` = *matrices*	Saves the cross-validation statistics
`RESIDUALS` = *variates*	Saves residuals for the optimal `LAMBDA`
`FITTEDVALUES` = *variates*	Saves fitted values for the optimal `LAMBDA`
`ESTIMATES` = *variates*	Saves parameter estimates for the optimal `LAMBDA`
`SE` = *variates*	Saves standard errors of the parameter estimates for the optimal `LAMBDA`
`SEFITTED` = *variates*	Saves standard errors of the fitted values, from bootstrapping, for the optimal `LAMBDA`
`LOWER` = *variates*	Saves lower confidence limits for the fitted values, from bootstrapping, for the optimal `LAMBDA`
`UPPER` = *variates*	Saves upper confidence limits for the fitted values, from bootstrapping, for the optimal `LAMBDA`

Description

The `RLASSO` procedure performs L1-penalized regression (*lasso*) using iteratively reweighted sums of squares. The lasso method minimizes the residual sums of squares subject to the constraint that the sum of the absolute values of the model coefficients is less than a constant or tuning parameter λ.

The response variate is specified by the Y parameter. The model to be fitted is defined by the TERMS option. The FACTORIAL option sets a limit on the number of variates and/or factors in the model terms generated from the TERMS model formula (as in the FIT directive).

Printed output is controlled by the PRINT option, with settings:

estimates	to print, for each value of λ, the lasso coefficients their standard errors on the standardized and original scales,
best	prints the lasso estimates for the optimal λ,
crossvalidation	to print the cross-validation results, with optimal lambda value,
progress	shows the progress of the k-fold cross-validation,,
correlation	to print the correlations between the explanatory variables in the TERMS formula,
fitted	to print the fitted values for the optimal λ, with their standard errors and confidence limits,
monitoring	to print monitoring information during boot strapping.

By default,PRINT=best.

Graphical output is controlled by the PLOT option:

coefficients	plots the standardized coefficient estimates against the shrinkage factor, and correlation, and
correlation	uses the DCORRELATION procedure to produce a graphical representation of the correlation matrix for elements in TERMS.

By default, nothing is plotted.

The LAMBDA option must be set to a variate defining the values to try for the tuning parameter λ. The MAXCYCLE option specifies the number of iterations (default 200). The TOLERANCE option specifies the convergence criterion for the iterative procedure (default 0.0001), and the adjustment to use to avoid division by zero in the penalty term (default 10^{-8}).

The VALIDATIONMETHOD option controls how RLASSO estimates the tuning parameter λ:

crossvalidation	uses k-fold cross-validation where the prediction error is calculated using the mean squared error,
gcv	uses the generalized cross-validation, as specified by Tibshirani (1996).

By default , VALIDATIONMETHOD=gcv.

For k-fold cross-validation the NCROSSVALIDATIONGROUPS option defines the number of subsets to use (default 10). The data are divided into roughly equal-sized subsets and the model is fitted with each subset removed in turn. The mean squared error is calculated for the omitted subset based on the model from fitting the remaining subsets. The value that minimizes the mean prediction error is taken as the optimal λ, and used to get the lasso estimates. The optimal value of λ can be saved by the BESTLAMBDA parameter, and the prediction error values can be saved by the CVSTATISTICS parameter.

RLASSO can use bootstrapping to provide standard errors and lower and upper confidence intervals for the fitted values. The NBOOT option specifies the number of bootstrap samples that are taken, and the CIPROBABILITY option sets the size of the confidence limits.

You can save results from the optimal fit using the RESIDUALS, FITTEDVALUES, ESTIMATES and SE, SEFITTED, LOWER and UPPER parameters. Note that the residuals are the simple residuals, rather than standardized residuals.

Options: PRINT, PLOT, TERMS, FACTORIAL, LAMBDA, VALIDATIONMETHOD, NCROSSVALIDATIONGROUPS, NBOOT, SEED, CIPROBABILITY, MAXCYCLE, TOLERANCE.

Parameters: Y, BESTLAMBDA, CVSTATISTICS, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, SEFITTED, LOWER, UPPER.

Method

Lasso is carried out by using iteratively reweighted least-squares. RLASSO approximates the absolute sum of the coefficients $\sum|\beta|$ by $\sum(\beta^2/|\beta|)$, and the penalty term $\lambda\sum(\beta^2/|\beta|)$ is imposed on the sum of squares of the parameter estimates β. The penalty term is applied to the diagonal elements of the sums-of-squares-and-products matrix by setting the RIDGE option of the TERMS directive. For a given value of λ, the algorithm iterates to find the lasso estimates. The shrinkage factor *s* is estimated by

$$s = t / \sum|\beta^{(0)}|$$

where $\sum|\beta^{(0)}|$ is the absolute sum of the full least squares estimates, and *t* is the absolute sum of the lasso estimates subject to

$$t \leq \sum|\beta^{(0)}|.$$

The columns of the design matrix in TERMS are standardized. However, estimated coefficients are available for both the standardized and unstandardized data.

Action with RESTRICT

There must be no restrictions.

References

Hastie, T., Tibshirani, R. & Friedman, J (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd Edition*. Springer, New York.

Tibshirani, R. (1996). Regression shrinkage and selection by lasso. *Journal of the Royal Statistical Society, Series B*, **58**, 267-288.

See also

Procedure: LRIDGE.

Genstat Reference Manual 1 Summary section on: Regression analysis.

RLFUNCTIONAL

Fits a linear functional relationship model (M.S. Dhanoa & D.B. Baird).

Options

PRINT = *string token*	Controls printed output (summary, estimates, fittedvalues, confidencelimits, grouptests); default summ, esti, conf, grou
METHOD = *string tokens*	Specifies what methods to use to fit the regression (bartlett, majoraxis, errorsinvariables, yonx, xony, reducedmajoraxis, standardmajoraxis, rangedmajoraxis, geometricmean, bisector, medyonx, medxony, qgeometricmean, qbisector); default bart
PLOT = *string tokens*	Controls what to plot (fitted, residuals, bootestimates, confidencelimits); default fitt
TITLE = *text*	The title for the analysis; default title uses the Y and X identifiers
NBOOT = *scalar*	The number of samples to take for the bootstrap confidence limits; default 200
SEED = scalar	Seed for bootstrap randomization; default 0
CIPROBABILITY = *scalar*	Defines the size of the confidence interval; default 0.95 i.e. 95%
CIMETHOD = *string token*	Method for confidence limits (parametric, bootstrap); default boot
GMETHOD = *string token*	Method for comparing slopes, elevations and locations between groups (majoraxis, standardmajoraxis); default uses standardmajoraxis for METHOD settings standardmajoraxis, reducedmajoraxis, rangedmajoraxis, geometricmean or bisector, and majoraxis otherwise
VRATIO = *scalar*	Ratio between variance of Y and X variables for METHOD=errorsinvariables; default 1
YRANGEMETHOD = *string token*	Type of range used for Y when METHOD=rangedmajoraxis (relative, interval); default rela
XRANGEMETHOD = *string token*	Type of range used for X when METHOD=rangedmajoraxis (relative, interval); default rela
WINDOW = *scalar*	Graphics window to use for fitted-value plots; default 1
KEYWINDOW = *scalar*	Graphics window to use for key; default 2

Parameters

Y = *variates*	Y-variate for each model
X = *variates*	X-variate for each model
SLOPE = *scalars*, *variates* or *matrices*	Saves the estimated slopes
INTERCEPT = *scalars*, *variates* or *matrices*	Saves the estimated intercepts
GROUPS = *factors*	Defines groups of units

RESIDUALS = *variates*, *matrices* or *pointers*
: Saves the residuals from the fitted models

FITTEDVALUES = *variates*, *matrices* or *pointers*
: Saves the fitted values

ESTIMATES = *variates*, *matrices* or *pointers*
: Saves the estimates

SE = *variates*, *matrices* or *pointers*
: Saves the standard errors of the estimates

LOWER = *variates*, *matrices* or *pointers*
: Saves lower values of confidence intervals for the estimates

UPPER = *variates*, *matrices* or *pointers*
: Saves upper values of confidence intervals for the estimates

LOWFITTEDVALUES = *variates*, *matrices* or *pointers*
: Saves the lower confidence limits from a bootstrap analysis of fitted values

UPPFITTEDVALUES = *variates*, *matrices* or *pointers*
: Saves the upper confidence limits from a bootstrap analysis of fitted values

TESTPROBABILITIES = *pointers*
: Saves the between-group test probabilities (in a symmetric matrix) for differences in slopes, elevations and locations

Description

RLFUNCTIONAL can be used to estimate the slope and intercept of a linear equation describing the relationship between two variables, when the observations on both variables are subject to error variation. This contrasts with the situation in ordinary linear regression, where we assume that only the y-variate is subject to error (the x-variate is assumed to be observed exactly). If the variation in the x-values is not accounted for, the estimate of the slope will be biased towards zero. For further details see Sokal & Rohlf (1995, Section 14.13) and Bartlett (1949). RLFUNCTIONAL can also fit standard linear regression models and quantile regression models so that these can be compared with the functional relationship models.

The y- and x-variates must be specified by the Y and X parameters respectively. The estimation methods to use are specified by the METHOD option, using the following settings.

bartlett
: uses Bartlett's three-group method (default).

majoraxis
: takes the major axis from a principal component analysis (this assumes that X and Y are equally variable).

errorsinvariables
: This fits a model that assumes the errors in Y and X are in proportion to the value specified by the VRATIO option. When VRATIO is one, this gives the same estimates as majoraxis (but not the same parametric confidence limits).

yonx
: uses ordinary least squares with the dependent variable Y and independent variable X.

xony
: uses ordinary least squares but with the dependent variable X and independent variable Y.

reducedmajoraxis
: estimates the slope as the geometric mean of the regression coefficients from regressions of Y on X and X on Y.

standardmajoraxis
: takes the geometric mean of the ordinary regression slopes (Y on X and X on Y). This is the same as reduced major axis regression, except that a different parametric estimator is

used for the confidence limits.

`rangedmajoraxis`	This scales the `Y` and `X` variables before fitting a major axis regression. The scalings are controlled by the `YRANGEMETHOD` and `XRANGEMETHOD` options, respectively. The `relative` setting scales the variable by its maximum, while the `interval` setting uses its range. With the `relative` setting, the values of the variable should all be positive.
`geometricmean`	takes the geometric mean of the ordinary regression slopes (`Y` on `X` and `X` on `Y`). This is the same the reduced major axis regression, except that a different parametric estimator is used for the confidence limits.
`bisector`	estimates the slope as the bisector of the ordinary regression slopes (`Y` on `X` and `X` on `Y`).
`medyonx`	fits the median (50% quantile) regression of `Y` on `X`.
`medxony`	fits the median (50% quantile) regression of `X` on `Y`.
`qgeometricmean`	takes the geometric mean of the median regression slopes (`Y` on `X` and `X` on `Y`).
`qbisector`	estimates the slope as the bisector of the median regression slopes (`Y` on `X` and `X` on `Y`).

The `GROUPS` parameter allows a factor to be specified to define groupings of the data units, so that separate relationships can be investigated for each group. The probabilities of differences in slopes, elevations (assuming a common slope) and locations (assuming a common slope and intercept for each group) between groups can be printed, or saved in a pointer using the `TESTPROBABILITIES` parameter. The pointer has three elements (labelled `'slopes'`, `'elevations'` and `'locations'`) which save symmetric matrices. The element on the diagonal of each symmetric matrix contains the overall probability that all groups have the same estimates, and the lower triangle contains the pairwise probabilities that two groups have the same estimates. The `GMETHOD` option allows you to specify whether the `majoraxis` or `standardmajoraxis` method is used to calculate these tests; the default is to use `standardmajoraxis` for `METHOD` settings `standardmajoraxis`, `reducedmajoraxis`, `rangedmajoraxis`, `geometricmean` or `bisector`, and `majoraxis` for the other `METHOD` settings. For details of the tests see Warton *et al.* (2006).

The `PRINT` option controls printed output, with settings:

`summary`	summary of the analyses,
`estimates`	estimated slopes and intercepts with standard errors,
`fittedvalues`	fitted values and residuals,
`confidencelimits`	includes confidence intervals with the estimates,
`grouptests`	tests of slopes, elevations and locations between groups.

The default is `PRINT=summ,esti,conf,grou`.

The `PLOT` option controls what graphs are printed, with settings:

`fitted`	creates a graph showing the observed data and the lines fitted by the various methods (all on a single graph),
`residuals`	uses the `DRESIDUALS` procedure to display diagnostic plots of the residuals from each method,
`bootestimates`	creates a histogram with a kernel-density smooth of the estimates from the bootstrap analysis for each method, and
`confidencelimits`	plot the fitted model for each method, with lower and upper confidence limits.

The `TITLE` option can supply a title for these plots. When there are no groups, the `WINDOW` option specifies the window to use for the `fitted` plot and `each confidence` plot, and the

KEYWINDOW specifies the window to use for their keys. If there are groups, these graphs are plotted in a trellis arrangement to show all results from every group simultaneously.

The slope and intercept can be saved individually using the SLOPE and INTERCEPT parameters, or together using the ESTIMATES parameter. Their standard errors can be saved using the SE parameter. Residuals and fitted values can be saved using the FITTEDVALUES and RESIDUALS parameters. Lower and upper values from a confidence interval for the estimates can be saved using the LOWER and UPPER parameters. The probability for the confidence interval is specified by the CIPROBABILITY option (default 0.95 i.e. 95%). The type of confidence interval (parametric or bootstrap) is controlled by the CIMETHOD option. The randomization seed for CIMETHOD=bootstrap is specified by the SEED option; the default of zero continues an existing sequence of random numbers if any have already been used in the current Genstat job, or obtains a random seed using system clock if none have been used already. The number of bootstrap samples is specified by the NBOOT option (default 200). When bootstrap confidence intervals are used, the upper and lower confidence interval for the fitted values can be saved using the LOWFITTEDVALUES and UPPFITTEDVALUES parameters.

If there are no groups and a single method, SLOPE and INTERCEPT save their estimates in scalars, while ESTIMATES, SE, FITTEDVALUES, RESIDUALS, LOWER and UPPER save variates. Alternatively, if there are groups and a single method, SLOPE and INTERCEPT save their estimates in variates, while ESTIMATES, SE, FITTEDVALUES, RESIDUALS, LOWER and UPPER save matrices with a column for each group. If there are several methods, each of these parameters saves a pointer with elements labelled by the relevant METHOD setting. The pointer elements are scalars, variates or matrices according to what is being saved and whether there are groups (as defined above).

Options: PRINT, METHOD, PLOT, TITLE, NBOOT, SEED, CIPROBABILITY, CIMETHOD, GMETHOD, VRATIO, YRANGEMETHOD, XRANGEMETHOD, WINDOW, KEYWINDOW.

Parameters: Y, X, SLOPE, INTERCEPT, GROUPS, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, LOWER, UPPER, LOWFITTEDVALUES, UPPFITTEDVALUES, TESTPROBABILITIES.

Method

RLFUNCTIONAL uses the methods described in Section 14.13 of Sokal & Rohlf (1995) and Warton *et al.* (2006). For further information, see *Dhanoa et al.* (2011).

Action with RESTRICT

If either the Y or X variates is restricted, the model is estimated using only the units not excluded by the restriction.

References

Bartlett, M.S. (1949). Fitting a straight line when both variables are subject to error. *Biometrics*, **5**, 207-212.

Dhanoa, M.S., Sanderson, R., Lopez, S., Dijkstra, E., Kebreab, E. & France, J. (2011). Regression procedures for relationships between random variables. In: *Modelling nutrient digestion and utilization in farm animals* (ed. D. Sauvant, J. Van Milgen, P. Faverdin & N. Friggens), 31-39. Wageningen Academic Publishers, Wageningen.

Sokal, R.R. & Rohlf, F.J. (1995). *Biometry (3rd edition)*. W.H. Freeman & Company, New York.

Warton, D.I., Wright, I.J., Falster, D.S. & Westoby, M. (2006). Bivariate line-fitting methods for allometry. *Biological Reviews*, **81**, 259-291.

See also

Directive: `PCP`.

Genstat Reference Manual 1 Summary sections on: Regression analysis, Multivariate and cluster analysis.

RLIFETABLE

Calculates the life-table estimate of the survivor function (D.A.Murray).

Options

PRINT = *string tokens*	Controls printed output (lifetable); default life
PLOT = *string tokens*	Type of graph to be plotted (survivor, hazard, pdf); default surv, haza, pdf
INTERVAL = *scalar* or *variate*	A scalar defining the width of the intervals or a variate containing the boundaries of the intervals

Parameters

TIMES = *variates*	Observed timepoints
CENSORED = *variates*	Variate specifying whether the corresponding element of each TIMES variate is censored (1) or represents failures (0)
FREQUENCY = *variates*	Variate containing frequencies for the elements of TIMES; by default these are all assumed to be 1
GROUPS = *factors*	Factor specifying the different groups for which to estimate life tables
LIFETABLE = *pointers*	Pointer to variates to save the information from each life table

Description

RLIFETABLE calculates the life-table estimate, or actuarial estimate, of the survivor function. The life-table method requires a fairly large number of observations so that survival times can be grouped into intervals. These are specified using the INTERVALS option. For equal intervals, you can set INTERVALS to a scalar to define their width. Alternatively you can set INTERVALS to a variate containing the lower boundaries of the intervals. The PLOT option can be used to produce plots of the survivor function (survivor), estimated hazard function (hazard) and the probability density function (pdf). You can set the option PRINT=* to suppress printing of the life table; by default PRINT=lifetable.

The observed timepoints (or the timepoints at which censoring took place) are specified using the TIMES parameter. The CENSORED parameter specifies a variate containing the value one if the corresponding element of TIMES is censored or zero if it was not. CENSORED can be omitted if there was no censoring. If there are several observations (all censored or all uncensored) at a time point, you can specify the time point only once and define the number of observations by specifying a variate of counts using the FREQUENCY parameter. This is particularly useful if the contents of the TIMES variate are intended to identify time intervals rather than discrete time points. The GROUPS parameter can be used to request separate life tables for different groups of data. The LIFETABLE parameter allows the life table to be saved in a pointer to a set of variates for each of the columns within the table.

Options: PRINT, PLOT, INTERVAL.
Parameters: TIMES, CENSORED, FREQUENCY, GROUPS, LIFETABLE.

Method

The methodology in RLIFETABLE is based on that described in Chapter 4 of Lee (1992).

Action with RESTRICT

The input variates and factors may be restricted identically. The life tables are based only on the units not excluded by the restriction.

Reference

Lee, E.T. (1992). *Statistical Methods for Survival Data Analysis*. Wiley, New York.

See also

Procedures: KAPLANMEIER, RPHFIT, RPROPORTIONAL, RSTEST, RSURVIVAL.
Genstat Reference Manual 1 Summary section on: Survival analysis.

RMGLM

Fits a model where different units follow different generalized linear models (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring); default mode, summ, esti
Y = *variate*	Response variate
TERMS = *formula*	Terms in the model
NBINOMIAL = *variate*	Binomial totals
DISPERSION = *scalar*	Dispersion parameter; default * for DIST=norm, gamm, inve or calc, and 1 for DIST=pois, bino, mult, nega, geom, expo or bern
WEIGHTS = *variate*	Prior weights; default 1
OFFSET = *variate*	Offset variate to be included in model; default * i.e. none
CONSTANT = *string token*	How to treat the constant (estimate, omit, ignore); default esti
FACTORIAL = *scalar*	Limit for expansion of model terms; default 3
FULL = *string token*	Whether to assign all possible parameters to factors and interactions (no, yes); default no
DATASET = *factor*	Indicates which generalized linear model to apply to each unit; default defined from NVALUES
LINEARPREDICTOR = *variate*	Initial values for linear predictor
MAXCYCLE = *scalar*	Maximum number of iterations; default 30
MVINCLUDE = *string token*	Whether to include units with missing values in the explanatory factors and variates (explanatory); default * i.e. omit these
SAVE = *identifier*	To name the regression save structure; default *

Parameters

NVALUES = *scalars*	Number of units for each generalized linear model
DISTRIBUTION = *string tokens*	Error distributions (normal, poisson, binomial, gamma, inversenormal, multinomial, calculated, negativebinomial, geometric, exponential, bernoulli); default norm
LINK = *string tokens*	Link functions (canonical, identity, logarithm, logit, reciprocal, power, squareroot, probit, complementaryloglog, calculated, logratio); default cano (i.e. iden for DIST=norm or calc; loga for DIST=pois; logi for DIST=bino, bern or mult; reci for DIST=gamm or expo; powe for DIST=inve; logr for DIST=nega or geom)
EXPONENT = *scalars*	Exponent for power links

Description

RMGLM is useful if you want to fit a model where there are several generalized linear models, each one applying to a different set of data units. This is required, for example, in the fitting of hierarchical generalized linear models (see HGANALYSE), and would also allow the fitting of multivariate generalized linear models.

The NVALUES parameter can specify a list of scalars defining the number of units following each generalized linear model. If NUNITS is used, the units are assumed to be ordered so that all the units with the first generalized linear model come first, then those with the second one, and so on. The DATASET option can then save a factor to indicate which generalized linear model applies to each unit. Alternatively, you can specify a list of null settings (*) for NVALUES, and supply a pre-defined factor using the DATASET option. The DISTRIBUTION parameter specifies the error distributions, the LINK parameter specifies the link function, and the EXPONENT exponent parameter specifies the exponent where there is a power link.

The Y option specifies response variate, and the NBINOMIAL option specifies the totals for binomial data. Prior weights can be supplied using the WEIGHTS option. The TERMS option specifies the terms to be fitted, and the FULL option controls the parameterization, as in the TERMS directive. The MVINCLUDE option allows units with missing values with missing values in factors or variates in the model to be included (by default these are excluded). Where this occurs, the factor or variate is taken to make no contribution to the fitted value for the unit concerned (see TERMS for more details).

The CONSTANT option indicates whether or not to fit a constant, and the FACTORIAL option specifies a limit (default 3) on the number of variates and factors in each term, as in the FIT directive. An offset can be supplied using the OFFSET option. The LINEARPREDICTOR option can supply initial values for linear predictor, and the MAXCYCLE option can set a limit (default 30) on the number of iterations. Printed output is controlled by the PRINT option, with the same settings as in the FIT directive.

After the fit, the RDISPLAY directive can be used to generate additional output, and the RKEEP directive can be used to save information, in the usual way.

Options: PRINT, Y, TERMS, NBINOMIAL, DISPERSION, WEIGHTS, OFFSET, CONSTANT, FACTORIAL, FULL, DATASET, LINEARPREDICTOR, MAXCYCLE, MVINCLUDE, SAVE.
Parameter: NVALUES, DISTRIBUTION, LINK, EXPONENT.

Method

RMGLM uses the calculated settings of the DISTRIBUTION and LINK options of MODEL.

Action with RESTRICT

You can restrict the units that Genstat will use for the fit by putting a restriction on the response variates, weight variate, offset variate, binomial totals, or any explanatory variate or factor. However, you must then supply the initial values for linear predictor (using the LINEARPREDICTOR option), as the default calculation requires use of RESTRICT.

See also

Procedures: GLM, HGANALYSE.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RMULTIVARIATE

Performs multivariate linear regression with accumulated tests; synonym FITMULTIVARIATE (H. van der Voet).

Options

PRINT = *string tokens*	Controls printed output (model, summary, accumulated); default mode, summ, accu
RPRINT = *string tokens*	Controls printed output from the univariate regression analyses (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring); default *
FACTORIAL = *scalar*	Limit for expansion of model terms; default 3
NOMESSAGE = *string tokens*	Which warning messages to suppress when fitting the complete model – messages are always suppressed when fitting models for individual tests (aliasing, marginality); default *
RESULTS = *pointer*	To save results from accumulated and summary tests in a pointer containing terms, degrees of freedom of terms, Wilks' Lambda, Rao's F-statistic, degrees of freedom for numerator and denominator of Rao's F and P-value of Rao's F

Parameter

TERMS = *formula*	List of explanatory variates and factors, or model formula

Description

RMULTIVARIATE calculates hierarchical tests for all terms in a multivariate linear regression model. These tests are based on Wilks' Lambda. The use of RMULTIVARIATE must be preceded by a MODEL statement to define the response variables and, if required, a vector of weights and an offset. Generalized linear models are not allowed. Note that the FIT directive performs a regression analysis for each of the response variables in turn, whereas RMULTIVARIATE performs multivariate modelling and testing.

The TERMS parameter specifies the model terms to be assessed. The FACTORIAL option sets a limit on the number of factors and variates in each term, similarly to the FACTORIAL option of FIT; by default this is 3. Printed output from the multivariate analysis is controlled by the PRINT option: model gives a description of the model, summary prints test results for the full model, while accumulated gives accumulated test results for each term in the model formula. The RPRINT option controls output from univariate regressions of the individual variates, which are performed (by FIT) in order to calculate the multivariate analysis. The NOMESSAGE option can be used to suppress aliasing and marginality warning messages when fitting the full model.

The RESULTS option can be used to save both accumulated and summary test results in a pointer. This pointer contains a text structure saving the individual model terms and six variates saving the number of degrees of freedom associated with each term, Wilks' Lambda, Rao's F-statistic, degrees of freedom for numerator and denominator of Rao's F-statistic and the calculated P-value. Directives RDISPLAY and RKEEP can be used subsequent to RMULTIVARIATE, to display further output and store results from the univariate regressions of each response variate.

Units with one or more missing values in any term are excluded from the analysis. This implies that successive calls of RMULTIVARIATE may give different test results if terms with missing values are dropped or added.

Options: PRINT, RPRINT, FACTORIAL, NOMESSAGE, RESULTS.
Parameter: TERMS.

Method
The implementation is straightforward using Genstat regression and the FSSPM directive. Terms in the multivariate linear model are tested by Rao's F-approximation for Wilks' Lambda (Rao 1973).

Action with RESTRICT
Any restriction applied to vectors used in the regression model will apply also to the results from RMULTIVARIATE.

Reference
Rao, C.R. (1973). *Linear Statistical Inference and its Applications*. Wiley, New York.

See also
Procedures: MANOVA, MVAOD.
Genstat Reference Manual 1 Summary sections on: Multivariate and cluster analysis, Repeated measurements.

RNEGBINOMIAL

Fits a negative binomial generalized linear model estimating the aggregation parameter (R.M. Harbord & R.W. Payne).

Options

†PRINT = *string tokens*	Printed output from the analysis (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, confidence, aggregation, loglikelihood); default mode, summ, esti, aggr
AGGREGATION = *scalar*	Saves the estimate of the aggregation parameter
_2LOGLIKELIHOOD = *scalar*	Saves the value of −2 × log-likelihood
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on number of factors in a treatment term; default 3
†POOL = *string token*	Whether to pool the deviance for the terms in the accumulated summary (yes, no); default no
NOMESSAGE = *string tokens*	Warnings to suppress from FIT (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default *
FPROBABILITY = *string token*	Printing of probabilities for variance ratios (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-statistics (yes, no); default no
SELECTION = *string tokens*	Statistics to be displayed in the summary of analysis produced by PRINT=summary (%variance, %ss, adjustedr2, r2, dispersion, %meandeviance, %deviance, aic, bic, sic); default disp
†PROBABILITY = *scalar*	Probability level for confidence intervals for parameter estimates; default 0.95
SEAGGREGATION = *scalar*	Saves the standard error of the estimated aggregation parameter
MAXCYCLE = *variate*	Maximum number of iteration for main and Newton-Raphson estimations; default !(15,15)
TOLERANCE = *variate*	Convergence criteria for deviance and *k*; default !(1E-4,1E-4)

Parameter

TERMS = *formula*	List of explanatory variates and factors, or model formula (as for FIT)

Description

The negative binomial distribution can be fitted as a generalized linear model using FIT only for a given value of the aggregation parameter *k*. RNEGBINOMIAL extends the fitting to include estimation of *k* from the data.

The negative binomial distribution is a discrete distribution with the relationship between mean and variance given by

variance = mean + mean**2/*k*,

where *k* is a positive constant known as the aggregation parameter. It provides a possible model for count data that show apparent overdispersion when a Poisson model is fitted. (Another model is the simpler constant overdispersion model, obtained by setting option DISPERSION=* in a MODEL statement with option DISTRIBUTION=poisson; see McCullough & Nelder 1989 and

Hinde & Demetrio 1998.)

The call to RNEGBINOMIAL must be preceded by a MODEL statement with option DISTRIBUTION=negativebinomial (otherwise an error message is printed). It is also necessary to specify the link function (e.g. by setting option LINK=logarithm for a log-link), as the default is the canonical log-ratio link, which is unlikely to be useful in practice (for example it requires the linear predictor to be negative).

The AGGREGATION and SEAGGREGATION option allow the estimate of *k* and its standard error to be saved. The _2LOGLIKELIHOOD option allows minus twice the maximized log-likelihood to be saved. This may be useful for comparing a sequence of nested models fitted by RNEGBINOMIAL using likelihood ratio testing. (The deviance cannot be used to compare models unless the value of *k* is the same for all the models, as it is the difference between the log-likelihood of a given model and a saturated model with the same value of *k*.) Printed output is controlled by the PRINT option, which has the same settings as for the FIT directive but with the addition of aggregation to control the printing of the estimate of *k* and its standard error (based on observed rather than expected information; see *Method*), and loglikelihood to print minus two times the log-likelihood.

The CONSTANT, FACTORIAL, POOL, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION and PROBABILITY options operate in the usual way (as for example in the FIT directive). The final two options, MAXCYCLE and TOLERANCE, can supply variates of length 2 that can be used to control the iterative process if required. The first element of MAXCYCLE sets the maximum number of times that the model is fitted as a generalized linear model for fixed *k*, while the second element sets the maximum number of Newton-Raphson iterations used to maximise the likelihood with respect to *k* for fixed fitted values. The alternating cycle stops when successive values of the deviance are within a tolerance set by the first element of the TOLERANCE option and successive values of the deviance are within a tolerance set by the second element.

Options: PRINT, AGGREGATION, _2LOGLIKELIHOOD, CONSTANT, FACTORIAL, POOL, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION, PROBABILITY, SEAGGREGATION , MAXCYCLE, TOLERANCE.
Parameter: TERMS.

Method

For fixed *k*, the negative binomial distribution is in the exponential family and the regression parameters determining the fitted values can be fitted as a generalized linear model using the FIT directive. For a fixed set of fitted values, *k* can be estimated by using the Newton-Raphson method to solve the score equation for *k*. Alternating between the two processes until convergence yields joint maximum likelihood estimates of *k* and the regression parameters. As the estimate of *k* is asymptotically independent of the other regression parameters (Lawless 1987), their standard errors can be obtained separately from the two processes. The standard error for *k* uses observed rather than expected information due to the use of Newton-Raphson rather than Fisher scoring.

The starting value of *k* is taken from the AGGREGATION option of the MODEL statement, which defaults to 1. This default appears to be a satisfactory initial value in practice, but the user may wish to specify a different value if convergence problems are encountered, or if speed is an issue and an approximate value of *k* is known.

Action with RESTRICT

Any restriction applied to vectors used in the regression model applies also to the results from RNEGBINOMIAL.

References

McCullagh, P. & Nelder, J.A. (1989). *Generalized Linear Models (second edition)*. Chapman & Hall, London.

Hinde, J. & Demetrio, C.G.B. (1998). Overdispersion: models and estimation. *Computational Statistics & Data Analysis*, **27**, 151-170.

Lawless, J.F. (1987). Negative binomial and mixed Poisson regression. *Canadian Journal of Statistics,* **15**, 209-225.

See also

Procedures: HGANALYSE, R0INFLATED.

Genstat Reference Manual 1 Summary section on: Regression analysis.

RNONNEGATIVE

Fits a generalized linear model with nonnegativity constraints; synonym FITNONNEGATIVE (P.W. Goedhart & C.J.F. ter Braak).

Options

PRINT = *string tokens*	Printed output required (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring); default mode, summ, esti
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
POOL = *string token*	Whether to pool ss in accumulated summary between all terms fitted in a linear model (yes, no); default no
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality); default *
FPROBABILITY = *string token*	Printing of probabilities for variance ratios (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-statistics (yes, no); default no
MAXCYCLE = *scalar*	Maximum number of iterations; default 100
TOLERANCE = *scalar*	Value against which the Kuhn-Tucker values are tested; default 10^{-8}
INITIALMODEL = *string token*	Initial model from which to start the iterative procedure (null, full, positive, own); default null
OWNINITIAL = *variates*	Specifies the variates that compose your own initial model; this option must be set when INITIALMODEL=own; default *
FORCED = *formula*	Model formula which is fitted irrespective of nonnegativity constraints; default *

Parameter

X = *variates*	List of predictors which are subject to nonnegativity constraints

Description

It is sometimes useful to impose nonnegativity constraints on regression coefficients. For example, the fitting of monotone regression splines (Ramsay 1988) requires nonnegative regression coefficients. Another example is regression of spectral data to determine the amounts of substances in a mixture. If an additive model holds, with the absorbance profiles as regressors, the amounts are estimated by the regression coefficients which should therefore be nonnegative. Note that an ordinary regression problem with general linear inequality constraints may be solved by using the solution to a derived regression problem with nonnegativity restrictions (Kennedy & Gentle 1980).

A call to RNONNEGATIVE must be preceded by a MODEL statement which defines the response variate and, if required, all other aspects of a generalized linear model. Only the first response variate is analysed. The only parameter, X, must be set to a list of explanatory variates which are subject to the nonnegativity constraints. The predictors with nonnegative coefficients are found by an iterative procedure which is explained in the method section. RDISPLAY and RKEEP can

be used subsequent to RNONNEGATIVE.

Options PRINT, CONSTANT, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY and TPROBABILITY are similar to the options of the FIT directive. Setting PRINT=monitoring provides monitoring of the iterative procedure. The MAXCYCLE option can be used to specify the maximum number of iterations. If the iterative procedure has not converged within the maximum number of iterations, a warning message is printed. The INITIALMODEL option provides different starting points for the iterative procedure. Setting null starts with no predictors in the initial model, full starts with all predictors, while the positive setting starts with those predictors that have a strictly positive regression coefficient in the full model. Finally, INITIALMODEL=own enables you to specify your own starting point. Option OWNINITIAL must then be set to a subset of predictors listed by the X parameter. Aliased terms, if any, are dropped after fitting the initial model. The use of the TOLERANCE option is explained in the method section.

It is sometimes desirable to include some predictors irrespective of the sign of their regression coefficient. Such predictors may be specified by means of the FORCED option. FORCED can be set to any model formula, i.e. it may contain factors and interactions as well as variates. The FORCED model formula is fitted first.

Units with one or more missing values in any term of the FORCED formula or the X predictors are excluded from the analysis. This implies that FIT used for a subset of predictors may give different results than RNONNEGATIVE.

Options: PRINT, CONSTANT, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, TPROBABILITY, MAXCYCLE, TOLERANCE, INITIALMODEL, OWNINITIAL, FORCED.
Parameter: X.

Method

For ordinary regression problems, the problem is to find the linear least squares solution subject to nonnegativity constraints, i.e.

$$\min_{\mathbf{b}} \|\mathbf{y} - \mathbf{X}\,\mathbf{b}\| \qquad \text{subject to } \mathbf{b} \geq 0$$

The Kuhn-Tucker conditions (Kennedy & Gentle 1980) are necessary and sufficient for finding the regression model with minimal sums of squares. These conditions are

$$KT1_j = [\mathbf{X}^T(\mathbf{y} - \mathbf{X}\,\mathbf{b})]_j = 0 \qquad \text{if } b_j > 0$$
$$KT2_j = [\mathbf{X}^T(\mathbf{y} - \mathbf{X}\,\mathbf{b})]_j \leq 0 \qquad \text{if } b_j = 0$$

These conditions also hold when only a subset of regression coefficients are subject to the nonnegativity constraint. In weighted regression, with diagonal matrix of weights **W**, the Kuhn-Tucker values are given by $[\mathbf{X}^T\mathbf{W}(\mathbf{y} - \mathbf{X}\,\mathbf{b})]$.

Lawson & Hanson (1974) use these conditions in an algorithm which begins with $\mathbf{b} = \mathbf{0}$. Next, b_j is allowed to enter the model where j is selected as the index of the maximum positive element of $KT2_j$. If at any stage negative regression coefficients are found, the predictor with the most negative b_j is dropped from the model. In this way predictors are added and dropped until the Kuhn-Tucker conditions are satisfied. Lawson & Hanson (1974) proved that this stepwise method always finds the model with minimal sums of squares. Their proof can be generalized to show that the minimum will be found irrespective of the initial model used.

McDonald & Diamond (1990) show that the Kuhn-Tucker values for generalized linear models are given by

$$[\mathbf{X}^T(\mathbf{y} - \boldsymbol{\mu})\{V(\boldsymbol{\mu})\,\partial\boldsymbol{\eta}/\partial\boldsymbol{\mu}\}^{-1}]$$

where $\boldsymbol{\mu}$ is the mean, $V(\boldsymbol{\mu})$ the variance function and $\boldsymbol{\eta}$ the linear predictor. These values can be calculated as follows

```
RKEEP  ITERATIVEWEIGHTS=iter; YADJUSTED=yadj;\
       LINEARPREDICTOR=lin
```

```
CALCULATE  kuhntuck = X * iter * (yadj - lin)
```

If the log-likelihood is strictly concave, as is usually the case for generalized linear models, the generalized Kuhn-Tucker conditions are necessary and sufficient and the iterative procedure finds the minimum of the constrained optimization problem. To increase numerical precision for generalized linear models, the procedure sets the TOLERANCE option of the RCYCLE directive to 1.0e−6.

Calculation of the Kuhn-Tucker conditions can be subject to considerable rounding errors. Therefore, before starting the stepwise procedure, the predictors are standardized. Moreover, the response and the fitted values are scaled identically before they are subtracted in the calculation of the Kuhn-Tucker values. Due to rounding errors, an aliased predictor may have a Kuhn-Tucker value slightly larger than 0 and may consequently enter the model. The Kuhn-Tucker values $KT2_j$ are therefore not tested against 0 but against the setting of the TOLERANCE option. Subsequent to the iterative procedure, aliased predictors, identified as having zero estimates and zero standard errors of estimates, are removed from the model. In the final fit the original non-standardized predictors are used.

Action with RESTRICT

Any restriction applied to vectors used in the regression model applies also to the results from RNONNEGATIVE.

References

Kennedy, W.J. & Gentle, J.E. (1980). *Statistical Computing*. Marcel Dekker, New York.

Lawson, C.L. & Hanson, R.J. (1974). *Solving Least Squares Problems*. Prentice & Hall, New York.

McDonald, J.W. & Diamond, I.D. (1990). On the fitting of generalized linear models with nonnegativity parameter constraints. *Biometrics*, **46**, 201-206.

Ramsay, J.O. (1988). Monotone regression splines in action. *Statistical Science*, **3**, 425-461.

See also

Genstat Reference Manual 1 Summary section on: Regression analysis.

ROBSSPM

Forms robust estimates of sum-of-squares-and-products matrices (P.G.N. Digby).

Options

PRINT = *string tokens*	Controls printed output (sspm, distances, weights, vcovariance, means, correlations, outliers); default * i.e. no output
B1 = *scalar*	The value from which the threshold distance is derived (see the Method Section); default 2
B2 = *scalar*	The value indicating the decline in weight as the distance of a unit above the threshold increases, (see the Method Section); default 1.25
MAXCYCLE = *scalar*	Maximum number of iterations; default 100
TOLERANCE = *scalar*	The minimum change in the average squared-weight that has to be achieved for the iterative process to converge; default 1.0^{-8}

Parameters

DATA = *pointers*	Supplies the set of variates in each datamatrix
SSPM = *SSPMs*	SSPM structure to contain the robust estimates of the sums of squares and products, the robust estimates of the means, and the sum of the weights for each datamatrix
DISTANCES = *variates*	To contain the Mahalanobis distances of the units from the mean
WEIGHTS = *variates*	To contain the weights used for each unit when forming the robust estimates
VCOVARIANCE = *symmetric matrices*	To contain the robust estimates of the matrices of variances and covariances
CORRELATIONS = *symmetric matrices*	This contains on output the correlations from the robust estimates of the variances and covariances

Description

ROBSSPM forms robust estimates of SSPMs, and the related variance-covariance and correlation matrices, using the method of Campbell (1980). This weights the units differentially so that those that are extreme, in a multivariate sense, contribute less to the calculated means and sums of squares and products. The extremeness of a unit is judged by its Mahalanobis distance from the estimated mean.

The input variates are specified, in a pointer, by the DATA parameter. They may be restricted or may contain some missing values, in which case the units concerned will be ignored.

Output is controlled by the PRINT option, with settings: sspm prints the estimated sums-of-squares-and-products, the estimated means, and the sum of the weights; distances prints the Mahalanobis distances for all the units, including any excluded by restrictions; weights prints the weights for all the units; vcovariance prints the estimated variance-covariance matrix; means prints the estimated means; correlations prints correlations derived from the variance-covariance matrix; outliers prints unit numbers, weights, and distances for outliers. By default there is no printed output.

If the outliers, weights or distances are to be printed then an appropriate summary of the number of units, number of outliers and so on will be printed too. The outlier information consists of the unit numbers, weights and Mahalanobis distances, printed across the page.

The weight given to each unit in forming the robust estimates is one if the unit's Mahalanobis distance from the mean is less than some threshold distance, and it decreases as the Mahalanobis distance increases above that threshold. The threshold and the form of the decrease in weight are controlled by options B1 and B2, which correspond to the corresponding quantities in the functions used by Campbell (1980), as explained in the Methods Section. By default, B1=2 and B2=1.25.

The estimation process is iterative, with the maximum number of iterations controlled by the MAXCYCLE option (default 100). It converges when the average change in the weights is less than some tolerance. The default tolerance is 1.0^{-8}, but this can be redefined by the TOLERANCE option. Lack of convergence usually indicates some problem with the data, perhaps that the threshold has been set too low.

Parameters SSPM, DISTANCES, WEIGHTS, VCOVARIANCE and CORRELATIONS allow the various components of the output to be saved.

Options: PRINT, B1, B2, MAXCYCLE, TOLERANCE.
Parameters: DATA, SSPM, DISTANCES, WEIGHTS, VCOVARIANCE, CORRELATIONS.

Method

Initial (unweighted) estimates of the means and sums of squares and products are formed from all the units, subject to any restriction on the data and excluding any units with missing values for any of the variates. From the estimates, Mahalanobis distances of the units from their means are calculated, and used to determine the weights for the units. The weights are then used to reform the SSPM structure, new distances are calculated, and so on. Convergence occurs when the average change in the derived weights is less than the defined tolerance.

The weight w of each unit is given by

$$w = 1 \qquad d \le t$$
$$W = (t/d) \times \exp(-0.5 \times (d-t)^2 / \mathrm{B2}^2) \qquad d > t$$

where t, the threshold distance, is given by

$$t = \sqrt{v} + \mathrm{B1} / \sqrt{2}$$

and v is the number of means.

As explained by Campbell (1980), under Fisher's square root approximation, B1 equates to a percentage point of the standard Gaussian distribution.

Campbell (1980) regards three possibilities as potentially most useful. If B1 is infinite, the usual (non-robust) estimates are obtained. With B1=2 and B2 infinite, the weight decreases inversely with distance ($w=t/d$); this can be obtained in the procedure by setting B2 to a missing value. Finally, there is the combination used as a default by ROBSSPM, namely B1=2 and B2=1.25.

Action with RESTRICT

If the DATA variates are restricted only the units not excluded by the restriction will be used in the estimation process. However, Mahalanobis distances will be formed for all units other than those where any of the variates is missing.

Reference

Campbell, N.A. (1980). Robust procedures in multivariate analysis I: robust covariance estimation. *Applied Statistics*, **29**, 231-237.

See also
Directive: FSSPM.
Procedures: FVCOVARIANCE, MPOLISH, TUKEYBIWEIGHT.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Multivariate and cluster analysis.

RPAIR

Gives t-tests for all pairwise differences of means from a regression or generalized linear model (J.T.N.M. Thissen & P.W. Goedhart).

Options

PRINT = *string tokens*	What to print (differences, sed, tvalues, tprobabilities); default diff, sed, tval
SORT = *string token*	Whether to sort the means into ascending order (no, yes); default no
COMBINATIONS = *string token*	Which combinations of factors in the current model to include (full, present, estimable); default esti (similar to the PREDICT directive)
ADJUSTMENT = *string token*	Type of adjustment with linear regression models (marginal, equal); default marg (similar to the PREDICT directive)
WEIGHTS = *table*	Weights classified by some or all standardizing factors; default * (similar to the PREDICT directive)
METHOD = *string token*	Method of forming margin (mean, total); default mean (similar to the PREDICT directive)
ALIASING = *string token*	How to deal with aliased parameters (fault, ignore); default faul (similar to the PREDICT directive)
SAVE = *identifier*	Specifies save structure of model to display; default * (i.e. that of the latest model fitted)

Parameters

TREATFACTORS = *pointers*	Each pointer contains a list of treatment factors classifying the table of means to be compared (the right-most factor changes fastest, then the second from the right, etc.); this parameter must be set
LABELS = *texts*	Structures containing strings to label rows (and columns) of the symmetric matrices of pairwise differences etc; the length of the text must equal the product of the numbers of factor levels as implied by the factor list in the TREATFACTORS pointer
NEWLABELS = *texts*	To save the row labels of the DIFFERENCES, SED, TVALUES and TPROBABILITIES matrices
DIFFERENCES = *symmetric matrices*	To save pairwise differences (treatment means on the diagonal)
SED = *symmetric matrices*	To save standard errors of the pairwise differences (missing values on the diagonal)
TVALUES = *symmetric matrices*	To save t-values (missing values on the diagonal)
TPROBABILITIES = *symmetric matrices*	To save t-probabilities (missing values on the diagonal)

Description

When analysing a (non-orthogonal) analysis of variance model or a generalized linear model (GLM) with the regression directives FIT, ADD etc., effects of factors in the model and their interactions if required, may be assessed from a suitable analysis of variance (deviance) table. With the PREDICT directive tables of estimated means and their standard errors can be obtained, but not standard errors of differences of means. The RPAIR procedure provides additional

information on such tables by calculating t-values and corresponding two-sided t-probabilities for tests of all pairwise differences of means.

The t-statistics used are based on the residual variance (deviance) and its degrees of freedom from the current regression model. However, if the DISPERSION option of the MODEL directive has been set to a numerical value (as is by default the case with a GLM with binomial, poisson or multinomial distribution), the degrees of freedom are set to 10000, approximating to the normal distribution.

It is assumed that the MODEL statement for the regression has defined only one response variate.

The TREATFACTORS parameter must be set to a pointer containing a list of factors classifying the table of means which are to be compared.

The PRINT option controls the output. By default a symmetric matrix of pairwise differences of means is printed with the means themselves down the diagonal. With a GLM these means and their pairwise differences are always calculated on the linear scale. The corresponding symmetric matrices of standard errors and of t-values are printed by default too.

The matrix rows (and columns) are ordered such that the right-most factor changes fastest, then the second from the right, etc. This default order can be changed by setting the SORT option to yes, in which case rows and columns of all matrices are rearranged to put the means on the diagonal of the matrix of differences into ascending order.

The LABELS parameter can be used to label the rows and columns of the matrices, which are then taken in default order. When the LABELS parameter has not been set and the TREATFACTORS pointer contains just one factor, by default the labels or levels of the factor are used for labeling; when the pointer contains more than one factor, the default row (and column) labels are combinations of factor settings indicated by the first letter of the factor identifier followed by an ordinal level.

The DIFFERENCES, SED, TVALUES and TPROBABILITIES parameters can be used to save the output. The row labels of these matrices can be saved through the NEWLABELS parameter.

The COMBINATIONS, ADJUSTMENT, WEIGHTS, METHOD, ALIASING and SAVE options are as in the PREDICT directive.

Options: PRINT, SORT, COMBINATIONS, ADJUSTMENT, WEIGHTS, METHOD, ALIASING, SAVE.
Parameters: TREATFACTORS, LABELS, NEWLABELS, DIFFERENCES, SED, TVALUES, TPROBABILITIES.

Method

The procedure uses the PREDICT directive to save a table of predictions and corresponding variance-covariance matrix. With a GLM the setting of option BACKTRANSFORM of the PREDICT directive is always none.

Action with RESTRICT

Any restrictions applied to vectors used in the regression apply also to the results from RPAIR.

See also

Procedures: ALLDIFFERENCES, AMCOMPARISON, AUMCOMPARISON, PAIRTEST, PPAIR.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RPARALLEL

Carries out analysis of parallelism for nonlinear functions; synonym FITPARALLEL (R.C. Butler).

Options

PRINT = *string tokens*	What to print (model, summary, accumulated, estimates, correlations, fittedvalues, monitoring); default mode, summ, accu, esti
CALCULATION = *expression structures*	Calculation(s) involving explanatory variate; no default (must be set)
METHOD = *string token*	Which models to fit (singleline, constantsseparate, linearseparate, nonlinearseparate); default nonl
CONSTANT = *string token*	How to treat constant (estimate, omit); default esti

Parameters

X = *variates*	Explanatory variate; must be set
GROUPS = *factors*	Grouping factor for data; must be set
RESULTS = *pointers*	To save results from model nonlinearseparate, if fitted; should be set only if METHOD=nonl

Description

This procedure mimics the testing of parallelism which can be carried out using FITCURVE, but caters for any nonlinear functions or sums of nonlinear functions. FITCURVE can be used successively to fit four models with varying degrees of parallelism between curves fitted to different levels of a grouping factor where each curve is the same function of an explanatory variate, but only has a limited choice of ten curves. RPARALLEL will fit these same four models for any function (or sum of functions) that the user defines.

Definitions – Take a response variate Y and explanatory variate X, and functions $f(X; \theta)$ to describe the relationship between them, where θ represents the parameters of f. Levels of a factor are denoted by i, and j denotes values of X for each level of the factor. A_i, B_i and θ_i are the constant, slope, and nonlinear parameters for factor level i respectively.

Single Line – the same model with the same parameters is fitted to all levels of a factor

$Y_{ij} = A + B \times f(X_{ij}; \theta)$ (cf. FITCURVE X)

Constants Separate – different values of the constant A are fitted for each level of a factor.

$Y_{ij} = A_i + B \times f(X_{ij}; \theta)$ (cf. FITCURVE *factor* + X)

Linear Separate – different values of constant A and "slope" B are fitted for each level of a factor.

$Y_{ij} = A_i + B_i \times f(X_{ij}; \theta)$ (c.f. FITCURVE [non=c] *factor* * X)

Nonlinear Separate – different values of all parameters are fitted for each level of a factor.

$Y_{ij} = A_i + B_i \times f(X_{ij}; \theta i)$ (c.f. FITCURVE [non=s] *factor* * X)

When sums of functions are fitted by RPARALLEL, the models are similar, but each includes a set of B_i's, f's and θ_i's, one set for each function.

The four models are fitted in a single call of the procedure (unlike FITCURVE) so that an accumulated analysis of variance can be compiled. The dependent variate and the parameters of the functions to be fitted are defined in the usual way using the MODEL and RCYCLE directives, and the explanatory variate and the grouping factor for the data are defined using the X and GROUPS parameters of the procedure. The constant term in the fitted equation can be omitted or estimated by setting the CONSTANT option appropriately. The METHOD option determines the most complex of the four models to be fitted, with all simpler models also fitted. For example,

if METHOD=linearseparate, the single-line and constants-separate models are also fitted. The CALCULATION option is set to an expression or list of expressions to define the form of the function to be fitted, as for FITNONLINEAR.

Printed output is controlled using the PRINT option, with monitoring, summary analysis of variance, estimates, and correlations being printed for each model fitted, but fitted values and accumulated analysis of variance being printed for the most complex model only. The results of fitting the complex model can be saved using RKEEP, providing this model is not nonlinearseparate; for that case the results can be saved only by setting the parameter RESULTS. This forms a pointer whose elements are labelled by the names of their contents: FITTEDVALUES, RESIDUALS, ESTIMATE, SE, DEVIANCE, DF. If RDISPLAY is used after METHOD=nonlinearseparate has been fitted, only the results of fitting to the last level of the GROUPS factor will be displayed.

Options: PRINT, CALCULATION, METHOD, CONSTANT.
Parameters: X, GROUPS, RESULTS.

Method

The single-line and constants-separate models are fitted using FITNONLINEAR in a similar manner to FITCURVE in a similar situation, but saving the results for later use in an Accumulated Analysis of Variance. The linear-separate model (i.e. Parallel Lines) is fitted by setting up expressions (in a pointer f) which calculate one dummy variable for each factor-level by function combination, which are fitted using FITNONLINEAR as follows:

```
FITNONLINEAR [CALCULATION=f] dummy[][] + GROUPS
```

The final Separate nonlinear model is fitted using a loop which restricts the data to each level of the factor in turn, and saves the sums of squares and estimates found for each subset. The final residual sums of squares for the whole model is calculated as the sum of the residual sums of squares for the individual parts, and the standard errors for the estimates are calculated using this and values saved from the matrix of second derivatives (INVERSE in RKEEP).

This method is described more fully by Butler & Brain (1990).

Action with RESTRICT

Restrictions of X or GROUPS are ignored, but the analysis is carried out on any restricted set of the dependent variate defined by the MODEL statement.

Reference

Butler, R.C. & Brain, P. (1990). Parallelism in non-linear models. *Genstat Newsletter*, **25**, 40-46.

See also

Directives: FITCURVE, FITNONLINEAR.
Procedures: NLAR1, NLCONTRASTS.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RPERMTEST

Does random permutation tests for regression or generalized linear model analyses (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (probability, accumulated, summary, critical); default prob
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of variates and/or factors in the terms to be fitted; default 3
NTIMES = *scalar*	Number of permutations to make; default 999
BLOCKSTRUCTURE = *formula*	Model formula defining any blocking to consider during the randomization; default none
EXCLUDE = *factors*	Factors in the block formula whose levels are not to be randomized
SEED = *scalar*	Seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically
†SUMMARY = *pointer*	Saves the summary analysis-of-variance (or deviance) table with permutation probabilities and critical values
†ACCUMULATED = *pointer*	Saves the accumulated analysis-of-variance (or deviance) table with permutation probabilities and critical values
†BINMETHOD = *string token*	How to permute binomial data (individuals, units; default indi

Parameter

TERMS = *formula*	List of explanatory variates and factors, or model formula, defining the model to fit

Description

In regression analyses, random permutation tests provide an alternative to using the F probabilities, printed for variance ratios in summary or accumulated analysis of variance tables, when the assumptions of the analysis are not satisfied. These assumptions can be assessed by studying the residual plots produced by RCHECK. In particular, the use of the F distribution to calculate the probabilities is based on the assumption that the residuals from each stratum have Normal distributions with equal variances, and so the histogram of residuals produced by RCHECK should look reasonably close to the Normal, bell-shaped curve. Experience shows the analysis is robust to small departures from Normality. RPERMTEST can be useful if the histogram looks very non-Normal. You can also use RPERMTEST to generate probabilities for deviances or deviance ratios in generalized linear models, instead of using the customary chi-square or F distributions (which are justified by asymptotic theory).

Before using RPERMTEST, you need to give a MODEL statement to define the y-variate and so on, as usual for a regression or generalized model. The terms to fit in the regression model are specified by the TERMS parameter of RPERMTEST. As in the FIT directive, this can supply a list of variates for a simple or multiple linear regression, or a model formula with variates and/or factors for more complicated models. As usual, the CONSTANT option indicates whether or not to fit the constant, and the FACTORIAL option sets a limit as usual on the number of variates and/or factors in each of the terms generated from a TERMS formula.

The NTIMES option defines how many random permutations to perform; by default there are

999 (as well as the "null" permutation where the data keep their original order). The SEED option allows you to specify the seed to use for the random-number generator that is used to construct them. The default, SEED=0, continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, it initializes the seed automatically. If NTIMES exceed the maximum possible number of permutations for the data, an "exact" test is performed in which every permutation is used once. This is feasible only for small datasets. There are *n*! (n factorial) permutations of *n* units: 3!=6, 4!=24, 5!=120, 6!=720, 7!=5040, 8!=40320, and so on.

If the regression is being used to analyse a designed experiment, you may need to use the BLOCKSTRUCTURE option to specify a block model to define how to do the randomization. The EXCLUDE option can then restrict the randomization so that one or more of the factors in the block model is not randomized. See the RANDOMIZE directive for further details.

The BINMETHOD option controls how the permutations are done for binomial data. The original data set will have contained a set of units, each recording a number of "successes" obtained from an observed number of individuals. The default, and recommended, method is to expand the data set to contain individuals themselves, and permute these. Alternatively, you can set BINMETHOD=units if you prefer to permute the units as a whole instead.

The probabilities are determined from the distribution of the statistics of interest, over the permuted datasets. In an ordinary regression, the statistics are the variance ratios from the summary-of-analysis or accumulated-analysis-of-variance tables. In generalized linear models they will be deviances when the dispersion is fixed, or deviance ratios when it is estimated (as defined by the DISPERSION option of the MODEL directive).

Output is controlled by the PRINT option, with settings:

probability	to print the probability for the whole regression model;
summary	to print the summary-of-analysis table with the usual probability for the regression model replaced by the probability from the permutation test;
accumulated	to print the accumulated analysis of variance or deviance table with the usual probabilities replaced by those from the permutation test;
critical	to accompany the summary or accumulated tables by a table giving estimated critical values for each of the statistics.

The SUMMARY and ACCUMULATED options can save the summary and accumulated table, respectively. They are saved in pointers with a variate or text for each of its columns (source, d.f. etc). The probability variate contains the probabilities from the permutation test, and there are three additional variates to save the critical values.

Options: PRINT, CONSTANT, FACTORIAL, NTIMES, BLOCKSTRUCTURE, EXCLUDE, SEED, SUMMARY, ACCUMULATED, BINMETHOD.
Parameter: TERMS.

Method

RPERMTEST uses RANDOMIZE to perform the permutations, taking account of any block structure of the date. The model is fitted, for each data set using either FIT or FITINDIVIDUALLY. (FITINDIVIDUALLY is needed if the accumulated table is required for a generalized linear model.) The ACCUMULATED and SUMMARY options of RKEEP are used to save the information from each analysis, and the QUANTILES function is used to calculate the critical values.

Action with RESTRICT

RPERMTEST takes account of any restrictions on any of the y-variates or x-variates or factors in

the model.

See also
Procedures: APERMTEST, CHIPERMTEST, FEXACT2X2.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RPHCHANGE

Modifies a proportional hazards model fitted by RPHFIT (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, loglikelihood); default mode, summ, esti
METHOD = *string token*	How to change the model (add, drop, switch); default add
POOL = *string token*	Whether to pool terms in the accumulated summary generated by the fit

Parameter

TERMS = *formula*	Model specifying the change

Description

This procedure allows you to modify the contents of a proportional hazards model that has been fitted by procedure RPHFIT. The change to the model is specified by the TERMS parameter. The setting of the METHOD option specifies how the model is to be changed:

add	adds the terms specified by the TERMS parameter to the fitted model;
drop	drops those terms from the fitted model; and
switch	drops any terms specified by the TERMS parameter that are already in the fitted model, and adds those that are not (i.e. this operates similarly to the SWITCH directive).

The default is METHOD=add. Note, though, that any term that is to be added must have been included in the full model specified by the MAXIMALMODEL option of RPHFIT. By default the changes are made individually, one term at a time, so that each one will have its own line in an accumulated analysis of deviance. However, you can set option POOL=yes to make them all at once.

The PRINT option controls printed output with similar settings to those of the FIT directive, except that there is an extra setting loglikelihood to print −2 times the log-likelihood and the number of degrees of freedom in the model after the change. The deviance produced for the terms in the regression model can be assessed using chi-square distributions as usual, but the residual deviance is not usable as the maximal model assumed by the generalized linear models method is inappropriate. So, the residual line is suppressed in the summary and accumulated analysis of deviance.

Options: PRINT, METHOD, POOL.
Parameter: TERMS.

Method

Further details of the method used here (and by RPHFIT) can be found in Aitkin *et al*. (1989).

Action with RESTRICT

None of the vectors must be restricted (and any restrictions will have been cancelled by RPHFIT).

Reference

Aitkin, M., Anderson, A., Francis, B. & Hinde, J. (1989). *Statistical Modelling in GLIM*. Oxford

University Press.

See also

Procedures: KAPLANMEIER, RLIFETABLE, RPHFIT, RPHDISPLAY, RPHKEEP, RPROPORTIONAL, RSTEST, RSURVIVAL.

Genstat Reference Manual 1 Summary section on: Survival analysis.

RPHDISPLAY

Prints output for a proportional hazards model fitted by RPHFIT (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, loglikelihood); default mode, summ, esti

No parameters

Description

This procedure allows you to display further output for a proportional hazards models that has been fitted by procedure RPHFIT. The output is controlled by PRINT option with similar settings to those of the FIT directive, except that there is an extra setting loglikelihood to print −2 times the log-likelihood and the number of degrees of freedom in the model. The deviance produced for the terms in the regression model can be assessed using chi-square distributions as usual, but the residual deviance is not usable as the maximal model assumed by the generalized linear models method is inappropriate. So, the residual line is suppressed in the summary and accumulated analysis of deviance.

Option: PRINT.

Method

Further details of the method used by RPHFIT can be found in Aitkin *et al*. (1989).

Action with RESTRICT

None of the vectors must be restricted (and any restrictions will have been cancelled by RPHFIT).

Reference

Aitkin, M., Anderson, A., Francis, B. & Hinde, J. (1989). *Statistical Modelling in GLIM*. Oxford University Press.

See also

Procedures: KAPLANMEIER, RLIFETABLE, RPHFIT, RPHCHANGE, RPHKEEP, RPROPORTIONAL, RSTEST, RSURVIVAL.

Genstat Reference Manual 1 Summary section on: Survival analysis.

RPHFIT

Fits a proportional hazards model to survival data as a generalized linear model (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, loglikelihood); default mode, summ, esti
MAXIMALMODEL = *formula*	Defines the full model to explore (using RPHCHANGE); default uses the model defined by the TERMS parameter
SUBJECTS = *factor*	Subject corresponding to each observation
TIMES = *factor* or *variate*	Time of each observation
CENSORED = *variate*	Contains the value 1 for censored observations, otherwise 0; if unset it is assumed that there is no censoring
OFFSET = *variate*	Offset to include in the model
POOL = *string token*	Whether to pool terms in the accumulated summary generated by the fit

Parameter

TERMS = *formula*	Model to fit

Description

The data for RPHFIT consist of a set of subjects observed at one or more times. The final time is usually at the time of death (or failure), otherwise (if the subject survives the trial) the observation is said to be censored. The CENSORED option can be used to specify a variate with an entry for each subject containing one when there is censoring, otherwise zero. If this is not specified, it is assumed that there is no censoring. The SUBJECTS option can specify a factor to indicate the subject corresponding to each observation; this can be omitted if there is only one observation per subject. The time at which each observation was made is specified by the TIME option, in either a factor or a variate.

The model to be fitted is specified using the TERMS parameter. You can modify the model later by using procedure RPHCHANGE. If you intend to use RPHCHANGE to include additional model terms, you should use option MAXIMALMODEL of RPHFIT to define the largest model that you may want to consider (this option acts similarly to the TERMS directive in ordinary generalized linear modelling). You can display further output using procedure RPHDISPLAY, and save information using procedure RPHKEEP.

The proportional hazards model (Cox 1972) makes the assumption that the subjects have a baseline hazard function which is modified proportionally by treatments and covariates. In RPHFIT it is assumed that the survival times follow a piecewise exponential distribution (Breslow 1974). This partitions the time axis using a set of discrete cut-points a_i, and assumes a constant baseline hazard γ_i between each one. This corresponds to an exponential distribution with mean $1/\gamma_i$ for the survival times (in the absence of treatments) within each time interval. A cut-point is defined at every time that a death (or failure) occurs and, if the covariates or treatments vary with time, also at every time when the subjects are observed.

To fit a proportional hazards model as a generalized linear model, the x-variates (i.e. covariates) and factors must be expanded so that, for each subject, there is a unit for every time interval up to the last one during which the subject was observed. If (as usually happens) the subject was not observed at every cutpoint, the covariates and treatments are taken to be constant during the intervals between the times of the observations. RPHFIT automatically produces the expanded sets of values (using procedure RPHVECTORS). These replace the original values while

RPHFIT is fitting and displaying the model. The original values are then reinstated before exit from the procedure, unless a fault is generated e.g. from the regression directives FIT &c. You can call RPHVECTORS directly if you do want to obtain the expanded values. Alternatively, procedure RPHKEEP can save the index variate that is used to construct them.

The y-variate used within the generalized linear model is an indicator that takes the value 0 if the subject was still surviving within the time interval concerned, otherwise it has the value 1. The model also contains an offset representing the log of the exposure time within each interval. Any additional offset can be specified, if required, using the OFFSET option. (These two variates are also obtainable from RPHKEEP.)

The PRINT option controls printed output with similar settings to those of the FIT directive, except that there is an extra setting loglikelihood to print −2 times the log-likelihood. The deviance produced for the terms in the regression model can be assessed using chi-square distributions as usual, but the residual deviance is not usable as the maximal model assumed by the generalized linear models method is inappropriate. So, the residual line is suppressed in the summary and accumulated analysis of deviance. By default the terms in the model are fitted individually so that they will all have their own lines in an accumulated analysis of deviance. However, you can set option POOL=yes to fit them all at once.

Options: PRINT, MAXIMALMODEL, SUBJECTS, TIMES, CENSORED, OFFSET, POOL.
Parameter: TERMS.

Method

The expanded sets of values for the variates and factors in the model are formed using procedure RPHVECTORS, together with the response and offset variates that are needed. Further details of the method can be found in Aitkin *et al*. (1989).

Action with RESTRICT

None of the vectors must be restricted, and any restrictions will be cancelled.

References

Aitkin, M., Anderson, A., Francis, B. & Hinde, J. (1989). *Statistical Modelling in GLIM*. Oxford University Press.

Breslow. N. (1974). Covariance analysis of censored survival data. *Biometrics*, **30**, 89-99.

Cox, D.R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society Series B*, **34**, 187-220.

See also

Procedures: KAPLANMEIER, RLIFETABLE, RPHCHANGE, RPHDISPLAY, RPHKEEP, RPHVECTORS, RPROPORTIONAL, RSTEST, RSURVIVAL.

Genstat Reference Manual 1 Summary section on: Survival analysis.

RPHKEEP

Saves information from a proportional hazards model fitted by RPHFIT (R.W. Payne).

Options

RESIDUALS = *variate*	Saves the standardized residuals
FITTEDVALUES = *variate*	Saves the fitted values
ESTIMATES = *variate*	Saves estimates of the parameters
SE = *variate*	Saves standard errors of the estimates
RESPONSE = *variate*	Saves the response variate defined for the generalized linear model
OFFSET = *variate*	Saves the offset variate defined for the generalized linear model
INDEX = *variate*	Index variate used to produce the expanded covariates and factors
RISKSET = *factor*	Saves the expanded time factor
_2LOGLIKELIHOOD = *scalar*	Saves −2 × log-likelihood for the fitted model
DFTERMS = *scalar*	Saves the number of d.f. in the model specified by TERMS

No parameters

Description

This procedure allows you to copy information into Genstat data structures from a proportional hazard model that has been fitted by procedure RPHFIT. You do not need to declare the structures in advance; Genstat will declare them automatically to be of the correct type and length.

The RESIDUALS and FITTEDVALUES options save the standardized residuals and the fitted values. The ESTIMATES and SE options save the parameter estimates and their standard errors. The RESPONSE and OFFSET options save the response variate and the offset variate that have been defined for the generalized linear model. The INDEX variate saves the variate of indexes used to construct the expanded x-variates and factors from original variates and factors of the model. The RISKSET option saves a variate indicating the time interval corresponding to each of their units. Finally, the _2LOGLIKELIHOOD option saves −2 times the log-likelihood, and the DFTERMS option saves the number of degrees of freedom in the model specified by TERMS.

Options: RESIDUALS, FITTEDVALUES, ESTIMATES, SE, RESPONSE, OFFSET, INDEX, RISKSET, _2LOGLIKELIHOOD, DFTERMS.
Parameters: none.

Method

The log-likelihood is calculated as described by Aitkin *et al*. (1989). The response variate and offset are recovered from a workspace structure that is defined by RPHFIT to hold details of the model. The other information is saved using RKEEP (which can also be used to save additional relevant output).

Reference

Aitkin, M., Anderson, A., Francis, B. & Hinde, J. (1989). *Statistical Modelling in GLIM*. Oxford University Press.

See also

Procedures: KAPLANMEIER, RLIFETABLE, RPHFIT, RPHCHANGE, RPHDISPLAY, RPHVECTORS, RPROPORTIONAL, RSTEST, RSURVIVAL.

Genstat Reference Manual 1 Summary section on: Survival analysis.

RPHVECTORS

Forms vectors for fitting a proportional hazards model as a generalized linear model (R.W. Payne).

Options

SUBJECTS = *factor*	Subject corresponding to each observation
TIMES = *factor* or *variate*	Time of each observation
CENSORED = *variate*	Contains the value 1 for censored observations, otherwise 0; if unset it is assumed that there is no censoring
RESPONSE = *variate*	Response variate for the generalized linear model
OFFSET = *variate*	Offset variate
INDEX = *variate*	Mapping variate used to produce the expanded variables
NEWSUBJECTS = *factor*	Expanded subjects factor
NEWTIMES = *factor* or *variate*	Expanded times factor
NEWOFFSET = *variate*	Offset variate for fitting the proportional hazards model

Parameters

X = *variates* or *factors*	Lists the x-variables that are to be expanded
NEWX = *variates* or *factors*	Identifiers to store the expanded x-variables; if no NEWX is specified, the expanded values overwrite the original values of X

Description

The data for a proportional hazards model consist of a set of subjects observed at one or more times. The final time is usually at the time of death (or failure), otherwise (if the subject survives the trial) the observation is said to be censored. The CENSORED option of RPHVECTORS can be used to specify a variate with an entry for each subject containing one when there is censoring, and zero when there is no censoring. If this is not specified, it is assumed that there is no censoring. The SUBJECTS option can specify a factor to indicate the subject corresponding to each observation; this can be omitted if there is only one observation per subject. The time at which each observation was made must be specified by the TIMES option, in either a factor or a variate.

The proportional hazards model (Cox 1972) can be fitted in Genstat using the RPHFIT procedure. To fit this as a generalized linear model, the vectors in the model must be expanded so that, for each subject, there is a unit for every time interval up to the last one during which the subject was observed. If (as usually happens) the subject was not observed at every cutpoint, the covariates and treatment factors are taken to be constant during the intervals between the times of the observations.

RPHVECTORS is used by RPHFIT to produce these expanded vectors, and is made available as a Library procedure in its own right to allow you to program your own proportional hazards analyses. The variates and factors to be expanded are specified using the X parameter, and the NEWX parameter can specify identifiers for the each expanded variate or factor. If NEWX is not specified, the expanded version will replace the original one. The INDEX option allows the variate of indexes used to produce the expanded vectors from the original ones to be saved. The NEWTIMES option can save an expanded factor indicating the time interval corresponding to each unit of the expanded vectors, and the NEWSUBJECTS option can save an expanded factor indicating the subject corresponding to each of their units. The RESPONSE option saves a variate containing one or zero according to whether or not the relevant subject "responded" (i.e. died or failed) during the time period corresponding to each unit of the expanded vectors.

The generalized linear model also needs an offset variate, containing the log of the exposure

time corresponding to each units of the expanded variables. This can be saved using the NEWOFFSET option. This will incorporate any additional offset, which can be specified by the OFFSET option.

Options: SUBJECTS, TIMES, CENSORED, RESPONSE, OFFSET, INDEX, NEWSUBJECTS, NEWTIMES, NEWOFFSET.
Parameters: X, NEWX.

Method
RPHVECTORS uses the standard Genstat manipulation commands.

Action with RESTRICT
None of the vectors must be restricted, and any restrictions will be cancelled.

Reference
Cox, D.R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society B*, **34**, 187-220.

See also
Procedure: RPHFIT.
Genstat Reference Manual 1 Summary section on: Survival analysis.

RPOWER

Calculates the power (probability of detection) for regression models (R.W. Payne).

Options

PRINT = *string token*	Prints the power (power); default powe
TERMS = *formula*	Specifies the terms (x-variates, factors or model terms) to be fitted in the analysis when the responses to be detected are specified by the RESPONSE parameter
FACTORIAL = *scalar*	Limit on the number of factors or variates in a model term generated from TERMS; default 3
PROBABILITY = *scalar*	Significance level at which the response is required to be detected (assuming a one-sided test); default 0.05
TMETHOD = *string token*	Type of test to be made (onesided, twosided, equivalence, noninferiority, fratio, chisquare); default ones
SAVE = *rsave*	Regression save structure to provide the information about the regression model

Parameters

RESPONSE = *variates*	Variate of fitted values calculated using regression parameters of the size to be detected; default * implies that the information is to be taken from a regression save structure
RDF = *scalars*	Number of residual degrees of freedom; if unset, this is obtained from the analysis of RESPONSE or from the regression save structure
RSS = *scalars*	Anticipated residual sum of squares; if unset, this is obtained from the analysis of RESPONSE or from the regression save structure
POWER = *scalars* or *variates*	Saves the power

Description

When planning a regression study, it can be useful to know how likely a response is to be detected. This probability of detection, known as the *power* of the study with respect to the response of interest, helps to determine whether the study is sufficiently large or accurate to achieve its purpose. RPOWER can consider any of the regression models that Genstat can analyse, and can calculate the power either for the assessment of the whole model (as represented by the regression sum of squares), or the assessment of individual parameters in the regression model.

To determine the power, you need to define the terms (x-variates, factors or model terms) to be fitted in the regression, and specify the anticipated amount of residual variability. This is most easily done by taking the analysis of a data set similar to the one to be used in the new study. To do this, you should analyse the earlier set of data with the regression directives in the usual way. Provided you do not fit any other regressions in the interim, RPOWER will pick up the information automatically from the save information held within Genstat about the most recent regression analysis. Alternatively, you can save the information explicitly in a regression save structure, by setting the SAVE option of MODEL, and then use this same save structure as the setting of the SAVE option of RPOWER.

Using a save structure allows you to specify any regression model, including any nonlinear or generalized linear model. If you merely have an ordinary linear regression model, you can set up the whole process within RPOWER if you prefer. The terms to be fitted in the model can be specified using the TERMS option of RPOWER. The setting can be a list of x-variates or a model

formula, as in the setting of the parameter of the FIT directive. The FACTORIAL option, as in FIT, sets a limit on the number of factors or variates in each of the terms generated from a model formula. The constant is included automatically. (So, if you want to omit the constant and fit a regression through the origin, you should specify a save structure instead.) The RESPONSE parameter then supplies a y-variate calculated with regression parameters set to the sizes of responses to be detected. For example, if we have a simple linear regression with x-variate X and wish to be able to detect a regression coefficient of size at least 2.5, we would calculate the response as

```
response = 2.5 * X
```

If we also wanted to check that we can detect a constant (or intercept) of size 3, the calculation would become

```
response = 2.5 * X + 3
```

RPOWER analyses the RESPONSE variate using the model specified by TERMS in order to obtain the values required to be detected for the various regression parameters.

The anticipated residual sum of squares can be specified by the RSS parameter, and the residual degrees of freedom by the RDF parameter. If these are not set, RPOWER takes the values from the regression save structure (if this is how the model has been specified) or from the analysis of the RESPONSE variate.

The PROBABILITY option specifies the significance level that you intent to use in the analysis to detect a response; the default is 0.05 (i.e. 5%). By default, RPOWER assumes that individual regression parameters are to be assessed by a one-sided t-test, but you can set option TMETHOD=twosided to assess them by a two-sided t-test instead.

Other settings of TMETHOD enable you to test individual parameters for equivalence or for non-inferiority. With equivalence (TMETHOD=equivalence), RESPONSE defines a threshold below which the parameter can be assumed to be equivalent to no response. If the future estimate of the parameter is b and the threshold is b_{lim}, the null hypothesis for equivalence is that either

$$b \leq -b_{lim}$$

or

$$b \geq b_{lim}$$

with the alternative hypothesis that they are equivalent, i.e.

$$-b_{lim} < b < b_{lim}$$

With non-inferiority (TMETHOD=noninferiority), the null hypothesis becomes

$$b \geq -b_{lim}$$

(which represents a simple one-sided t-test).

You can also set TMETHOD=fratio, to assess the power of the F test for the regression in the summary analysis of variance (or deviance); this is an overall test for the whole regression model. Alternatively, if RPOWER is using a save structure from the analysis of a generalized linear model with a non-Normal distribution, you can set TMETHOD=chisquare to assess the power of a chi-square test on the deviance due to the regression model (see Section 3.5 of Part 2 of the *Guide to the Genstat Command Language*).

The POWER parameter can save the power, in a scalar if TMETHOD is set to fratio or chisquare; otherwise in a variate. They are printed by default, but you can set option PRINT=* to stop this.

Options: PRINT, TERMS, FACTORIAL, PROBABILITY, TMETHOD, SAVE.
Parameters: RESPONSE, RDF, RSS, POWER.

Method

The standard error of the i'th regression parameter is

```
SQRT( IMAT$[i] * RSS / RMS )
```

where IMAT$[i] is the value in the i^{th} diagonal element of the inverse matrix, obtainable using the INVERSE parameter of RKEEP. The sum of squares (or the deviance) due to the regression and the corresponding number of degrees of freedom are obtainable by using RKEEP to save the total sum of squares and number of degrees of freedom, and those for the residual. The required powers can then be calculated using Genstat's probability functions for the F, chi-square and t distributions as appropriate.

See also
Procedures: APOWER, VPOWER.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RPROPORTIONAL

Fits the Cox proportional hazards model to survival data (A.I. Glaser & R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (estimates, vcovariance, residuals, survivor, _2loglikelihood); default esti, _2lo
FACTORIAL = *scalar*	Sets a limit on the number of factors in the terms formed from the TERMS formula
TIMES = *factor* or *variate*	Time of each observation
CENSORED = *variate*	Contains the value 1 for censored observations, otherwise 0; if unset it is assumed that there is no censoring
OFFSET = *variate*	Offset to include in the model
BLOCKS = *factor*	Blocking factor defining groups of observations with different baseline hazard functions
INITIAL = *scalar* or *variate*	Initial values for the parameters in the model
RESIDUALS = *variate*	Saves the Cox-Snell residuals
ESTIMATES = *variate*	Saves the parameter estimates
SE = *variate*	Saves standard errors of the estimates
VCOVARIANCE = *symmetric matrix*	Saves the variance-covariance matrix of the estimates
_2LOGLIKELIHOOD = *scalar*	Saves −2 × log-likelihood for the fitted model
DFTERMS = *scalar*	Saves the number of d.f. in the model specified by TERMS
SURVIVOR = *variate* or *matrix*	Saves estimates of the survivor function, in a variate if BLOCKS is unset, otherwise in a matrix with a column for each block
EXIT = *scalar*	Exit code, set to zero if the fit was successful
MAXCYCLE = *scalar*	Maximum number of iterations to use; default 50
TOLERANCE = *scalar*	Defines the convergence criterion; default 0.000001

Parameter

TERMS = *formula*	Defines the model to fit

Description

RPROPORTIONAL fits the Cox proportional hazards model by a direct maximization of the likelihood, using NAG algorithm G12BAF. This is much more efficient for large data sets than the alternative method, used in procedure RPHFIT, which fits a generalized linear model to an expanded data set (see RPHFIT for details).

The data for RPROPORTIONAL consist of a time observation made on each of a set of subjects. Usually, this will be the time of death (or failure). Alternatively, an observation may be *censored*; the time will then be the time at which the subject left the trial (prior to failure or death). If you have censored data, you must use the CENSORED option to supply a variate with the value one in the censored observations, and zero elsewhere. The times are supplied by the TIME option, in either a factor or a variate.

The proportional hazards model (Cox 1972) makes the assumption that the subjects have a baseline hazard function which is modified proportionally by treatments and covariates. In RPROPORTIONAL it is assumed that the survival times follow a piecewise exponential distribution. This partitions the time axis using a set of discrete cut-points a_i, and assumes a constant baseline hazard λ_i between each one. This corresponds to an exponential distribution

with mean $1/\lambda_i$ (in the absence of treatments) for the survival times within each time interval. A cut-point is defined at every time that a death or a censored observation occurs. You can supply a factor, using the BLOCKS option, to define groupings of subjects. The baseline hazards are then assumed to differ between (but not within) the groups. These groupings may arise, for example, from trials that take place on different days or in different locations. They are often known as strata, but in the sense used in surveys (see e.g. SVSTRATIFIED) rather than as in ANOVA.

The model to be fitted is specified by the TERMS parameter. The FACTORIAL option sets a limit on the number of factors and/or variates in the model terms that it defines. An offset can be specified, if required, using the OFFSET option.

The PRINT option controls printed output with settings:

estimates	estimates of parameters;
vcovariance	variance-covariance matrix of the estimates;
residuals	Cox-Snell residuals (see e.g. Collett 2003, Section 4.1.1);
survivor	estimated survival function;
_2loglikelihood	−2 × log-likelihood for the fitted model, the d.f. in the fitted model, and the change from the previous model (if relevant) fitted by RPROPORTIONAL.

The MAXCYCLE option specifies the maximum number of iterations to use when fitting the model (default 50), and the TOLERANCE option defines the convergence criterion (default 0.000001). The EXIT parameter can save a scalar containing the following values to indicate the success or failure of the estimation:

0 success,
1 convergence has not been achieved within MAXCYCLE iterations,
2 convergence is assumed to been achieved, although the value of the deviance has not decreased from the previous iteration.

At other times an error message may occur indicating a *Failure from NAG algorithm*. If the failure code is equal to 3 or 4, alternative starting values should be set using the INITIAL option. If this still fails to converge, it may be that there are insufficient data for the suggested model, and a simpler model may be required.

The RESIDUALS, ESTIMATES, SE, VCOVARIANCE, _2LOGLIKELIHOOD, DFTERMS and SURVIVOR options can be used to save output from the analysis.

Options: PRINT, FACTORIAL, TIMES, CENSORED, OFFSET, BLOCKS, INITIAL, RESIDUALS, ESTIMATES, SE, VCOVARIANCE, _2LOGLIKELIHOOD, DFTERMS, SURVIVOR, EXIT, MAXCYCLE, TOLERANCE.
Parameters: TERMS.

Method

RPROPORTIONAL uses the NAG directive to run the G12BAF algorithm from the NAG Library. This calculates the parameter estimates by maximizing an approximation of the marginal likelihood using a Newton-Raphson iterative technique.

Action with RESTRICT

None of the vectors must be restricted.

References

Cox, D.R. (1972). Regression models and life tables (with discussion). *Journal of the Royal Statistical Society, Series B*, **34**, 187-220.

Collett, D. (2003). *Modelling Survival Data in Medical Research*. Chapman and Hall, London.

See also
Procedures: KAPLANMEIER, RLIFETABLE, RPHFIT, RSTEST, RSURVIVAL.
Genstat Reference Manual 1 Summary section on: Survival analysis.

RQLINEAR

Fits and plots quantile regressions for linear models (D.B. Baird).

Options

PRINT = *string tokens*	What to print (model, estimates, summary, fittedvalues, correlations, wald, jointqtest, separateqtest); default mode, esti, summ, wald
PLOT = *string tokens*	What to plot (rhistogram, phistograms, fittedvalues, estimates, bootestimates); default rhis, phis, fitt
TERMS = *formula*	Terms to be fitted
WEIGHTS = *variate*	Weights for data values; default equally weighted
CONSTANT = *string token*	Whether to include a constant in the model (omit, estimate); default esti
FACTORIAL = *scalar*	Limit on number of factors or variates in a term; default 3.
FITINDIVIDUALLY = *string token*	Whether to fit the regression model one term at a time (yes, no); default no
FULL = *string token*	Whether to assign all possible parameters to factors and interactions (yes, no); default no
BMETHOD = *string token*	Bootstrap method (xy, weightedxy); default xy
NBOOT = *scalar*	Number of times to bootstrap data to estimate confidence limits; default 200
SEED = *scalar*	Seed for bootstrap randomization; default 0
CIPROBABILITY = *scalar*	Probability level for confidence interval; default 0.95
XPLOT = *variate*	Variate to plot fitted values against; default 1st variate in model

Parameters

Y = *variates*	Response variate
PRQUANTILES = *scalars* or *variates*	Proportions at which to calculate quantiles; default 0.5
RESIDUALS = *variates* or *pointers*	Residuals from regression for each quantile
FITTEDVALUES = *variates* or *pointers*	Fitted values from regression for each quantile
ESTIMATES = *variates* or *pointers*	Estimated coefficients of model terms for each quantile
SE = *variates* or *pointers*	Standard errors of the estimated coefficients for each quantile
VCOVARIANCE = *symmetric matrices* or *pointers*	Variance-covariance matrix of estimates for each quantile
DF = *scalars* or *variates*	Numbers of degrees of freedom fitted by the model
LOWER = *variates* or *pointers*	Lower confidence limit of coefficients for each quantile
UPPER = *variates* or *pointers*	Upper confidence limit of coefficients for each quantile
LOWFITTEDVALUES = *variates* or *pointers*	Lower confidence limit of fitted values for each quantile
UPPFITTEDVALUES = *variates* or *pointers*	Upper confidence limit of fitted values for each quantile
OBJECTIVE = *scalars* or *variates*	Optimal values of the objective function
EXIT = *scalars* or *variates*	Exit codes indicating whether the estimation was successful

Description

RQLINEAR calculates and plots quantile regressions. The dependent variate is specified by the Y parameter. The proportions (between 0 and 1) for which the model is to be fitted are specified by the PRQUANTILES parameter, as a scalar is there is only one, or a variate if there are several. The default value for PRQUANTILES is 0.5, i.e. the median.

The model defining the explanatory terms is specified by the TERMS option, and can include variates, factors and polynomial terms, and interactions between them. RQLINEAR cannot fit LOESS or SPLINE models. The FACTORIAL, CONSTANT and FULL options control how the model is constructed, as in the ordinary regression commands (see e.g. FIT or TERMS). FACTORIAL option sets a limit on the number of factors and/or variates in each terms, CONSTANT option allows you to omit the constant term, and FULL controls how each term is parameterized.

Output is controlled by the PRINT option with settings:

model	the details of model that is being fitted;
summary	a summary of the fit;
estimates	the model estimates (and confidence limits, standard errors and t-values if bootstrapping is used);
fittedvalues	the residuals and fitted values from the model;
correlation	correlations between the estimates;
wald	Wald Statistic for each model term;
jointqtest	the significance of the joint changes in the model parameters (excepting the intercept) between the quantiles; and
separateqtest	the significance of the changes in the individual model parameters between the quantiles.

Correlations and Wald statistics are available only if bootstrapping is done. If option FITINDIVIDUALLY=yes, the model terms are added in one at a time, and the Wald statistics are given for for each step. Otherwise only an overall test of the full model versus the null model (i.e. just the constant) is provided. The settings jointqtest and separateqtest are relevant only if several quantiles have been requested by PRQUANTILES. These compare the differences between all the quantiles in a single test. So if you want to compare quantiles for two specific proportions, you should set PRQUANTILES to just those two values.

The PLOT option controls what plots are displayed, with settings

rhistogram	histograms of residuals;
phistograms	histograms of the bootstrap estimates for each parameter;
fittedvalues	observed and fitted values plotted against the explanatory variate specified by the XPLOT option (if XPLOT is not set, the first explanatory variate is used);
estimates	parameter estimates plotted against the quantiles;
bootestimates	parameter estimates and bootstrap confidence limits plotted against quantiles (note this plot can be slow to produce).

For the fitted plot, the observed and fitted values can be plotted against a specific variate given by the option XPLOT, rather than just the default which is the first variate in the TERMS statement.

The BMETHOD option controls the method that is used to obtain standard errors and confidence limits by bootstrapping for the parameter estimates and fitted values. The xy setting re-samples the units with replacement; this is the default. Alternatively, the weightedxy setting uses all the units but with weights are generated from a exponential distribution with mean 1. Bootstrapping can be slow, you can set BMETHOD=* to stop any being done. The NBOOT option specifies the number of bootstrap samples that are taken, and the CIPROBABILITY option sets

the size of the confidence limits. The SEED option defines the seed for the random numbers that are used to select the bootstrap samples. The default of zero continues the existing sequence of random numbers if any have already been used in the current Genstat job. If none have been used, Genstat picks a seed at random.

The results from the model fit can be saved in various parameters. The ESTIMATES, FITTEDVALUES, RESIDUALS, LOWER, UPPER, SE, LOWFITTEDVALUES and UPPFITTEDVALUES parameters save their results in variates if only one quantile has been defined, or in pointers to a set of variates (one for each quantile) if there were several. Similarly VCOVARIANCE saves a symmetric matrix, or a pointer to several symmetric matrices, while DF, OBJECTIVE and EXIT save either a scalar or a variate (with a value for each quantile). EXIT saves the value of the exit code from the estimation of each set of regression quantiles by the FRQUANTILES directive (which is used inside RQLINEAR): a value of zero indicates that the estimation was successful, a value of one means the solution is non-unique (this may not be a problem, as the returned solution will still be optimal), and a value of two means the algorithm has failed.

Options: PRINT, PLOT, TERMS, WEIGHTS, CONSTANT, FACTORIAL, FITINDIVIDUALLY, FULL, BMETHOD, NBOOT, SEED, CIPROBABILITY, XPLOT.
Parameters: Y, PRQUANTILES, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, VCOVARIANCE, DF, LOWER, UPPER, LOWFITTEDVALUES, UPPFITTEDVALUES, OBJECTIVE, EXIT.

Method

The TERMS directive is used to form a design matrix for the model, and the FRQUANTILES directive is then used to to estimate the regression quantiles. For further details of the underlying methodology, see Koenker & D'Orey (1987) or Koenker (2005).

Action with RESTRICT

Restrictions on the Y variate or on variates or factors in the TERMS model are combined, and only those units which are unrestricted in all structures are used in the regression. However, restrictions on the WEIGHTS variate are ignored.

References

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press, New York.
Koenker, R.W. & D'Orey, V. (1987). Algorithm AS229 computing regression quantiles. *Applied Statistics*, **36**, 383-393.

See also

Directive: FRQUANTILES.
Procedures: RQNONLINEAR, RQSMOOTH.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RQNONLINEAR

Fits and plots quantile regressions for nonlinear models (D.B. Baird).

Options

PRINT = *string tokens*	What to print (model, estimates, summary, fittedvalues, correlations, monitoring); default mode, esti, summ
PLOT = *string tokens*	What to plot (rhistogram, phistograms, fittedvalues, confidencelimits); default phis, fitt, conf
X = *variates*	Variates to fit in the model
DATA = *variates or factors*	Data to bootstrap in parallel with Y; default takes the variates and factors of the same length as Y involved in the CALCULATION expressions
CONSTANT = *string token*	Whether to include a constant in the model (omit, estimate); default esti
CALCULATION = *expression structures*	Calculation of explanatory variates involving nonlinear parameters
PARAMETERS = *pointer*	Pointer to scalars representing the nonlinear parameters to be optimized in the expressions
INITIAL = *variate*	Initial values for parameters
LOWPARAMETERS = *variate*	Lower bound for parameters
UPPPARAMETERS = *variate*	Upper bound for parameters
STEPLENGTHS = *variate*	Step sizes for parameters
LINEARPARAMETERS = *pointer*	Pointer to scalars representing the linear parameters in the model (including the constant)
METHOD = *string token*	Which optimization method to use (gaussnewton, newtonraphson, fletcherpowell, simplex); default gaus
NBOOT = *scalar*	Number of times to bootstrap data to estimate confidence limits; default 100
SEED = *scalar*	Seed for bootstrap randomization; default 0
CIPROBABILITY = *scalar*	Probability level for confidence interval; default 0.95
MAXCYCLE = *scalar*	Maximum number of iterations for optimization; default 200
XPLOT = *variate*	Variate to plot fitted values against; default is the first variate on the right-hand side of the CALCULATION expressions

Parameters

Y = *variates*	Response variates
PRQUANTILE = *scalars*	Proportion at which to calculate the quantile for each response variate; default 0.5
RESIDUALS = *variates*	Residuals from the nonlinear model
FITTEDVALUES = *variates*	Fitted values from the nonlinear model
ESTIMATES = *variates*	Estimates of the parameters in the model (nonlinear, linear and constant)
SE = *variates*	Standard errors of the parameters
VCOVARIANCE = *symmetric matrices*	Variance-covariance matrix for the parameters

LOWER = *variates*	Lower confidence limits for the parameters
UPPER = *variates*	Upper confidence limits for the parameters
LOWFITTEDVALUES = *variates*	Lower confidence limits for the fitted values
UPPFITTEDVALUES = *variates*	Upper confidence limits for the fitted values
OBJECTIVE = *scalars*	Optimal values of the objective function
TITLE = *texts*	Titles for fitted value graphs

Description

RQNONLINEAR calculates and plots quantile nonlinear regressions. The dependent variate is specified by the Y parameter. The proportion (between 0 and 1) for which the model is to be fitted is specified by the PRQUANTILE parameter, as a scalar is there is only one. The default value is 0.5, i.e. the median.

The X option lists the variates that are to be fitted in the model. Some of these will be functions of nonlinear parameters, which must be be supplied (as a set of scalars in a pointer) using the PARAMETERS option. The CALCULATION option supplies a list of expression structures to calculate the values of the relevant X variates (from the parameters and other data structures). By default the model will include the constant, but this can be omitted by setting option CONSTANT=omit. The LINEARPARAMETERS option can be set to a pointer containing a set of scalars to represent the linear parameters in the model (i.e. the regression coefficients and the constant, if present). Initial values, lower and upper bounds and step lengths for the parameters are supplied, in variates, by the INITIAL, LOWPARAMETER, UPPPARAMETER and STEPLENGTHS options, respectively. The METHOD option specifies the method to use to estimate the nonlinear parameters. The settings gaussnewton, newtonraphson and fletcherpowell use the FITNONLINEAR directive, with the Gauss-Newton, Newton-Raphson or Fletcher-Powell optimization methods, respectively. These methods require initial values to be supplied. The simplex setting uses the SIMPLEX procedure, which requires lower and upper bounds to be supplied. The MAXCYCLE option specifies the maximum number of iterations to be used.

Output is controlled by the PRINT option with settings:

model	a description of the model;
summary	a summary of the fit;
estimates	the model estimates (and confidence limits, standard errors and t-values if bootstrapping is used);
fittedvalues	the residuals and fitted values from the model;
correlation	correlations between the estimates; and
monitoring	monitoring information for the fit.

Correlations are available only if bootstrapping is done.

The PLOT option controls what plots are displayed, with settings

rhistogram	histograms of residuals;
phistograms	histograms of the bootstrap estimates for each parameter;
fittedvalues	observed and fitted values plotted against the explanatory variate specified by the XPLOT option (if XPLOT is not set, the first explanatory variate is used);
confidenceintervals	includes confidence intervals in the fitted-value plot (available only if bootstrapping is done).

For the fitted plot, the observed and fitted values can be plotted against a specific variate given by the option XPLOT, rather than just the default which is the first variate in the right-hand side of the CALCULATION expressions. The TITLE parameter can supply a title for the plot.

The NBOOT option specifies the number of bootstrap samples that are taken, and the CIPROBABILITY option sets the size of the confidence limits. The SEED option defines the seed for the random numbers that are used to select the bootstrap samples. The default of zero continues the existing sequence of random numbers if any have already been used in the current

Genstat job. If none have been used, Genstat picks a seed at random. RQNONLINEAR can automatically select the data vectors to bootstrap along with the Y variate: they consist of all the variates or factors on the right-hand side of the CALCULATION expressions that are of the same length as Y, plus any X variates that are not calculated by the expressiions. If this does not produce the correct set of vectors for bootstrapping, you can specify them automatically using the DATA option.

The results from the nonlinear fit can be saved by the parameters RESIDUALS, FITTEDVALUES, ESTIMATES, SE, VCOVARIANCE, DF, LOWER, UPPER, LOWFITTEDVALUES, UPPFITTEDVALUES and OBJECTIVE.

Options: PRINT, PLOT, X, DATA, CONSTANT, CALCULATION, PARAMETERS, INITIAL, LOWPARAMETER, UPPPARAMETER, STEPLENGTHS, LINEARPARAMETERS, METHOD, NBOOT, SEED, CIPROBABILITY, MAXCYCLE, XPLOT.

Parameters: Y, PRQUANTILES, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, VCOVARIANCE, LOWER, UPPER, LOWFITTEDVALUES, UPPFITTEDVALUES, OBJECTIVE, TITLE.

Method

The nonlinear parameters are estimated by either FITNONLINEAR or SIMPLEX, operating on a target function in which the objective function from the quantile regression is calculated by the RQOBJECTIVE function. The FRQUANTILES directive is then used to obtain the estimates of the linear parameters. For further details of the underlying methodology, see Koenker & D'Orey (1987) or Koenker (2005).

Action with RESTRICT

Restrictions on the Y variate or on X variates or factors are combined, and only those units which are unrestricted in all structures are used in the regression.

References

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press, New York.

Koenker, R.W. & D'Orey, V. (1987). Algorithm AS229 computing regression quantiles. *Applied Statistics*, **36**, 383-393.

See also

Directive: FRQUANTILES.
Procedures: RQLINEAR, RQSMOOTH.
Function: RQOBJECTIVE.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RQSMOOTH

Fits and plots quantile regressions for loess or spline models (D.B. Baird).

Options

PRINT = *string tokens*	What to print (model, summary, fittedvalues); default mode, summ
PLOT = *string tokens*	What to plot (rhistogram, fittedvalues); default fitt
METHOD = *string token*	Smoothing method (loess, spline); default spli
DF = *scalar*	Spline Degrees of Freedom (3-40); default 4
KNOTS = *variate*	Knot points for smoothing splines; default * uses equally spaced percentiles of the X variate
KERNEL = *string token*	What Kernel to use for Loess (normal, epanechnikov, quadratic, triweight, tukeybiweight, quartic, linear, uniform); default norm
LMETHOD = *string token*	Span method for Loess (constant, adaptive); default adap
BANDWIDTH = *scalar*	Bandwidth for smoothing between 0 and 1; default 0.4
ORDER = *scalar*	Order of local polynomial; default 1
NGRIDPOINTS = *scalar*	Number of points on smooth curve; default 100
NBOOT = *scalar*	Number of times to bootstrap data to estimate confidence limits; default 0 i.e. no bootstrapping
SEED = *scalar*	Seed for bootstrap randomization; default 0
CIPROBABILITY = *scalar*	Probability level for confidence interval; default 0.95
TITLE = *text*	Title for plots; default * generates titles from the structure names
ARRANGEMENT = *string token*	Whether to plot fitted regressions by the GROUPS parameter in a trellis plot (single, trellis); default sing

Parameters

Y = *variates*	Response variate
X = *variates*	Explanatory variate
PRQUANTILES = *scalars* or *variates*	Proportions at which to calculate quantiles; default 0.5
GROUPS = *factors*	Groups for which independent curves are fitted
GRID = *variates*	Grid of equidistant points at which the smooth is calculated
OUTGROUPS = *factors*	Groups for the fitted smoothed values saved by the SMOOTH parameter
SMOOTH = *variates* or *pointers*	Fitted smooth estimated at the NGRIDPOINTS points given in GRID
SLOPE = *variates* or *pointers*	Fitted slope from model for the same points as SMOOTH
RESIDUALS = *variates* or *pointers*	Residuals from regression for each quantile
FITTEDVALUES = *variates* or *pointers*	Fitted values from regression for each quantile
LOWSMOOTH = *variates* or *pointers*	Lower confidence limit of smooth for each quantile
UPPSMOOTH = *variates* or *pointers*	Upper confidence limit of smooth for each quantile
SESMOOTH = *variates* or *pointers*	Standard error of coefficients for each quantile

Description

RQSMOOTH calculates and plots a smooth quantile regression for a given dependent variate y and an explanatory variable x, specified by the Y and X parameters, respectively. You can also specify groups, by supplying a factor using the GROUPS parameter; the model is then fitted independently within each group. The type of the smooth model, either loess or spline, is specified by the METHOD option. The quantiles (between 0 and 1) for which the model is to be fitted are specified by the PRQUANTILES parameter, as a scalar is there is only one, or a variate if there are several. The default value for PRQUANTILES is 0.5, i.e. the median.

For a spline model, the number of degrees of freedom can be specified using the DF option. This must be greater or equal to 3 and less then or equal to 40. The knot points for the spline basis curves can be set using the KNOTS option. This must have DF points and no missing values. If KNOTS is not provided, the default knot points are DF equally spaced percentiles of the X variate.

For a loess model the bandwidth is set by the BANDWIDTH option, and must lie between 0 and 1; the default is 0.4. With large bandwidths the function will be smoother but less responsive, allowing for higher bias where the curve is rapidly changing. With smaller bandwidths the curve will be more responsive the curve, but the confidence limits around the curve will be larger. So the choice of bandwidth controls the trade-off between variance and bias. The loess model uses a moving window centred around the point to be predicted. The width of this window is controlled by the bandwidth and the LMETHOD option. Setting LMETHOD=constant gives a constant window width of BANDWIDTH * RANGE(X). Alternatively, setting LMETHOD=adaptive uses a varying window width, defined so that it always contains the proportion of the total points, defined by bandwidth. The window will thus be narrower where the points are denser. A local polynomial is fitted to the points in the window. The order is defined by the ORDER option as either 1 (linear) or 2 (quadratic). The points are in the polynomial regression weighted by their distance from the point that is to be predicted. The weighting function W(d) is selected using the KERNEL option, with settings:

uniform	$W(d) = 1$
linear	$W(d) = 1 - \text{ABS}(d)$
quadratic	$W(d) = 1 - d^2$
quartic	$W(d) = (1 - d^2)^2$
triweight	$W(d) = (1 - d^2)^3$
Normal	$W(d) = \text{PRNORMAL}(d)$
epanechnikov	synonym of quadratic
tukeybiweight	synonym of quartic

where d is the distance within the window from the predicted point, scaled to take the values −1 and +1 at the lower and upper window edges.

Output is controlled by the PRINT option with settings:

model	the details of model that is being fitted;
summary	a summary of the fit; and
fittedvalues	the residuals and fitted values from the model.

The PLOT option controls what plots are displayed, with settings

rhistogram	histograms of residuals; and
fittedvalues	observed and fitted values plotted against the explanatory variate specified by the XPLOT option (if XPLOT is not set, the first expolanatory variate is used).

The ARRANGEMENT option controls whether the models for each group are displayed in a trellis plot or in a single plot with all groups together.

Bootstrapping can be used to estimate standard errors and confidence limits for the fitted values. The NBOOT option specifies the number of bootstrap samples that are taken; the default is zero, which indicates that no bootstrapping is to be done. The CIPROBABILITY option sets

the size of the confidence limits. The SEED option defines the seed for the random numbers that are used to select the bootstrap samples. The default of zero continues the existing sequence of random numbers if any have already been used in the current Genstat job. If none have been used, Genstat picks a seed at random.

The results from the model fit can be saved in various parameters. They will be saved in a variate if only one quantile has been defined, or in a pointer to a set of variates (one for each quantile) if there were several. The fitted curve(s) can be saved by the SMOOTH parameter, and the slope of the fitted curve by the SLOPE parameter. The NGRIDPOINTS option controls how many points are estimated on each curve. The GRID parameter can save the positions of the points, which will be spaced equally between the minimum and maximum value of X. The UPPSMOOTH, LOWSMOOTH and SESMOOTH parameters save variates containing the bootstrap confidence limits and standard errors of the estimated curve respectively. If a GROUPS factor has been specified, the estimated values for the curves have NLEVELS(GROUPS) * NGRIDPOINTS points, with the values for group 1 being given first, followed by those for group 2, and so on. The OUTGROUPS factor can save a factor to identify the groups within the variates.

Options: PRINT, PLOT, METHOD, KERNEL, LMETHOD, BANDWIDTH, ORDER, DF, KNOTS, NGRIDPOINTS, NBOOT, SEED, CIPROBABILITY, TITLE, ARRANGEMENT.

Parameters: Y, X, PRQUANTILES, GROUPS, GRID, OUTGROUPS, SMOOTH, SLOPE, RESIDUALS, FITTEDVALUES, LOWSMOOTH, UPPSMOOTH, SESMOOTH.

Method

The FRQUANTILES directive is used to fit the quantile regression for a design matrix generated for the spline basis or a locally weighted regression about the points in the smooth. For further details of the underlying methodology, see Koenker & D'Orey (1987) or Koenker (2005).

Action with RESTRICT

Restrictions in the Y and X variate and GROUPS factor are combined, and only those units which are unrestricted in all structures are used in the regression.

References

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press, New York.

Koenker, R.W. & D'Orey, V. (1987). Algorithm AS229 computing regression quantiles. *Applied Statistics*, **36**, 383-393.

See also

Directive: FRQUANTILES.

Procedures: RQLINEAR, RQNONLINEAR.

Genstat Reference Manual 1 Summary section on: Regression analysis.

RQUADRATIC

Fits a quadratic surface and estimates its stationary point (R.W. Payne).

Options

PRINT = *string tokens*	What to print (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, monitoring, confidence, stationary); default mode, summ, esti
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalars*	Limit for expansion of model terms; default 3
POOL = *string token*	Whether to pool ss in accumulated summary between all terms fitted in a linear model (yes, no); default no
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, leverage, residual, aliasing, marginality, vertical, df, inflation); default *
FPROBABILITY = *string token*	Printing of probabilities for variance and deviance ratios (yes, no); default no
TPROBABILITY = *string token*	Printing of probabilities for t-statistics (yes, no); default no
SELECTION = *string tokens*	Statistics to be displayed in the summary of analysis produced by PRINT=summary, seobservations is relevant only for a Normally distributed response, and %cv only for a gamma-distributed response (%variance, %ss, adjustedr2, r2, seobservations, dispersion, %cv, %meandeviance, %deviance, aic, bic, sic); default %var, seob if DIST=normal, %cv if DIST=gamma, and disp for other distributions
PROBABILITY = *scalar*	Probability level for confidence intervals for parameter estimates; default 0.95
STATIONARY = *scalars*	Saves the estimated value of y at the stationary point
SESTATIONARY = *scalars*	Saves the standard error of the estimated value of y at the stationary point
TYPESTATIONARY = *scalars*	Identifies the type of stationary point (2 for maximum, 1 for maximum on a ridge, −2 for minimum, −1 for minimum on a ridge, or 0 for saddle point)
PREDICTIONS = *matrix*	Saves predictions
PLOT = *string tokens*	What to plot (contour, surface); default * i.e. nothing
COLOURS = *text* or *variate*	Colours for the plots

Parameters

X = *variates*	X-variates whose linear, quadratic and product terms define the quadratic surface
ESTIMATE = *scalars*	Estimated value of each x-variate at the stationary point
SE = *scalars*	Standard error of the estimated value of each x-variate at the stationary point

LEVELS = *variates*	Values at which to evaluate each X for plots and predictions

Description

RQUADRATIC fits a quadratic surface of several variates, and estimates the stationary point. It is used similarly to FIT. It must be preceded by a MODEL statement, and can be followed by RCHECK, RDISPLAY, RGRAPH, RKEEP, ADD, DROP, SWITCH and so on. It also has options PRINT, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION and PROBABILITY which operate similarly to those of FIT, except that PRINT has an additional setting stationary to print the stationary point.

The x-variates whose linear, quadratic and product terms define the quadratic surface are specified by the X parameter. There are also parameters ESTIMATE and SE to save the estimated value of each x-variate, and its standard error, at the stationary point. The y-value at the stationary point, and its standard error, can be saved by the STATIONARY and SESTATIONARY options. The TYPESTATIONARY option saves a scalar, with one of the following values to identify the type of stationary point: 2 maximum, 1 maximum on a ridge, −2 minimum, −1 minimum on a ridge, or 0 saddlepoint.

The PREDICTIONS option can save predictions from the fitted quadratic model. The LEVELS parameter specifies a variate for each X, to specify the values at which to form predictions. The predictions are stored in a matrix. The final column contains the predictions, and the earlier columns (one for each X variate) store the set of x-values at which each prediction was made.

The PLOT option specifies which plots to display, with settings:

contour	for a contour plot, and
surface	for surface plot.

By default nothing is plotted. The COLOURS option specifies a text or variate to define the colours to use. (This is used as the setting of the PENFILL parameter of DCONTOUR and DSURFACE.) The default is a text containing the values 'darkgreen' and 'yellow'.

Options: PRINT, CONSTANT, FACTORIAL, POOL, DENOMINATOR, NOMESSAGE, FPROBABILITY, TPROBABILITY, SELECTION, PROBABILITY, STATIONARY, SESTATIONARY, TYPESTATIONARY, PREDICTIONS, PLOT, COLOURS.

Parameters: X, ESTIMATE, SE, LEVELS.

Method

RQUADRATIC forms variates with the quadratic and product terms of the x-variates, and fits these together with the x-variates themselves. The RFUNCTION directive is then used to estimate the x- and y-values at the stationary point, with their standard errors. The type of stationary point is identified by an eigenvalue decomposition of the symmetric matrix of estimated regression coefficients of the product and quadratic terms, as described in Section 9.4 of Wu & Hamada (2000).

Action with RESTRICT

As in FIT, the y-variate (specified in an earlier MODEL directive) can be restricted to analyse a subset of the data.

Reference

Wu, C.F.J & Hamada, M. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization*. Wiley, New York.

See also
Directive: AFRESPONSESURFACE.
Procedures: AGBOXBEHNKEN, AGCENTRALCOMPOSITE, VSURFACE.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RRETRIEVE

Retrieves a regression save structure from an external file (R.W. Payne).

No options

Parameters

FILENAME = *texts*	Name of the file storing the save structure
EXIT = *scalars*	Scalar that contains the value one if the save structure could not be retrieved successfully, otherwise zero
SAVE = *regression save structures*	Save structure that has been retrieved

Description

RRETRIEVE retrieves a regression save structure, stored earlier by the RSTORE procedure in an external file. It can then be used to produce further output from the analysis. (See, for example, directives RDISPLAY and RKEEP, or procedures RCHECK, RGRAPH and RSPREADSHEET.)

The name (and path) of the file that stores the save structure is specified, in a text, by the FILENAME parameter. The save structure is saved by the SAVE parameter. The EXIT parameter can return a scalar containing the value one if the save structure could not be retrieved successfully. Otherwise it contains zero.

Options: none.
Parameters: FILENAME, EXIT, SAVE.

Method

RSTORE stores the save structure in a Genstat backing-store file using the STORE directive, and RRETRIEVE retrieves it using the RETRIEVE directive.

See also

Directive: FIT.
Procedures: ARETRIEVE, RSTORE.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RSCHNUTE

Fits a general 4 parameter growth model to a non-decreasing Y-variate; synonym `FITSCHNUTE` (A. Keen).

Options

`PRINT` = *string tokens*	What to print (`model`, `summary`, `estimates`, `correlations`, `fittedvalues`, `accumulated`, `monitoring`); default `mode`, `summ`, `esti`
`T1` = *scalar*	Timepoint defining y_1; default the first timepoint with $\mu > 0.4 \times y_2$ (μ and y_2 are obtained by an approximating model)
`T2` = *scalar*	Timepoint defining y_2; default `*` takes the last observed timepoint
`NGRID` = *scalar*	The number of points for a grid search with parameters *a* and/or *b*; default 7
`PLUS` = *scalar*	The constant added to the observed and fitted values, in order to obtain a suitable variance function in case of other than normal error distribution; default `*` takes the smallest possible value for the response given the rounding off
`A` = *scalar*	Fixed value for parameter *a* of the growth model, defining a submodel; only 0 is appropriate; default `*`
`B` = *scalar*	Fixed value for parameter *b* of the growth model; default *
`ALOWER` = *scalar*	Lower bound for parameter *a* of the growth model; default $-40/(t_2-t_1)$
`AUPPER` = *scalar*	Upper bound for parameter a of the growth model; default $40/(t_2-t_1)$
`BLOWER` = *scalar*	Lower bound for parameter *b* of the growth model; default −20
`BUPPER` = *scalar*	Upper bound for parameter *b* of the growth model; default 20
`MAXCYCLE` = *scalar*	Maximum number of iterations; default 20
`TOLERANCE` = *scalar*	Convergence criterion; default 0.0004

Parameters

`T` = *variates*	Observed timepoints for each fit
`MGRID` = *matrices*	Deviances from the gridsearch in *a* and/or *b*
`RT` = *pointers*	Pointer of two variates: the fitted growth rates and relative growth rates at the observed timepoints
`OWNT` = *variates*	A variate of arbitrary timepoints to be specified by the user e.g. for obtaining a smooth plot of fitted values
`ROWNT` = *pointers*	Pointer of three variates: the fitted values, growth rates and relative growth rates at the timepoints specified in `OWNT`
`EXTRA` = *pointers*	Pointer of eight scalars, with: 1) the starting point of the curve below which the response equals 0, 2) the endpoint of the curve where the reponse is infinite, 3) the lower asymptote of the curve, 4) the upper asymptote of the curve, 5) the inflexion point, 6) the fitted value at the point of inflexion, 7) the growth rate at the point of

inflexion, 8) the relative growth rate at the point of inflexion; if no finite value for a scalar exists, the value is set to be missing

Description

Schnute (1981) has described a general four parameter growth model for a continuous response y with an expectation μ that is a non-decreasing function of time t. The Method section contains a short overview of the essential features of the model, including the meaning of the parameters and the generality of the model. The parameterization is statistically stable. The procedure RSCHNUTE has been developed to provide a robust tool for fitting this model. Numerical problems are reduced to a minimum, so that attention can be focused on proper model specification. The default settings are such that usually convergence is obtained without intervention. However, options and parameters have been included to control the iteration process if necessary.

The MODEL directive must be set, outside the procedure, each time it is used. The MODEL settings are changed within the procedure, so it must also be reset if any of the ordinary directives for fitting regressions, curves or generalized linear models is to be used after calling RSCHNUTE.

Errors can be specified as Poisson, gamma and inverse normal as well as normal. Their variance function can be expressed as:

$$\sigma^2 = k \times (\mu^q) ,$$

with q = 0, 1, 2 and 3 for the normal, Poisson, gamma and inverse normal distributions respectively. Increasing q implies increasing the importance of deviance contributions from observations with small values of μ. If μ is very small, the variance function is usually unrealistic for non-normal distributions, due to rounding and approximation errors. For the gamma and inverse normal distributions it is theoretically impossible to obtain observations equal to 0, but in practice, due to rounding and approximation errors, such observations do occur. In calculating the deviance, small values of y or μ cause numerical problems. Therefore, by default, a small constant c is added to y as well as to μ, leaving the model unaffected but changing the variance function to: $\sigma^2 = k \times (\mu + c)^q$. Parameter PLUS provides a means for specifying c.

Submodels are models with fixed values of parameters a and/or b of the growth model. They can be specified by setting the options A and B. The generality of the model can also be restricted by specifying a particular range of a and b, using the options ALOWER, BLOWER, AUPPER and/or BUPPER of the procedure.

Two of the model parameters, y_1 and y_2, can be modified by the user, by specifying the options T1 and T2. This should be done very cautiously and only if proven necessary, because choice of T1 and T2 affects the iteration process considerably; however, suitable default values are provided.

Initial estimates of all four parameters of the growth model (or less, if A and/or B have been specified) are derived within the procedure, by fitting an approximate generalized linear model followed by a grid search. Usually it is not necessary to take any action in order to obtain convergence. If, however, some prior knowledge about the shape of the curve is available, the iteration process can be improved by restricting the range of a and b as far as possible. The number of gridpoints for the grid search can be changed by option NGRID, with a maximum of 12. The deviances in the gridpoints can be saved setting parameter MGRID. This allows inspection of the likelihood-surface (conditional on initial values for y_1 and y_2). The maximum number of iteration cycles and the convergence criterion can be set by the options MAXCYCLE and TOLERANCE.

The growth rate and relative growth rate at the observed timepoints can be saved by setting parameter RT. Fitted values, growth rates and relative growth rates for arbitrary timepoints, given in parameter OWNT, can be saved by setting parameter ROWNT. This can be useful not only for a

plot of the fitted curve and its derivative, but also for obtaining estimates of y_1 and y_2 with the same meaning in different situations. Special features of the fitted curve, (start- and end-point, asymptotes and various estimates at the point of inflexion) can be saved by setting the parameter EXTRA.

Other results of the nonlinear regression can be obtained in the usual way outside the procedure, using the RKEEP directive.

Options: PRINT, T1, T2, NGRID, PLUS, A, B, ALOWER, AUPPER, BLOWER, BUPPER, MAXCYCLE, TOLERANCE.
Parameters: T, MGRID, RT, OWNT, ROWNT, EXTRA.

Method

The expectation μ of y satisfies the following equation (with parameters y_1, y_2, a and b):

$$\mu^b = (y_1^b) + [\,(y_2^b) - (y_1^b)\,] \times f(t; a)$$

with $f(t; a) = \{\, 1 - \exp[\,-a \times (t-t_1)\,]\,\} / \{\, 1 - \exp[-a \times (t_2-t_1)\,]\,\}$

For limiting forms of the equation, substitute:

$f(t; a) = (t-t_1) / (t_2-t_1)$ if $a=0$

$x^b = \log(x)$ if $b=0$; $x = \mu, y_1, y_2$

y_1 and y_2 are the values of μ at an early timepoint t_1 and a late timepoint t_2 respectively. Timepoints t_1 and t_2 must be fixed. y_1 and y_2 determine location and scale. The other two parameters, a and b, determine the shape of the curve, this can range from curves with horizontal asymptotes to curves with vertical asymptotes and includes the straight line, S-shaped curves (Richards and Von Bertalanffy curves), exponential and power curves. An overview of the submodels, showing the generality of the model, is presented in the table below.

$a>0$ $b>0$	Generalized Von Bertalanffy:	$y_\infty \times \{\, 1 - \exp[-a \times (t-t_0)]\,\}^{1/b}$
$a>0$ $b=0$	Gompertz:	$y_\infty \times \exp\{\, -\exp[-a \times (t-t_*)]\,\}$
$a>0$ $b<0$	Richards:	$y_\infty \times \{\, 1 + h \times \exp[-a \times (t-t_0)]\,\}^{1/b}$; $b=-h$
$a=0$ $b>0$	power:	$[\,\alpha + (\beta \times t)\,]^{1/b}$; $\beta > 0$
$a=0$ $b=0$	pure exponential:	$\alpha \times (\beta^t)$; $\beta > 0$
$a=0$ $b<0$	power of inverse linear:	$1 / [\alpha + (\beta \times t)\,]^{1/h}$; $\beta > 0$, $b=-h$
$a<0$ $b>0$	power of exponential:	$\{\alpha + [\beta \times \exp(-a \times t)]^{1/b}$; $\beta > 0$
$a<0$ $b=0$		$\alpha \times \exp[\,\beta \times \exp(\,-a \times t)\,]$; $\beta > 0$
$a<0$ $b<0$	inverse of power of exponential:	$1 / \{\alpha + [\beta \times \exp(-a \times t)]^{1/h}$; $\beta < 0$, $b=-h$

In these equations y_∞ (the asymptote as t tends to infinity), t_0 (the starting point of growth), t_* (the point of inflexion), α and β are functions of y_1 and y_2; these are different functions in the different equations. The Richards curve is the generalized logistic, which includes the logistic (with $b = -1$). S-shaped curves have $a>0$ and $b<1$. If an inflexion point exists, the relative growth rate at that point equals $a/(1-b)$ and the value of μ at the point of inflexion relative to the upper asymptote then equals $(1-b)^{1/b}$.

From the above equations it can be seen that the growth model contains the curves exponential, logistic, glogistic, gompertz and ldl of the directive FITCURVE (the last four with CONSTANT=omit only). However, in contrast to FITCURVE, all facilities of MODEL

can be used and not just the normal distribution for error. But there is no possibility to include a dependence on other factors, to fix the response at the origin or to add a constant to the model.

Action with `RESTRICT`
The response variate and/or the time variate may be restricted. The restrictions must be identical. Only the units not excluded by the restriction will be analysed.

Reference
Schnute, J. (1981). A Versatile Growth Model with Statistically Stable Parameters. *Can. J. Fish. Aquat. Sci*, **38**, 1128-1140.

See also
Directives: `FITCURVE`, `FITNONLINEAR`.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RSCREEN

Performs screening tests for generalized or multivariate linear models (H. van der Voet).

Options

`PRINT` = *string tokens*	Printed output required (`model`, `pool`, `starscheme`, `tests`, `pvalues`); default `mode`, `pool`, `star`
`CONSTANT` = *string token*	How to treat the constant (`estimate`, `omit`); default `esti`
`FACTORIAL` = *scalar*	Limit for expansion of model terms; default 3
`NOMESSAGE` = *string tokens*	Which warning messages to suppress when fitting the complete model (`aliasing`, `marginality`): warning messages are always suppressed when fitting models for individual tests; default *
`EXCLUDEHIGHER` = *string token*	Whether to exclude higher-order interactions in the conditional regression model for each tested term (`yes`, `no`); default `no`
`FORCED` = *formula*	Terms always included in the model (no tests on these terms); default *
`TESTED` = *text*	To save the names of individual terms which are tested
`NELEMENTS` = *variate*	To save the number of identifiers composing each individual term
`MARGINAL` = *pointer*	To save results from marginal tests for each tested term in a pointer containing the test statistic, corresponding degrees of freedom and the calculated probability
`CONDITIONAL` = *pointer*	To save results from conditional tests for each tested term in a pointer containing the test statistic, corresponding degrees of freedom and the calculated probability
`MVINCLUDE` = *string token*	Whether to include units with missing values in non-relevant explanatory variates or factors when calculating conditional and marginal tests (`yes`, `no`); default `no`

Parameter

`FREE` = *formula*	List of explanatory variates and factors, or model formula; each term from the expanded `FREE` formula is tested in a marginal and in a conditional test, unless the term is also part of the `FORCED` formula

Description

`RSCREEN` calculates marginal and conditional tests for all terms in a (multivariate) linear or generalized linear model. For multivariate linear regression models these tests are based on Wilks' Lambda. `RSCREEN` also performs pooled testing of all main effects, of all 2-factor interactions, etc.

A call to `RSCREEN` must be preceded by a `MODEL` statement which defines the response variate(s) and, if required, a vector of weights, an offset and other aspects of a generalized linear model. More than one response variable is allowed for ordinary linear models, in which case multivariate linear regression models are fitted and tests are based on Rao's F approximation of Wilks' Lambda. If there is one response variable, tests are based on (scaled) deviances or deviance ratios, according to the setting of the `DISPERSION` option in the `MODEL` directive. Deviance ratios are always based on the mean deviance of the full model.

The `FREE` parameter specifies the model terms which have to be tested. The limit for

expanding the FREE model formula can be set with the FACTORIAL option with default value 3. Two tests are performed for each term in the expanded model formula:

1. a marginal test: the term is added to the simplest possible model. For example, the main effect of A is added to the null model and the interaction term A.B is added to a model containing only main effects A and B.
2. a conditional test: the term is added to the most complex possible model containing no terms involving the term which is tested. For example, interaction A.B is added to the model with all terms except those involving A.B, like for example the interaction A.B.C. Note that e.g. the interaction C.D.E will be included in the model when testing A.B. The inclusion of any higher-order term can be prevented by setting option EXCLUDEHIGHER=yes.

It is sometimes desirable to include specific terms in every model. Such terms may be specified by means of the FORCED option. The FORCED model formula is fitted first and no test results are given for the FORCED terms. The CONSTANT option controls whether the constant parameter is included in the model.

By default any units with missing values in any of the explanatory variates or factors will be excluded from all of the tests. However, if you have many missing values that spread unevenly over the explanatory variables, there may be few units with non-missing values for every variable. If you have only a single y-variate, you may then want to set option MVINCLUDE=explanatory. RSCREEN will then use all the available units when constructing each marginal or conditional test. So it ignores missing values in any explanatory variable that is not involved in the test. This provides more information for each test, but the tables of tests should be interpreted with care as different tests may be based on different sets of units.

The PRINT option controls output. The model setting gives a description of the model. The pool setting prints an accumulated analysis of variance or deviance in which terms with the same number of identifiers, e.g. main effects or two-factor interactions, are pooled. PRINT=tests prints both marginal and conditional test statistics, while setting pvalues prints (approximate) P-values from chi-square or F-tests. Finally, PRINT=starscheme prints significance of P-values by a conventional star notation. The default setting of PRINT is model, pool, starscheme.

Output can be saved by means of options TESTED, NELEMENTS, MARGINAL and CONDITIONAL. TESTED saves the individual model terms in a text structure, while NELEMENTS saves the number of identifiers composing each individual term. MARGINAL and CONDITIONAL save test results in a pointer which contains four variates. These variates save the test statistic, the corresponding degrees of freedom for numerator and denominator and the calculated (approximate) probability. For chi-square tests the degrees of freedom for the denominator are set to missing. For multivariate linear regression models, Rao's F-statistic and the corresponding degrees of freedom are saved. Note that, when MVINCLUDE=no, units with one or more missing values in any term are excluded from the analysis. This implies that FIT used for a subset of terms may give different results than RSCREEN.

All regression warnings are suppressed, except when fitting the full model. This is to prevent the printing of long lists of similar warnings like "Iterative weights have become 0, or have been held at a limit".

If RSCREEN is used for log-linear models, with the option EXCLUDEHIGHER set to yes, the marginal and conditional tests are equal to the marginal and partial tests of Brown (1976), which are available e.g. in BMDP. RSCREEN can also be used to implement the model selection strategy used in GLIMPSE, as described in McCullagh & Nelder (1989), pages 91-93. However, RSCREEN does not use approximations for models that require an iterative fitting process.

Options: PRINT, CONSTANT, FACTORIAL, NOMESSAGE, EXCLUDEHIGHER, FORCED, TESTED, NELEMENTS, MARGINAL, CONDITIONAL, MVINCLUDE.

Parameter: FREE.

Method

Most of the implementation is straightforward. The null model for the marginal test for term t is constructed as #FORCED + ((#FREE - #FORCED) -* c[]) - #t, where c[] is the classifying set of factors and variates comprising #FREE - #FORCED excluding factors and variates in term t. The null model for the conditional test is #FORCED + #FREE -* #t.

When the DISPERSION option of the MODEL directive is set to *, terms are tested by means of F statistics, which are deviance ratios based on the mean deviance of the full model. For a fixed dispersion parameter chi-square statistics are used, i.e. deviance differences scaled by the dispersion parameter. Terms in multivariate linear models are tested by Rao's F-approximation for Wilks' Lambda (Rao 1973). These are always based on residual variation calculated for the full model.

Smoothing splines are not allowed in the FREE formula due to a limitation of the FCLASSIFICATION directive.

Action with RESTRICT

Any restriction applied to vectors used in the regression model applies also to the results from RSCREEN.

References

Brown, M.B. (1976). Screening effects in multidimensional contingency tables. *Applied Statistics*, **25**, 37-46.

McCullagh, P. & Nelder, J.A. (1989). *Generalized Linear Models (second edition)*. Chapman & Hall, London.

Rao, C.R. (1973). *Linear Statistical Inference and its Applications*. Wiley, New York.

See also

Procedures: ASCREEN, RSEARCH, RWALD, VSCREEN.

Genstat Reference Manual 1 Summary section on: Regression analysis.

RSEARCH

Helps search through models for a regression or generalized linear model (P.W. Goedhart).

Options

PRINT = *string token*	Printed output required (model, results); default mode, resu
METHOD = *string tokens*	Model selection method to employ (allpossible, forward, backward, fstepwise, bstepwise, accumulated, pooled); default allp
FORCED = *formula*	Model formula to include in every model; default *
CONSTANT = *string token*	How to treat the constant (estimate, omit); default esti
FACTORIAL = *scalar*	Limit for expansion of all model terms; default 3
DENOMINATOR = *string token*	Whether to base ratios in accumulated summaries on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
INRATIO = *scalar*	Criterion for inclusion of terms for forward selection, backward elimination and stepwise regression; default 1.0
OUTRATIO = *scalar*	Criterion for exclusion of terms for forward selection, backward elimination and stepwise regression; default 1.0
MAXCYCLE = *scalar*	Limit on number of times to repeat stepwise selection methods, unless no change is made; default 50
CRITERION = *string token*	Criterion for selecting best models among all possible models (r2, adjusted, cp, ep, aic, bic, sic, meandeviance, deviance); default adju
EXTRA = *string token*	Criterion which is also printed for the selected best models (r2, adjusted, cp, ep, aic, bic, sic, meandeviance, deviance); default cp when DISPERSION=*, and mean otherwise
AFACTORIAL = *scalar*	Limit for expansion of FREE model terms for the fitting of all possible models; default 3
PENALTY = *scalar*	Penalty for Mallows Cp and Akaike's information criterion AIC; default 2
NTERMS = *scalar*	Limit on the number of terms to be fitted when fitting all possible models; default 16
NBESTMODELS = *scalar*	Number of best models printed for each subset size; default 8
PPROBABILITY = *scalar*	When METHOD=allpossible, only models with all probabilities less than PPROBABILITY are printed; default 1 i.e. all models are printed
FINALMODELS = *pointer*	Pointer to save the final models for forward, backward, fstepwise and bstepwise regression methods
ALLMODELS = *pointer*	Pointer to save formulae for all possible regression models containing the fitted terms of all the models; every formula includes the FORCED formula if set
ESTIMATES = *pointer*	Pointer to save variates for all possible regression models containing the parameter estimates
SE = *pointer*	Pointer to save variates for all possible regression models containing standard errors of the parameter

	estimates
`RESULTS` = *pointer*	Pointer to save variates for all possible regression models containing the criteria (`r2`, `adjusted`, `cp`, `ep`, `aic`, `sic` or `bic`, `deviance`, `meandeviance`), degrees of freedom for residual and total number of fitted parameters *p*
`STATISTICS` = *pointer*	Pointer to save variates for all possible regression models containing the test statistics. These are F-to-delete statistics (i.e. deviance ratios) when the `DISPERSION` option of the `MODEL` directive is set to `*`, and Chi-square-to-delete statistics (i.e. deviance differences scaled by the dispersion parameter) for a fixed dispersion parameter
`DF` = *pointer*	Pointer to save variates for all possible regression models containing the degrees of freedom for the numerator of the test statistics
`PROBABILITIES` = *pointer*	Pointer to save variates for all possible regression models containing the probabilities of the test statistics
`MARGINALTERMS` = *string token*	How to treat terms that are marginal to other terms in the `FREE` formula (`forced`, `free`); default `forc`

Parameter

`FREE` = *formula*	Model formula specifying the candidate model terms

Description

There are various methods for choosing a regression model when there are many candidate model terms, see e.g. Montgomery & Peck (1992) or Miller (1990). The `STEP` directive provides forward selection, backward elimination and stepwise regression. However these methods result in only one model and alternative models, with an equivalent or even better fit, are easily overlooked. Especially in observational studies with many non-orthogonal terms there are frequently a number of alternative models, and then selection of just one well-fitting model is unsatisfactory and possibly misleading. A preferable method is to fit all possible regression models, and to evaluate these according to some criterion. In this way a number of best regression models can be selected. However the fitting of all possible regression models is very computer intensive. It should also be used with caution, because models can be selected which appear to have a lot of explanatory power, but contain noise variables only, see e.g. Flack & Chang (1987). This may occur particularly when the number of parameters is large in comparison with the number of units, as illustrated by the example for `RSEARCH`. Terms should therefore not be selected on the basis of a statistical analysis alone.

`RSEARCH` can be used to perform these model selection methods. The call to `RSEARCH` must be preceded by a `MODEL` statement which defines the response variate and, if required, all other aspects of a (generalized) linear model. Only one response variate is allowed unless the `DISTRIBUTION` option of `MODEL` is set to `multinomial`. The `FREE` parameter specifies the candidate model terms. These may include variates, factors, interactions and regression functions like `POL` and `SSPLINE`. The `METHOD` option controls which model selection methods are employed:

`accumulated`	prints an accumulated analysis of deviance in which all model terms are added one by one to the model in the given order;
`pooled`	prints an accumulated analysis of deviance in which terms with the same number of identifiers, e.g. main effects or

	two-factor interactions, are pooled;
`forward`	prints an accumulated analysis of deviance resulting from forward selection;
`backward`	prints an accumulated analysis of deviance resulting from backward elimination;
`fstepwise`	prints an accumulated analysis of deviance resulting from stepwise regression starting with no candidate terms in the model;
`bstepwise`	prints an accumulated analysis of deviance resulting from stepwise regression starting with all candidate terms in the model;
`allpossible`	prints summary statistics for a number of best models among all possible models.

For each model with `METHOD=allpossible`, the selection criterion and the degrees of freedom of the included terms are printed. The probability for the hypothesis that an included term can be deleted as the last term is also printed. These probabilities are based on F-to-delete statistics (i.e. deviance ratios) when the `DISPERSION` option of the `MODEL` directive is set to `*`, and Chi-square-to-delete statistics (i.e. deviance differences scaled by the dispersion parameter) for a fixed dispersion parameter.

The `PPROBABILITY` option allows you to reduce the amount of output when `METHOD=allpossible`. If this is set, only models where all the probabilities are less than `PPROBABILITY` are printed. (By default `PPROBABILITY=1`, and so they are all printed.)

It is sometimes desirable to include specific terms in every model. Such terms may be specified by means of the `FORCED` option. The `FORCED` model terms are always fitted first. The `CONSTANT` option controls whether the constant parameter is included in the model. The limit for expanding the `FREE` and `FORCED` model formulae can be set with the `FACTORIAL` option, which has default value 3. The `PRINT` option can be used to control the output from `RSEARCH`.

The criteria for inclusion and exclusion of terms for forward selection, backward elimination and stepwise regression can be specified by the `INRATIO` and `OUTRATIO` options respectively. The `MAXCYCLE` option specifies the number of steps. These operate exactly as in the `STEP` directive. The `DENOMINATOR` option controls the way in which variance ratios are calculated in accumulated analysis of deviance summaries.

All possible regression models are fitted only when the number of candidate `FREE` model terms does not exceed 16. If the `FREE` formula specifies a main effects model, i.e. a model without interactions, the main effects are the candidate terms. When the `FREE` formula contains interactions, the default is to remove any terms marginal to an interaction from the `FREE` formula, and include them instead in the `FORCED` formula. However, you can set option `MARGINALTERMS` to `free` to retain them in `FREE` formula. Note that `RSEARCH` considers only models that obey the principle of marginality. This states that a model that includes an interaction term must also include all its marginal terms. For example, a model that includes the interaction `A.B` must also include the main effects `A` and `B`.

The `AFACTORIAL` option can be used to limit the expansion of the `FREE` model terms for the fitting of all possible regression models. The expansion is limited in addition to the limitation imposed by the `FACTORIAL` option. As an example, the following calls to `RSEARCH` result in identical candidate model terms, namely `a.b`, `a.c`, `b.c` and `d`, for all possible regression models:

```
RSEARCH [METHOD=forward,backward,allpossible;\
         FACTORIAL=3; AFACTORIAL=2] a*b*c + d
RSEARCH [METHOD=forward,backward,allpossible;\
         FACTORIAL=2; AFACTORIAL=2; FORCED=a+b+c] a*b*c + d
```

However, forward selection starts with no terms in the first call and with the model `a+b+c` in the

second call. Backward elimination starts with the full model including the three factor interaction `a.b.c` in the first call, while this term is not fitted in the second call.

The `CRITERION` option controls the selection of the best models among all possible regression models. The criteria employed in `RSEARCH` are defined as follows:

`r2`	$100 \times [1 - \text{Dev} / \text{Dev0}]$
`adjusted`	$100 \times [1 - (\text{Dev} / (n-p)) / (\text{Dev0} / (n-p_0))]$
`cp`	$\text{Dev} / f + 2 \times p - n$
`ep`	$\text{Dev} \times (n+1) \times (n-2) / [n \times (n-p) \times (n-p-1)]$
`aic`	$\text{Dev} / f + 2 \times p$
`sic` or `bic` (synonyms)	$\text{Dev} / f + \text{Ln}(n) \times p$
`deviance`	Dev
`meandeviance`	$\text{Dev} / (n-p)$

where

Dev	is the deviance of the current model;
Dev0	is the deviance of the null model;
p	is the number of fitted parameters of the current model;
p_0	is the number of fitted parameters of the null model;
n	is the number of units;
f	is the dispersion parameter.

The null model is the model with only a constant term, which may include the fitting of a grouping factor for a within groups regression and/or the fitting of cut-points for an ordinal response model.

The dispersion parameter f is specified by the `DISPERSION` option of the `MODEL` directive or, when `DISPERSION is set to *`, is estimated by the mean deviance of the model with all the candidate terms. In ordinary linear regression R^2, adjusted R^2 and Mallows Cp are widely used. When R^2 is used, there is no penalty for adding a term, i.e. R^2 always improves with the addition of a term. When adjusted R^2 or Cp is employed, there is a penalty for adding a term. Adjusted R^2 improves when the F-ratio due to the addition of the term is larger than 1, while Cp improves when the F-ratio is larger than 2. Clearly, Cp is the more conservative criterion and will tend to select models with fewer terms as compared to R^2 and adjusted R^2. Minimizing Cp minimizes the mean squared error of prediction in ordinary linear regression in the case where predictions will be made at the same values as are present in the current data set. Models with negligible bias have Cp » p. For predictions at new random values, as is common in observational studies, Ep estimates the mean squared error of prediction; then Ep should be minimized. Thompson (1978) and Miller (1990) discuss Cp and Ep in detail.

Criteria suggested for generalized linear models are the Akaike information criterion (AIC) and the Schwarz (Bayesian) information criterion (SIC, or its synonym BIC). The definition of both criteria used here is different from that in the literature. The deviance is used instead of the maximum value of the log-likelihood, which implies a constant shift for distributions without dispersion parameter. Moreover, in the spirit of generalized linear models, the deviance is scaled by the dispersion parameter. This makes AIC equivalent to Cp. Clearly, SIC is the more conservative criterion, especially when the number of units is large.

Note that the best models have a small Cp, Ep, AIC, SIC, deviance and mean deviance, but a large R^2 and adjusted R^2. The default penalty of 2 in the definition of Cp and AIC can be altered by setting the `PENALTY` option, in which case Cp and AIC improves when the F-ratio is larger than `PENALTY`. The `EXTRA` option specifies an extra criterion which is printed alongside the selection criterion. The default for `CRITERION` is `adjusted`. The default for `EXTRA` is `cp` when `DISPERSION` is set to `*`, and `meandeviance` otherwise.

The `NTERMS` option specifies the maximum number of candidate terms in a model. This can be used when only models with few candidate terms are relevant or to reduce the computational burden. For example with 12 candidate terms there are 4096 different models, while there are

only 299 models with maximally three terms. Specifying NTERMS=3 then saves a considerable amount of computing time. The NBESTMODELS option specifies the number of best models within each subset size for which summary statistics are printed.

The FINALMODEL option can be used to save the last models for forward selection, backward elimination and fstepwise and bstepwise regression. Results of the fitting of all possible regression models can be saved by means of the parameters ALLMODELS, ESTIMATES, SE, RESULTS, STATISTICS, DF and PROBABILITIES. This saves results from all the fitted models not only from those that are printed. This includes the constant model.

All regression warnings are suppressed. This is to prevent the printing of long lists of similar warnings like "Iterative weights have become 0, or have been held at a limit". Note that the printed output of all possible regression models is adjusted to the width of the output file.

Options: PRINT, METHOD, FORCED, CONSTANT, FACTORIAL, DENOMINATOR, INRATIO, OUTRATIO, MAXCYCLE, CRITERION, EXTRA, AFACTORIAL, PENALTY, NTERMS, NBESTMODELS, PPROBABILITY, FINALMODELS, ALLMODELS, ESTIMATES, SE, RESULTS, STATISTICS, DF, PROBABILITIES, MARGINALTERMS.
Parameters: FREE.

Method

First the FREE and FORCED formulae are checked using subsidiary procedure _RSEARCHCHECK, and terms that appear in both are dropped from the FREE formula. Then the full model is fitted and aliased predictors are dropped from both formulae. Forward selection, backward elimination and stepwise regression are straightforward implemented using the STEP directive.

The fitting of all possible regression models uses a sequence of models in which, within each subset size, every model is fitted by dropping one term from the previous model and adding another term. Test statistics are calculated as though the tested term is the last term to enter the model. When the DISPERSION option of the MODEL directive is set to *, terms are tested by means of F-to-delete statistics, which are deviance ratios. For a fixed dispersion parameter Chi-square-to-delete statistics, i.e. deviance differences scaled by the dispersion parameter, are used to calculate probabilities.

Smoothing splines are not allowed in the FREE formula for METHOD=allpossible due to a limitation of the FCLASSIFICATION directive.

Action with RESTRICT

Factors and variates in the FREE and FORCED formulae should not be restricted. Any restriction applied to vectors used in the MODEL statement applies also to the results from RSEARCH.

References

Flack, V.F. & Chang, P.C. (1987). Frequency of selecting noise variables in subset regression analysis: a simulation study. *The American Statistician*, **41**, 84-86.

Miller, A.J. (1990). *Subset Selection in Regression*. Chapman & Hall, London.

Montgomery, D.C. & Peck, E.A. (1992). *Introduction to Linear Regression Analysis (second edition)*. Wiley, New York.

Thompson, M.L. (1978). Selection of variables in multiple regression: Part I. A review and evaluation. *International Statistical Review*, **46**, 1-19.

See also

Directive: STEP.
Procedures: ASCREEN, RSCREEN, RWALD.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RSPREADSHEET

Puts results from a regression, generalized linear or nonlinear model into a spreadsheet (R.W. Payne).

Options

DISPERSION = *scalar*	Dispersion parameter to be used as estimate for variability in s.e.s; default as set in MODEL
RMETHOD = *string token*	Type of residual to use (deviance, Pearson, simple, deletion); default * i.e. as set in MODEL
DMETHOD = *string token*	basis of estimate of dispersion, if not fixed by DISPERSION option (deviance, Pearson); default * i.e. as set in MODEL
SPREADSHEET = *string tokens*	Which spreadsheets to form (summary, estimates, fittedvalues, accumulated); default summary, estimates, fittedvalues
SPESTIMATES = *string tokens*	What to include in the estimates spreadsheet (estimates, se, testimates, prestimates); default esti, se, test, pres
SPFITTEDVALUES = *string tokens*	What to include in the fitted-values spreadsheet (y, fittedvalues, residuals, leverages, sefittedvalues); default y, fitt, resi, leve
SAVE = *regression save structure*	Specifies which analysis to save; default * i.e. most recent regression

Parameters

Y = *variates*	Y-variate of the analysis to be saved
RESIDUALS = *variates*	Identifier of variate to save the residuals from each analysis; default residuals
FITTEDVALUES = *variates*	Identifier of variate to save the fitted values from each analysis; default fittedvalues
LEVERAGES = *variates*	Identifier of variate to save the leverages from each analysis; default leverages
ESTIMATES = *variates*	Identifier of variate to save the estimates from each analysis; default estimates
SE = *variates*	Identifier of variate to save s.e.'s of the estimates from each analysis; default se
TESTIMATES = *variates*	Identifier of variate to save the t-statistics of the estimates from each analysis; default t_statistics
PRESTIMATES = *variates*	Identifier of variate to save the t-probabilities of the estimates from each analysis; default t_probabilities
SEFITTEDVALUES = *variates*	Identifier of variate to save s.e.'s of the fitted values from each analysis; default sefittedvalues
SUMMARY = *pointers*	Identifier of pointer to save the summary analysis-of-variance (or deviance) from each analysis; default summary
ACCUMULATED = *pointers*	Identifier of pointer to save the accumulated analysis-of-variance (or deviance) from each analysis; default accumulated
OUTFILENAME = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create

Description

RSPREADSHEET puts results from a regression, generalized linear or nonlinear model into a spreadsheet. By default the results are from the most recent regression, but you use the SAVE option to specify the save structure (from a MODEL statement) from some other analysis. You can use the Y parameter to indicate the y-variate, if the SAVE structure contains results from more than one.

The SPREADSHEET option specifies which pages of the spreadsheet to form, with settings:

summary	summary analysis of variance (or deviance for a generalized linear model),
estimates	estimates with the standard errors etc.,
fittedvalues	fitted values, y-variate, residuals etc., and
accumulated	summary analysis of variance (or deviance for a generalized linear model).

By default, SPREADSHEET=summ,esti,fitt.

The SPESTIMATES option specifies which columns to include in the estimates spreadsheet, with settings:

estimates	estimates,
se	standard errors of estimates,
testimates	t-statistics of of estimates, and
prestimates	t-probabilities of estimates.

By default they are all included.

The SPFITTEDVALUES option specifies which columns to include in the estimates spreadsheet, with settings:

y	y-variate,
fittedvalues	fitted values,
residuals	residuals,
leverages	leverages, and
sefittedvalues	standard errors of fitted values.

By default SPFITTEDVALUES=y,fitt,resi,leve.

To help avoid clashes between the columns of the spreadsheets if you want to save results from more than one analysis, the parameters RESIDUALS, FITTEDVALUES, LEVERAGES, ESTIMATES, SE, TESTIMATES, PRESTIMATES, SEFITTEDVALUES, SUMMARY, ACCUMULATED allow you to specify identifiers for the columns (or sets of columns) that will store the corresponding results in the current spreadsheets. Their defaults are mainly the same as the parameter names, but in lower-case letters. The exceptions are that TESTIMATES and PRESTIMATES have defaults t_statistics and t_probabilities, respectively.

You can save the data in either a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx), by setting the OUTFILENAME option to the name of the file to create. If the name is specified without a suffix, '.gwb' is added (so that a Genstat workbook is saved). If OUTFILENAME is not specified, the data are put into a spreadsheet opened inside Genstat.

Options: DISPERSION, RMETHOD, DMETHOD, SPREADSHEET, SPESTIMATES, SPFITTEDVALUES, SAVE.

Parameters: Y, RESIDUALS, FITTEDVALUES, LEVERAGES, ESTIMATES, SE, TESTIMATES, PRESTIMATES, SEFITTEDVALUES, SUMMARY, ACCUMULATED, OUTFILENAME.

Action with RESTRICT

If the Y variate is restricted, that restriction will carry over into the fitted-values spreadsheet.

See also
Directive: SPLOAD.
Procedures: ADSPREADSHEET, ASPREADSHEET, AUSPREADSHEET, FSPREADSHEET, VSPREADSHEET.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RSTEST

Compares groups of right-censored survival data by nonparametric tests (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (test); default test
METHOD = *string tokens*	Types of test required (logrank, breslow, petoprentice, taroneware); default logr, bres, peto, taro
BLOCKS = *factor*	Factor specifying groupings for a stratified test; default * i.e. none

Parameters

TIMES = *variates*	Observed timepoints
CENSORED = *variates*	Variate specifying whether the corresponding element of TIMES is censored (1) or not (0)
GROUPS = *factors*	Factor specifying the different groups
TESTS = *pointers*	Pointer to variates (length 3) to save test statistic, d.f. and probability value for each chosen method

Description

RSTEST compares two or more groups of right-censored survival data using nonparametric tests. The type of test to be performed is specified by the METHOD option, with settings logrank, breslow, petoprentice and taroneware.

The observed timepoints or the timepoints at which censoring took place are specified using the TIMES parameter. The CENSORED parameter specifies a variate containing the value one if the corresponding element of TIMES is censored or zero if it was not. CENSORED can be omitted if there was no censoring. The groups to be compared are indicated using the GROUPS parameter. The BLOCKS option can be used to specify a factor to indicate different groupings for a stratified test, for example these might represent different centres or laboratories.

The TESTS parameter allows the statistics to be saved in a pointer to a set of variates (length 3) for each of the chosen methods containing the statistic, its degrees of freedom and probability level. If you are saving the tests you may want to set option PRINT=* to stop them being printed.

Options: PRINT, METHOD, BLOCKS.
Parameters: TIMES, CENSORED, GROUPS, TESTS.

Method

The log-rank and Wilcoxon (Breslow) tests are calculated according to the method outlined in Chapter 2 of Collet (1994). The Wilcoxon (Peto-Prentice) and Tarone-Ware tests are evaluated using the method detailed in Section 11.1.2 of Collet (1994).

Action with RESTRICT

The input variates and factors may be restricted identically. The tests are based only on the units not excluded by the restriction.

Reference

Collett, D. (1994). *Modelling Survival Data in Medical Research*. Chapman & Hall, London.

See also
Procedures: KAPLANMEIER, RLIFETABLE, RPHFIT, RPROPORTIONAL, RSURVIVAL.
Genstat Reference Manual 1 Summary section on: Survival analysis.

RSTORE

Stores a regression save structure in an external file (R.W. Payne).

No options

Parameters

FILENAME = *texts*	Name of the file to store the save structure
EXIT = *scalars*	Scalar that contains the value one if the save structure could not be stored successfully, otherwise zero
SAVE = *regression save structures*	Save structure to be stored; default stores the save structure from the most recent regression analysis

Description

RSTORE stores a regression save structure in an external file. It can then be loaded back into Genstat in a later run, by the RRETRIEVE procedure, so that further output can be produced from the analysis. (See, for example, directives RDISPLAY and RKEEP, or procedures RCHECK, RGRAPH and RSPREADSHEET.)

The name (and path) of the file to store the save structure is specified, in a text, by the FILENAME parameter. The save structure is specified by the SAVE parameter. If this is unset, ASTORE stores the save structure from the most recent regression. The EXIT parameter can return a scalar containing the value one if the save structure could not be stored successfully. Otherwise it contains zero.

Options: none.
Parameters: FILENAME, EXIT, SAVE.

Method

The save structure is stored in a Genstat backing-store file by the STORE directive.

See also

Directive: FIT.
Procedures: ASTORE, RRETRIEVE.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RSURVIVAL

Models survival times of exponential, Weibull, extreme-value, log-logistic or lognormal distributions (R.W. Payne & D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated, loglikelihood); default mode, summ, esti
TIMES = *variate*	Time of each observation
DISTRIBUTION = *string token*	Distribution of the survival times (exponential, weibull, extremevalue, loglogistic, lognormal); default expo
CENSORED = *variate*	Indicator for censored observations: 0 if uncensored, 1 if right censored (subject survived the whole trial), −1 if left censored (log-logistic distribution only); default assumes no censored observations
PLOT = *string token*	What to plot (survivorfunction); default *
GRAPHICS = *string token*	Type of graphics (lineprinter, highresolution) default high
ALPHA = *scalar*	Saves the estimated value of the parameter α of the Weibull and extreme-value distributions, if the scalar is input with a non-missing value this provides the initial estimate for α (which will also be the final estimate if MAXCYCLE=1)
_2LOGLIKELIHOOD = *scalar*	Saves −2 multiplied by the log-likelihood
SIGMA = *scalar*	Saves the estimated value of the shape parameter sigma of the log-logistic and lognormal distributions
SURVIVOR = *variate*	Saves estimates of the survivor function
PARAMETERIZATION = *string token*	Controls the parameterization used when saving the survivor function for the Weibull distribution (ph, aft); default ph
MAXCYCLE = *scalar*	Maximum number of iterations to use to estimate α; default 20
TOLERANCE = *scalar*	Convergence limit for α; default 10^{-5}

Parameter

TERMS = *formula*	Defines the model to fit

Description

RSURVIVAL models survival times assuming that they follow either an exponential, Weibull, extreme-value, log-logistic or lognormal distribution, as indicated by the DISTRIBUTION option. It also caters for right-censored observations, where the subject concerned survived the trial: the CENSORED option can be used to specify a variate with an entry for each subject containing one where the subject survived, otherwise zero. The log-logistic caters for left-censored observations, which they can be specified by an entry of −1 in the CENSORED variate. The model to be fitted to the survival times is specified using the TERMS parameter.

The analysis is performed using the generalized linear models facilities of Genstat. For the exponential, Weibull and extreme-value distributions a y-variate (= 1 − CENSORED) is specified indicating whether the subject died or survived, and an offset variate is included which depends

on the time variate (see Chapter 6 of Aitkin *et al*. 1989). For the exponential distribution this offset is simply the logarithm of the times. With the Weibull distribution it is the Weibull parameter α multiplied by the logarithm of the times, while for the extreme-value distribution it is the parameter α multiplied by the times. The parameters of the TERMS model and α itself are estimated alternately (with number of cycles controlled by the MAXCYCLE option) until successive estimates are within a tolerance specified by the TOLERANCE option. The ALPHA option can input an initial value for α and save the estimated value. By setting the MAXCYCLE option to one, α can be fixed at the initial value; this is useful for comparing one model with another, when the value of α should be fixed at the value estimated from the more complicated model. The log-logistic distribution is fitted using a logistic regression model with number of successes $1-c$ and binomial denominator $2-c-b$ (where c is an index for a right-censored observation and b is an index for a left-censored observation) using an offset variate of the logarithm of times divided by σ. The parameters of the TERMS model and σ (shape parameter) are estimated alternately (with number of cycles controlled by the MAXCYCLE option) until successive estimates are within a tolerance specified by the TOLERANCE option. For the lognormal distribution maximization of the log-likelihood is achieved using an EM algorithm details of which are given in Section 6.19 of Aitkin *et al*. (1989). The SIGMA option can be used to save the estimated value of the shape parameter for both the log-logistic and lognormal distributions. The importance of variables in the lognormal model should be assessed by omitting the variable and comparing −2 times the log-likelihood; this can be saved using the _2LOGLIKELIHOOD option. The SURVIVOR option allows you to save estimates of the survivor function. For the Weibull distribution the PARAMETERIZATION option can be used to choose whether to produce the estimates for the survivor function using the proportional hazards or accelerated failure time parameterization.

The PRINT option controls printed output with similar settings to those of the FIT directive, except that there is an extra setting loglikelihood to print −2 times the log-likelihood. Further information can be printed subsequently by using RDISPLAY in the usual way. The PLOT option can be set to survivorfunction to produce plots of the empirical survivor function against the value predicted by the model, when the exponential, Weibull and extreme-value distributions are selected (see Aitken *et al*. 1989, pages 275-276). The GRAPHICS option determines the type of graph, with settings highresolution (the default) or lineprinter.

Options: PRINT, TIMES, DISTRIBUTION, CENSORED, PLOT, GRAPHICS, ALPHA, _2LOGLIKELIHOOD, SIGMA, SURVIVOR, PARAMETERIZATION, MAXCYCLE, TOLERANCE.
Parameter: TERMS.

Method

Full details of the method can be found in Chapter 6 of Aitkin *et al*. (1989). For the exponential distribution (pages 269-270), the survivor function is

$$S(t) = \exp(-\lambda t)$$

with

$$\lambda = \exp(\Sigma(b_i x_i))$$

where b_i are the parameter estimates, x_i are the appropriate values of the explanatory variates, and t is the time. The Weibull distribution (page 280) is defined with density function

$$f(t) = \alpha \lambda t^{(\alpha-1)} \exp(-\lambda (t^{\alpha}))$$

and has survivor function

$$S(t) = \exp(-\lambda t^{\alpha}).$$

The extreme-value distribution (pages 283-284) has survivor function

$$S(t) = \exp(-\lambda \exp(\alpha t)).$$

The loglogistic distribution (pages 295-297) has the survivor function

$S(t) = 1 / \{ 1 + (t / \theta)^a \}$

with

$\theta = \exp(\sum(b_i \times x_i))$

and $a = 1 / \sigma$.

The lognormal distribution (pages 297-300) has survivor function

$S(t) =$ CUNORMAL(log($(t - \sum(b_i \times x_i)) / \sigma$))

Action with RESTRICT

The vectors involved in the analysis may be restricted as usual for a generalized linear model.

Reference

Aitkin, M., Anderson, A., Francis, B. & Hinde, J. (1989). *Statistical Modelling in GLIM*. Oxford University Press.

See also

Procedures: KAPLANMEIER, RLIFETABLE, RPHFIT, RPROPORTIONAL, RSTEST.

Genstat Reference Manual 1 Summary section on: Survival analysis.

RTCOMPARISONS

Calculates comparison contrasts within a multi-way table of means (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (contrasts); default cont
COMBINATIONS = *string token*	Factor combinations for which to form the predicted means (full, present, estimable); default esti
ADJUSTMENT = *string token*	Type of adjustment to be made when forming the predicted means (marginal, equal, observed); default marg
WEIGHTS = *table*	Weights classified by some or all of the factors in the model; default *
OFFSET = *scalar*	Value of offset on which to base predictions; default mean of offset variate
METHOD = *string token*	Method of forming margin (mean, total); default mean
ALIASING = *string token*	How to deal with aliased parameters (fault, ignore); default faul
BACKTRANSFORM = *string token*	What back-transformation to apply to the values on the linear scale, before calculating the predicted means (link, none); default link
SCOPE = *string token*	Controls whether the variance of predictions is calculated on the basis of forecasting new observations rather than summarizing the data to which the model has been fitted (data, new); default data
NOMESSAGE = *string tokens*	Which warning messages to suppress (dispersion, nonlinear); default *
DISPERSION = *scalar*	Value of dispersion parameter in calculation of s.e.s; default is as set in the MODEL statement
DMETHOD = *string token*	Basis of estimate of dispersion, if not fixed by DISPERSION option (deviance, Pearson); default is as set in the MODEL statement
NBINOMIAL = *scalar*	Supplies the total number of trials to be used for prediction with a binomial distribution (providing a value *n* greater than one allows predictions to be made of the number of "successes" out of *n*, whereas the value one predicts the proportion of successes); default 1
SAVE = *identifier*	Regression or ANOVA save structure for the analysis from which the comparisons are to be calculated

Parameters

CONTRAST = *tables*	Defines the comparisons to be estimated
ESTIMATES = *scalars*	Saves the estimated contrasts
SE = *scalars*	Saves standard errors of the contrasts

Description

RTCOMPARISONS makes comparisons within multi-way tables of predicted means from a linear or generalized linear regression or an analysis of variance. The model should previously have been fitted by the FIT or ANOVA directives in the usual way. The SAVE option can be used to specify the save structure from the analysis for which the comparisons are to be calculated (see the SAVE option of the MODEL or ANOVA directives). If SAVE is not specified, the comparisons

are calculated from the most recent regression analysis.

Each comparison is specified in a table supplied by the CONTRAST parameter. For a regression or generalized linear models analysis, RTCOMPARISONS calculates the means using the PREDICT directive. The first step (A) of the calculation forms the full table of predictions, classified by every factor in the model. The second step (B) averages the full table over the factors that do not occur in the table of means. The COMBINATIONS option specifies which cells of the full table are to be formed in Step A. The default setting, estimable, fills in all the cells other than those that involve parameters that cannot be estimated, for example because of aliasing. Alternatively, setting COMBINATIONS=present excludes the cells for factor combinations that do not occur in the data, or COMBINATIONS=full uses all the cells. The ADJUSTMENT option then defines how the averaging is done in Step B. The default setting, marginal, forms a table of marginal weights for each factor, containing the proportion of observations with each of its levels; the full table of weights is then formed from the product of the marginal tables. The setting equal weights all the combinations equally. Finally, the setting observed uses the WEIGHTS option of PREDICT to weight each factor combination according to its own individual replication in the data. Alternatively, you can supply your own table of weights, using the WEIGHTS option. The COMBINATIONS and ADJUSTMENT options are irrelevant if a SAVE structure is from an ANOVA analysis – the means are then obtained using AKEEP (and correspond to those that would be printed by ANOVA). The options OFFSET, METHOD, ALIASING, BACKTRANSFORM, SCOPE, NOMESSAGE, DISPERSION, DMETHOD and NBINOMIAL are also relevant only to regression, and operate exactly as in the PREDICT directive.

The PRINT option controls printed output, with setting:

contrasts	to print the contrasts (default).

The ESTIMATE parameter allows you to save the estimated contrast, and the SE parameter can save its standard error.

Options: PRINT, COMBINATIONS, ADJUSTMENT, WEIGHTS, OFFSET, METHOD, ALIASING, BACKTRANSFORM, SCOPE, NOMESSAGE, DISPERSION, DMETHOD, NBINOMIAL, SAVE.
Parameters: CONTRAST, ESTIMATE, SE.

Method

The predicted means and their variances and covariances are obtained using the PREDICT directive for a regression analysis, or using AKEEP for an analysis of variance. The comparisons and their standard errors are then calculated using Genstat's table and matrix calculation facilities.

See also

Directive: PREDICT.
Procedures: FCONTRASTS, RCOMPARISONS, VTCOMPARISONS.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RUGPLOT

Draws "rugplots" to display the distribution of one or more samples (P.W. Lane).

Options

GRAPHICS = *string token*	What type of graphics to use (highresolution, lineprinter); default high
TITLE = *text*	Title for diagram; default *
AXISTITLE = *text*	Title for axis; default *
WINDOW = *scalar*	Window in which to draw high-resolution plot; default *, taken as 11 if SCREEN=clear, or 1 if SCREEN=keep
SCREEN = *string token*	Whether to clear screen before high-resolution plot (clear, keep); default clea
ORIENTATION = *string token*	Orientation of plots (down, across); default down
JITTER = *number*	Ratio of jitter width to range of data in high-resolution plot; default 0.01
SEED = *number*	Seed for generating random numbers used in jittering; default 0, i.e. continue from last generation, or initialize from system clock

Parameters

DATA = *variates*	Data to be summarized; no default
GROUPS = *factor*	Factor to divide values of a single variate into groups; default *
RUGLABELS = *texts*	Labels for individual rugs; default *, i.e. identifiers of variates or labels or levels of factor
POSITION = *scalar* or *variate*	Position on *x*-axis (or on *y*-axis if ORIENTATION=across) at which to plot each rug; if GROUPS is set, positions for each level of the factor are taken from a variate; default is to draw a single rug on the axis, and to spread multiple rugs across the window

Description

RUGPLOT draws pictures to display the distribution of one or more sets of data. In the simplest case, with the DATA parameter set to a single variate, RUGPLOT will draw a single vertical "rug": that is, a series of short horizontal lines on the vertical axis, positioned at each value of the variate. The option ORIENTATION=across produces a horizontal rug. A rug can be added to an existing plot by specifying SCREEN=keep, and setting the WINDOW option to specify the window where the rug is to be drawn. With SCREEN=keep, the default window is 1; with SCREEN=clear, window 11 is used after defining it to fill the whole graphical frame.

If several variates are supplied, a rug is drawn for each of them using the same scale. Alternatively, if a single variate is specified by the DATA parameter, a factor with the same number of values as the variate may be defined by the GROUPS parameter, and a box will be drawn for each level of the factor. The rug plots are spread out across the window by default. The POSITION parameter can be set to specify where each rug is to be positioned on the *x*-axis (or *y*-axis if ORIENTATION=across). The setting should be in the range (0, *n*) for a plot with SCREEN=clear, where n is the number of rugs to be drawn; with SCREEN=keep, the position should be specified in the units of the axis last drawn in the window.

Line-printer rugplots can be drawn by setting option GRAPHICS=lineprinter. The plot is drawn with asterisks, or digits to represent points that are effectively coincident. If the page size is small, as in interactive mode, line-printer plots with ORIENTATION=down are very cramped: the PAGE option of the OUTPUT directive can be used to increase the depth of the graphs. The

option ORIENTATION=down cannot be selected for line-printer plots with more than 14 rugs.

The TITLE and AXISTITLE options can be set to specify the titles displayed at the top of the plot and along the axis, for either graphics mode. The RUGLABELS parameter allows you to specify labels that will identify each rug, in place of the default labels taken from the variate identifiers, or factor labels or levels if the GROUPS parameter is set. Long identifiers or labels may overlap each other if ORIENTATION=down, or they may overlap the rug-plots if ORIENTATION=across; a maximum of eight characters is recommended.

In high-resolution plots, all data values are "jittered" to try to remove ties. This involves adding a small random value: by default the ratio of the maximum adjustment to the range of all the data is 1:100. This can be modified by setting the JITTER option to 0 to suppress jittering, or to some other ratio than the default of 0.01. The SEED option can be set to specify the seed of the random-number generation, if a reproducible plot is required.

Options: GRAPHICS, TITLE, AXISTITLE, WINDOW, SCREEN, ORIENTATION, JITTER, SEED.
Parameters: DATA, GROUPS, RUGLABELS, POSITION.

Method

High-resolution rugs are plotted using the minus or vertical-bar symbol, in vertical or horizontal plots respectively. Line-printer plots use the default plotting symbols, the asterisk or digits to represent coincident points.

Action with RESTRICT

Restrictions on the supplied variates are taken into account. The grouping factor and texts holding ruglabels, if specified, should not be restricted.

See also

Directive: DHISTOGRAM.
Procedures: BOXPLOT, DOTPLOT, STEM.
Genstat Reference Manual 1 Summary section on: Graphics.

RUNTEST

Performs a test of randomness of a sequence of observations (P.W. Goedhart).

Options

PRINT = *string token*	Controls printed output (results); default resu
NULL = *scalar*	Defines the boundary between the two types; default 0

Parameters

DATA = *variates*	Sequences of observations
SAVE = *pointers*	To save the number of runs, the number of positive and negative observations and the lower and upper tail probabilities of the test

Description

The data are assumed to be in an ordered sequence of observations of two types, n_1 of the first type and n_2 of the second type. A run is defined to be a succession of observations of the same type. A clue to lack of randomness is provided by the total number of runs in the sequence. If the data are in random order, the expected number of runs is $1 + 2n_1n_2/(n_1+n_2)$. A low number of runs might indicate positive serial correlation while a high number might arise from negative serial correlation.

The DATA parameter is used to specify the sequence of observations. Observations larger than option NULL are considered to be of the first type (positive) while observation smaller than NULL are of the second type (negative). Missing values and observations that equal NULL are not taken into account. The PRINT option controls printed output, while the SAVE parameter can be used to specify a pointer containing five scalars to save the number of runs, the number of positive observations (that is, those larger than NULL), the number of negative observations and the lower and upper tail probabilities of the number of runs.

Options: PRINT, NULL. Parameters: DATA, SAVE.

Method

When the number of observations of type one and two are both smaller than 11, exact left and right tail probabilities are taken from Table 3.1 from Draper & Smith (1981). In other cases a normal approximation with continuity correction is used.

Action with RESTRICT

The DATA variate can be restricted so that the test uses only a subset of the units.

Reference

Draper & Smith (1981). *Applied Regression Analysis (second edition)*. Wiley, New York.

See also

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

RWALD

Calculates Wald and F tests for dropping terms from a regression (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (waldtests); default wald
FACTORIAL = *scalar*	Limit on number of factors in the model terms generated from the TERMS parameter; default 3
Y = *variate*	Y-variate from whose analysis to calculate the statistics; default is the last y-variate in SAVE
RDF = *scalar*	Saves the residual d.f. used to calculate F probabilities when the dispersion is not fixed
SAVE = *regression save structure*	Specifies the save structure (from MODEL) containing the analysis for which to calculate the tests; default is the save structure from the most recent regression

Parameters

TERMS = *formula*	Model terms for which tests are required
WALDSTATISTIC = *scalar* or *pointer* to *scalars*	Saves Wald statistics
DF = *scalar* or *pointer* to *scalars*	Saves d.f. of Wald statistics
PROBABILITY = *scalar* or *pointer* to *scalars*	Saves the probabilities for the Wald statistics if the dispersion is fixed, or the corresponding F statistics if it is estimated

Description

RWALD provides Wald tests to help you decide whether any terms can be dropped from a regression model. The model must have been fitted already by the regression commands (MODEL, FIT etc.) in the usual way. The tests are usually produced for the most recent regression analysis, but you can set the SAVE and Y options to request tests from an earlier analysis.

By default, RWALD produces tests for all the terms that can be dropped from the model: that is, for every term that is not marginal to another term in the model. For example, in the formula

```
A + B + C + D + A.B + A.D + B.D
```

the terms C, A.B, A.D and B.D can be dropped as there are no other terms in the model that contain all their factors (i.e. none to which they are marginal). However, A cannot be dropped until A.B and A.D have been dropped. You can use the TERMS parameter to request Wald tests for a specific set of terms. A missing value is then given for any term that cannot be dropped. The FACTORIAL option sets a limit on the number of factors or variates in each term that is formed from the TERMS formula (default 3).

If option PRINT=waldtests (the default), RWALD prints a table with columns containing the Wald statistic, its number of degrees of freedom and a probability value. With an ordinary linear regression, RWALD will also print an F statistic, and use this to obtain the probability. Provided there is no aliasing between the parameters of the terms, these F statistics and probabilities will be identical to those that would be printed in the Change lines of the Summary of Analysis if the terms were dropped from the model explicitly by using the DROP or TRY directives. The advantage of RWALD is that the model does not have to be refitted (excluding each term) to calculate the information. It thus provides a much more efficient method of assessing the model.

F statistics are also given with any generalized linear model in which the dispersion is not fixed (e.g. models involving the gamma distribution). However, in generalized linear models with a fixed dispersion (e.g. binomial or Poisson), the probabilities are obtained by treating the Wald statistics as chi-square statistics. The deviances and deviance ratios used by TRY and DROP

are calculated from the likelihoods of the generalized linear models, whereas the Wald and F statistics are essentially based on weighted sums of squares. So probabilities calculated by RWALD will no longer be identical to those given by TRY and DROP. However, both sets of probabilities are based on the asymptotic properties of their statistics, and so they should give similar conclusions.

The WALDSTATISTIC parameter can save the statistics, and the DF parameter can save their numbers of degrees of freedom. If you are making a Wald test for a single term, you can supply a scalar for each of these parameters. However, if you have several terms, you must supply a pointer which will then be set up to contain as many scalars as there are terms. Similarly the PROBABILITY parameter saves the probabilities for the Wald statistics if the dispersion is fixed, or the corresponding F statistics if it is estimated. The number residual degrees of freedom for the F statistics can be saved, in a scalar, by the RDF option. This contains a missing value if the dispersion is fixed.

Options: PRINT, FACTORIAL, Y, RDF, SAVE.
Parameters: TERMS, WALDSTATISTIC, DF, PROBABILITY.

Method
RWALD uses FCLASSIFICATION to form the list of terms that can be dropped. It then calculates the statistics using estimates and variances saved using RKESTIMATES.

See also
Procedures: ASCREEN, RSCREEN.
Genstat Reference Manual 1 Summary section on: Regression analysis.

RXGENSTAT

Submits a set of commands externally to R and reads the output (M.F. D'Antuono & D.A. Murray).

Options

PRINT = *string tokens*	Controls printed output (summary, output); default outp
RPATH = *text*	Path specifying the location of the R executable
REXE = *text*	Name of the R executable to run; default 'Rterm.exe'
RARGS = *text*	Command line arguments to be used with the R executable; default '--no-restore --no-save'
SCRIPT = *text*	A set of R commands to run within R
SFILE = *text*	A file containing a set of R commands to run within R
RGEN = *text*	Name of a file to save the full set of commands used within R
ROUT = *text*	Name of a file to save the output from R

Parameters

WORKDIRECTORY = *texts*	Working directory to use within R; default current Genstat working directory
IDATA = *pointers*	Pointer to data structures to export to R (the data are exported into the file specified by the IRDAFILE parameter)
IRDAFILE = *texts*	Name of an R data (rda) file to import into R
ISAVE = *texts*	Pointer to data structures to import from R (the data are imported from the file specified by the ORDAFILE parameter)
ORDAFILE = *text*	Name of an R data (rda) file used to export data from R

Description

RXGENSTAT allows a set of commands to be submitted externally to R and can read output generated from the run. To use RXGENSTAT, the R software must be installed on the current system. The R executable used when submitting a script is specified using the REXE option, by default this uses the R for Windows terminal front-end executable (Rterm.exe). The location of the R executable (usually the bin directory of the R installation) used to run the R script should be specified using the RPATH option. The directory for the path should be specified as a text containing the absolute pathname, for example in Windows the default directory for the executables for R version 2.3 would be

```
C:/Program Files/R/R-2.3.1/bin
```

Additional command line arguments to be used when submitting commands can be supplied using the RARGS option, by default '--no-restore --no-save'.

Data can be exported from Genstat to R by supplying a pointer to the data structures using the IDATA parameter. The data structures can either be factors, variates and texts of equal length, scalars, or a matrix. RXGENSTAT saves the data to an R data (rda) file using the EXPORT procedure and then the data can be accessed within the R script in the usual way. For example, data from a file called idata.rda could be accessed as idata$x etc. The name of the R data file should be supplied using the IRDAFILE parameter. Data can be imported back into Genstat by saving data within R into a rda file and then specifying the name of this R data file in Genstat using the ORDAFILE parameter. The ISAVE parameter can also be used to save a pointer to the imported data structures. A set of R commands can either be supplied within a text using the SCRIPT option or within an R script file (.r) using the SFILE option.

To execute commands within R, RXGENSTAT creates an R script file (.r) that contains the commands for setting the working directory, loading any data and running additional R commands (supplied using the SCRIPT or SFILE option). This file can be saved using the RGEN option. The output generated by R can also be saved using the ROUT option. By default, the working directory will be the current directory, however, an alternative directory can be supplied using the WORKDIRECTORY parameter.

The PRINT option controls printed output, with the settings:

summary	to print a summary of any data that are imported, and
output	to print the R output.

Options: PRINT, RPATH, REXE, RARGS, SCRIPT, SFILE, RGEN, ROUT.
Parameters: WORKDIRECTORY, IDATA, IRDAFILE, ISAVE, ORDAFILE.

Method
In Windows the commands are submitted to R by creating a bat file containing a command line and then executing this within a windows command processor.

Action with RESTRICT
Any data restrictions will be ignored.

See also
Directive: SUSPEND.
Procedure: A2RDA, BGXGENSTAT.
Genstat Reference Manual 1 Summary section on: Program control.

RYPARALLEL

Fits the same regression model to several response variates, and collates the output (P. Brain, R.W. Payne & D.B. Baird).

Options

`PRINT` = *string tokens*	Controls printed output (`model`, `summary`); default `*` i.e. none
`TERMS` = *formula*	Defines the regression model to fit on each variate
`WEIGHTS` = *variate* or *symmetric matrix*	Weights for the regression; default 1
`OFFSET` = *variate*	Offset; default `*` i.e. none
`CONSTANT` = *string token*	How to treat the constant (`estimate`, `omit`); default `esti`
`FACTORIAL` = *scalar*	Limit for expansion of model terms; default 3
`FULL` = *string token*	Whether to assign all possible parameters to factors and interactions (`yes`, `no`); default `no`
`POOL` = *string token*	Whether to pool the information on each term in the analysis of variance (`yes`, `no`); default `no`
`RMETHOD` = *string token*	Type of residuals to form (`deviance`, `Pearson`, `simple`); default `devi`
`SPREADSHEET` = *string tokens*	What results to save in a book of spreadsheets (`aov`, `residuals`, `fittedvalues`, `estimates`, `se`, `testimates`, `prestimates`); default `*` i.e. none

Parameters

`Y` = *variates* or *pointers*	Y-values for each set of analyses
`RESIDUALS` = *matrices*	Saves residuals from each set of analyses
`FITTEDVALUES` = *matrices*	Saves fitted values from each set of analyses
`ESTIMATES` = *matrices*	Saves estimates from each set of analyses
`SE` = *matrices*	Saves s.e.'s of estimates
`TESTIMATES` = *matrices*	Saves t-statistics of estimates
`PRESTIMATES` = *matrices*	Saves t-probabilities of estimates
`DF` = *pointers*	Saves degrees of freedom for the model terms or variates in each analysis of variance
`SS` = *pointers* or *variates*	Saves sums of squares for the model terms in each analysis of variance
`MS` = *pointers* or *variates*	Saves mean squares for the model terms in each analysis of variance
`RDF` = *variates*	Saves degrees of freedom from the "residual" lines in each analysis of variance
`RSS` = *variates*	Saves sums of squares from the "residual" lines
`RMS` = *variates*	Saves mean squares from the "residual" lines
`TDF` = *variates*	Saves degrees of freedom from the "total" lines in each analysis of variance
`TSS` = *variates*	Saves sums of squares from the "total" lines
`TMS` = *variates*	Saves mean squares from the "total" lines
`VR` = *pointers* or *variates*	Saves variance ratios for the model terms in each analysis of variance
`PRVR` = *pointers* or *variates*	Saves probabilities of the variance ratios
`OUTFILENAME` = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create

Description

The RYPARALLEL procedure fits the same regression model (in "parallel") to several response variates, combining and summarizing the information from all the analyses. The response variates are supplied in pointer, using the Y parameter of RYPARALLEL. The model for the regressions is specified by the TERMS, WEIGHTS, OFFSET, CONSTANT, FACTORIAL and FULL options, which operate exactly as in ordinary regression (see the MODEL, TERMS and FIT directives).

The RESIDUALS and FITTEDVALUES parameters allow you to save the residuals and fitted values from the regressions. These are defined as matrices, with a row for each y-variate, and a column for each unit. The RMETHOD option indicates what sort of residual to form, as in the other Genstat regression commands. By default, standardized residuals are formed, but you can set RMETHOD=simple to form simple residuals instead.

The ESTIMATES, SE, TESTESTIMATES and PRESTIMATES parameters save the estimates, standard errors, t-statistics and t-probabilities for the parameters in the regression model. These are defined as matrices, with a row for each y-variate, and a column for each parameter.

The DF, SS, MS, RDF, RSS, RMS, TDF, TSS, TMS, VR and PRVR parameters store information from the analysis of variance table. (DF, SS, MS, VR and PRVR are from the "regression" line, RDF, RSS and RMS are from the "residual" line, and TDF, TSS and TMS are from the "total" line.) With the default setting no of the POOL option each of these is a pointer containing a variate for each term in the TERMS formula. The variates each have a unit for every y-variate. Alternatively, if you set POOL=yes, the parameters each have a single variate, with the values pooled over the terms.

Printed output is controlled by the PRINT option, with settings:

model	for a description of the regression model, and
summary	for a summary of the significance levels found over the analyses for each parameter in the model.

The SPREADSHEET option allows you to save the various output components in spreadsheets. You can save these in either a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx), by setting the OUTFILENAME option to the name of the file to create. If the name is specified without a suffix, '.gwb' is added (so that a Genstat workbook is saved). If OUTFILENAME is not specified, they are put into a spreadsheet opened inside Genstat.

Options: PRINT, TERMS, WEIGHTS, OFFSET, CONSTANT, FACTORIAL, FULL, POOL, RMETHOD, SPREADSHEET.

Parameters: Y, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, TESTIMATES, PRESTIMATES, DF, SS, MS, RDF, RSS, RMS, TDF, TSS, TMS, VR, PRVR, OUTFILENAME.

Method

The analyses are performed by the FIT directive and by matrix calculations.

Action with RESTRICT

Any restrictions on the y-variates will be removed.

See also

Procedures: AYPARALLEL, MAREGRESSION.

Genstat Reference Manual 1 Summary section on: Regression analysis.

R0INFLATED

Fits zero-inflated regression models to count data with excess zeros (D.A. Murray).

Options

PRINT = *string token*	Controls printed output (model, summary, estimates, fittedvalues, monitoring); default mode, summ, esti
DISTRIBUTION = *string token*	Distribution of response variable (poisson, binomial, negativebinomial); default pois
METHOD = *string token*	Method used for model fitting (em, conditional); default em
CONSTANT = *string token*	How to treat constant for count state (estimate, omit); default esti
ZCONSTANT = *string token*	How to treat constant for zero-inflation state (estimate, omit); default esti
XTERMS = *formula*	List of explanatory variates and factors, or model formula for count state of model
ZTERMS = *formula*	List of explanatory variates and factors, or model formula for zero-inflation state of model
WEIGHTS = *variate*	Variate of weights for weighted zero-inflated regression (EM model only)
OFFSET = *variate*	Offset variate to be used in the model (EM model only)
XGROUPS = *factor*	Absorbing factor defining the groups for within-groups regression for the count state model (EM model only)
ZGROUPS = *factor*	Absorbing factor defining the groups for within-groups regression for the zero-inflation state model (EM model only)
MAXCYCLE = *scalar*	Maximum number of iterations for EM algorithm; default 100
TOLERANCE = *scalar* or *variate*	Convergence criteria for EM algorithm, k and in the generalized linear models; default !(1.E-4, 1.E-4, 1.E-4)
ZPARAMETERIZATION = *string token*	Parameterization of the probability of the zero-inflation model (zero, nonzero): if unset, zero is used for the EM model and nonzero for the conditional model

Parameters

Y = *variates*	Response variate
NBINOMIAL = *scalars* or *variates*	Total numbers for DISTRIBUTION=binomial
RESIDUALS = *variates*	Saves the simple residuals
FITTEDVALUES = *variates*	Saves the fitted values
ESTIMATES = *variates*	Saves the estimates of the parameters
SE = *variates*	Saves the standard errors of the estimates
RSAVE = *identifiers*	Saves the regression structure for the final generalized model fitted for the count model
ZSAVE = *identifiers*	Saves the regression structure for the final binomial regression fitted for the zero-inflation model

Description

R0INFLATED can be used to fit zero-inflated regression models to count data with excess zeros. The procedure allows the data to be modelled using two different approaches. The first possibility is to fit a *zero-inflated Poisson regression model* (ZIP), a *zero-inflated binomial regression model* (ZIB) or a *zero-inflated negative binomial regression model* (ZINB) using an EM algorithm (Lambert 1992). In this analysis, the response variable of counts is assumed to be distributed as a mixture of a distribution (such as Poisson) and a degenerate distribution at zero. In these models, a generalized linear model with a Poisson or negative binomial distribution and log link, or with a binomial distribution and logit link, is used for the count model. A generalized linear model with a binomial distribution and logit link is used for the zero-inflation model.

The alternative is to fit the *conditional model* of Welsh *et al.* (1996), which assumes that the data are in one of two states: a state where zeros are observed, or a state where counts are recorded. A binomial model with a logit link is used for the zero state. A truncated Poisson, truncated binomial or truncated negative binomial model is used for the count state.

The response variable is supplied, in a variate, using the Y parameter. The NBINOMIAL parameter must also be set when DISTRIBUTION=binomial, to give the number of binomial trials for each unit. The XTERMS and ZTERMS options each specifies a formula, to describe the count model and the zero-inflation model respectively. The CONSTANT and ZCONSTANT options control whether a constant parameter is included in the count and zero-inflation models.

The METHOD option specifies the type of model to fit: the em setting fits the ZIP, ZIB and ZINB mixture models, and the conditional setting fits the conditional model. The DISTRIBUTION option specifies the distribution for the count model. Note that a log link is always used for the count model with the Poisson and negative binomial distributions, and a logit link is used with the binomial distribution.

The XGROUPS and ZGROUPS options can specify factors whose effects you want to eliminate from the count or zero-inflation state respectively, before any regression is fitted. This method of elimination is sometimes called absorption. (See the GROUPS option of the MODEL directive.) It gives less information than you would get if you included the factor explicitly in the model. For example, no standard errors are produced. However, it saves space and time when data from many different groups are to be modelled. These options are only available for the EM model.

The ESTIMATES and SE parameters save the parameter estimates and their standard errors. R0INFLATED puts them into variates, using the same order as in the display produced by the PRINT option. The simple residuals and the fitted values can be saved using the RESIDUALS and FITTEDVALUES parameters.

The RSAVE and ZSAVE parameters allow you to specify identifiers for the regression save structures for the count and zero-inflation states of the model. These structures store the final state of the regression models fitted. Note that the standard errors for the parameter estimates in the regression save structures will not be correct and should instead be obtained using the SE parameter or by the R0KEEP procedure.

For the mixture models, the WEIGHTS option can specify a variate holding weights for each unit, and the OFFSET option allows you to include an offset (i.e. a variable in the regression model with a regression coefficient fixed at one).

The PRINT option controls printed output, with settings:

model	gives a description of the model, including response and explanatory variates for count and zero-inflation models;
summary	displays minus twice log-likelihood, the Akaike information coefficient (AIC) and the Schwarz (Bayesian) information coefficient (BIC or SIC);
estimates	gives the estimates of the parameters in the model with standard errors based on the asymptotic variance-covariance matrix derived from the inverse of the observed

Fisher information matrix;

`fittedvalues` displays a table of unit labels, values of response variate, fitted values and residuals;

`monitoring` displays monitoring information of the iterative algorithm.

The iterative process for the EM algorithm is controlled by the `MAXCYCLE` option which defines the maximum number of cycles, and the `TOLERANCE` option which sets convergence criteria. The EM algorithm cycle stops when successive values of the log-likelihood are within a tolerance set by the first element of the `TOLERANCE` option. The second and third elements of `TOLERANCE` control the convergence criterion for the aggregation parameter (k) for the negative binomial model and for the generalized linear model, respectively.

The `ZPARAMETERIZATION` option controls how the probability for the zero-inflation model is specified. Note that the parameters in the model specification for the mixture and conditional models have different interpretations. In the mixture model the default setting is `zero`, which parameterizes the model such that ω is the probability of the excess zeros. Alternatively, you can set `ZPARAMETERIZATION=nonzero`, to parameterize the model such that ω is the probability that an observation is generated through the distribution. In the conditional model the default setting is `nonzero`, which parameterizes the model such that $\omega = 1 - p(x)$ where $p(x)$ is the probability of detecting at least one observation, given that there is at least one observation. Alternatively, if you set `ZPARAMETERIZATION=zero`, the parameterization is that $\omega = p(x)$. For further details, see the *Method* section.

Options: `PRINT`, `DISTRIBUTION`, `METHOD`, `CONSTANT`, `ZCONSTANT`, `XTERMS`, `ZTERMS`, `WEIGHTS`, `OFFSET`, `XGROUPS`, `ZGROUPS`, `MAXCYCLE`, `TOLERANCE`, `ZPARAMETERIZATION`.
Parameters: `Y`, ,`NBINOMIAL`, `RESIDUALS`, `FITTEDVALUES`, `ESTIMATES`, `SE`, `RSAVE`, `ZSAVE`.

Method

The zero-inflated Poisson (mixture) regression model has the distribution

$$\begin{aligned} \Pr(Y=y) &= \omega + (1-\omega) \times \exp(-\lambda) \quad \text{for } y=0 \\ &= (1-\omega) \times \exp(-\lambda) \times \lambda^y / y! \quad \text{for } y>0 \end{aligned}$$

where λ and ω are given by the following models

$$\log(\lambda) = \mathbf{X}\boldsymbol{\beta}$$
$$\log(\omega/(1-\omega)) = \mathbf{Z}\boldsymbol{\alpha}$$

where $\mathbf{X}$ and $\mathbf{Z}$ are covariate matrices and $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ are vectors of unknown parameters.

The zero-inflated binomial (mixture) regression model has the distribution

$$\begin{aligned} \Pr(Y=y) &= \omega + (1-\omega) \times (1\text{-p})^n \quad \text{for } y=0 \\ &= (1-\omega) \times p^y \times (1-p)^{n-y} \times n! / (y! \times (n-y!)) \quad \text{for } y>0 \end{aligned}$$

where p and ω are given by the following models

$$\log(p/(1-p)) = \mathbf{X}\boldsymbol{\beta}$$
$$\log(\omega/(1-\omega)) = \mathbf{Z}\boldsymbol{\alpha}$$

The zero-inflated negative binomial (mixture) regression model has the distribution

$$\begin{aligned} \Pr(Y=y) &= \omega + (1-\omega) \times (1+\lambda \times k)^{-(1/k)} \quad \text{for } y=0 \\ &= (1-\omega) \times \Gamma(y+1/k) / (y! \times \Gamma(1/k)) \\ &\quad \times (1+\lambda \times k)^{-(y+1/k)} \quad \text{for } y>0 \end{aligned}$$

where λ and ω are given by the same models as for the Poisson distribution, and k is the extra-variation parameter in the negative binomial distribution.

The maximum likelihood estimates for $\boldsymbol{\beta}$, $\boldsymbol{\alpha}$ and k are obtained using an EM algorithm (Lambert 1992). The standard errors for the parameter estimates are derived using the incomplete data observed information matrix as proposed by Lambert (1992). The default parameterization for the mixture models estimates ω, the probability of excess zeros. You can use the `ZPARAMETERIZATION` option to change the parameterization to estimate ω', the probability that an observation is generated through the distribution instead ($\omega' = 1-\omega$).

In the Poisson case of the conditional model, y has a truncated Poisson distribution (λ). So the probability model is

$$\Pr(Y=y) = \omega \text{ for } y=0$$
$$= (1 - \omega) \times \exp(-\lambda) \times \lambda^y) / \{ y! \times (1 - \exp(-\lambda) \} \text{ for } y>0$$

where λ and ω are given by the following models

$$\log(\lambda) = \mathbf{X}\,\boldsymbol{\beta}$$
$$\log(\omega/(1-\omega)) = \mathbf{Z}\,\boldsymbol{\alpha}$$

In the truncated binomial case, y has a truncated binomial distribution. So the probability model is

$$\Pr(Y=y) = \omega \text{ for } y=0$$
$$= (1 - \omega) \times p^y \times (1 - p)^{n-y} / (1 - (1 - p)^n) \times n! / (y! \times (n-y!)) \text{ for } y>0$$

where p and ω are given by the following models

$$\log(p/(1-p)) = \mathbf{X}\,\boldsymbol{\beta}$$
$$\log(\omega/(1-\omega)) = \mathbf{Z}\,\boldsymbol{\alpha}$$

In the negative binomial case, y has a truncated negative binomial (λ, k). So the probability model is

$$\Pr(Y=y) = \omega \text{ for } y=0$$
$$= (1 - \omega) \times \Gamma(y + 1/k) / (y! \times \Gamma(1/k)) \times (1 + k \times \lambda)^{-(y + 1/k)} \times (1 - (1 + k \times \lambda)^{-1/k})^{-1}, \text{ for } y>0$$

where λ and ω are given by the same models as for the Poisson distribution, and k is the extra-variation parameter in the negative binomial distribution.

The truncated Poisson model is fitted using an iteratively re-weighted least squares algorithm (see Welsh *et al.* 1996). The truncated binomial and negative binomial models are fitted using `FITNONLINEAR`.. The default parameterization for the mixture models estimates ω' (=1−ω), the probability of detecting at least one observation given that there is at least one observation, as in Welsh *et al*. (1996). You can use the `ZPARAMETERIZATION` option to change the parameterization to estimate ω, the probability of detecting a zero observation, instead.

Action with `RESTRICT`

If a parameter is restricted the statistics will be calculated using only those units included in the restriction.

References

Hall, D,B. (2000). Zero-inflated Poisson and Binomial regression with random effects: a case study. *Biometrics*, **56**, 1030-1039.

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics*, **34**, 1-14.

Ridout, M., Demetrio, C.G.B. & Hinde, J. (1998). Models for count data with many zeros. *International Biometrics Conference, Cape Town*.

Welsh, A.H., Cunningham, R.B., Donnelly, C.F. & Lindenmayer, D.B. (1996). Modelling the abundance of rare species: statistical models for counts with extra zeros. *Ecological Modelling*, **88**, 297-308.

See also

Procedures: `RNEGBINOMIAL`, `R0KEEP`.

Genstat Reference Manual 1 Summary section on: Regression analysis.

R0KEEP

Saves information from a zero-inflated regression model for count data with excess zeros fitted by R0INFLATED (D.A. Murray).

Options

RESIDUALS = *variate*	Saves the simple residuals
FITTEDVALUES = *variate*	Saves the fitted values
ESTIMATE = *variate*	Saves the parameter estimates
SE = *variate*	Saves the standard errors of the parameter estimates
VCOVARIANCE = *symmetric matrix*	Saves the variance-covariance matrix of estimates for the ZIP, ZIB and ZINB models
XFITTEDVALUES = *variate*	Saves the fitted values for the count model
XSEFITTEDVALUES = *variate*	Saves the standard errors of the fitted values for the fitted values of the count model
ZFITTEDVALUES = *variate*	Saves the fitted values for the zero model
ZSEFITTEDVALUES = *variate*	Saves the standard errors of the fitted values for the fitted values of the zero model
_2LOGLIKELIHOOD = *scalar*	Saves −2 times the log-likelihood
AIC = *scalar*	Saves the Akaike information coefficient
SIC = *scalar*	Saves the Schwarz (Bayesian) information coefficient

No parameters

Description

This procedure allows you to copy information into Genstat data structures from a model that has been fitted to count data with excess zeros by procedure R0INFLATED. You do not need to declare the structures in advance; Genstat will declare them automatically to be of the correct type and length.

The RESIDUALS and FITTEDVALUES options save the simple residuals and the fitted values. The ESTIMATES and SE options save the parameter estimates and their standard errors. The VCOVARIANCE option saves the variance-covariance matrix of estimates from either a ZIP or ZINB model. The ZFITTEDVALUES and ZSEFITTEDVALUES options save the fitted values and standard errors of fitted values for the zero state. Similarly, the XFITTEDVALUES and XSEFITTEDVALUES options save the fitted values and standard errors of fitted values for the count state. The _2LOGLIKELIHOOD option saves −2 times the log-likelihood, and the AIC and SIC options save the Akaike and Schwarz (Bayesian) information coefficients respectively.

Options: RESIDUALS, FITTEDVALUES, ESTIMATES, SE, VCOVARIANCE, XFITTEDVALUES, XSEFITTEDVALUES, ZFITTEDVALUES, ZSEFITTEDVALUES, _2LOGLIKELIHOOD, AIC, SIC.
Parameters: none.

See also

Procedure: R0INFLATED.
Genstat Reference Manual 1 Summary section on: Regression analysis.

R2LINES

Fits two-straight-line (broken-stick) models to data (A.W.A. Murray & J.T. Wood).

Options

PRINT = *string token*	What to print (model, summary, estimates, fittedvalues, intercepts); default mode, summ, esti
PLOT = *string tokens*	What to plot (breakpoint, lines, residuals); default * i.e. nothing
HORIZONTAL = *string token*	Forces either the left- the or right-hand line to be horizontal (left, right); default * i.e. neither
CIPROBABILITY = *scalar*	Sets the probability level of the confidence interval about the X value at the intersection; default 0.95
†NGRIDLINES = *scalar*	Controls the number of points used in the initial search for the intersection of the lines; default 100
TERMS = *variates*	Additional x-variates to include in the model; default none
†METHOD = *string token*	Optimization method (gaussnewton, newtonraphson, fletcherpowell); default newt

Parameters

Y = *variates*	Response variates to be modelled
X = *variates*	Explanatory variable for each response variate
TITLE = texts	Title to use on the graphs for each response variate
FITTEDVALUES = *variates*	Saves fitted values
RESIDUALS = *variates*	Saves standardized residuals
ESTIMATES = *variates*	Saves estimates from each model (i.e. intersection coordinates and slopes of the fitted lines)
SE = *variates*	Saves standard errors of the estimates
INTERCEPTS = *variates*	Saves the intercepts
LOWER = *scalars*	Saves the lower bound of the confidence interval about the x-value at the intersection
UPPER = *scalars*	Saves the upper bound of the confidence interval about the x-value at the intersection
PARTIALLIKELIHOOD = *pointers*	Saves the partial likelihood and grid values for partial likelihood plots

Description

R2LINES fits a model consisting of two straight line segments (a broken-stick or split-line model) to the data. The HORIZONTAL option can be set to left or right to force either the left- or the right-hand line to be horizontal. A check is made to ensure that the overall best intersection point is used for the two lines. The NGRIDLINES option specifies the number of extra points used between each pair of x's in the initial search for the best intersection point; default 100. The METHOD option specifies the optimization method that is then used to estimate the intersection point. The default is to use the Newton-Raphson method. (See the RCYCLE directive for details.)

The response variate is specified by the Y parameter, and the explanatory variate by the X parameter. You can also use the TERMS option to include additional x-variates in the model.

Information can be saved from the analysis by using the FITTEDVALUES, RESIDUALS, ESTIMATES and SE parameters, in the usual way. The LOWER and UPPER parameters can save the lower and upper values of a confidence interval for the x location of the intersection (or

breakpoint) of the lines. The INTERCEPTS parameter can save a variate containing the intercept with the y-axis and of the two lines with the x-axis. The probability for the interval is specified by the CIPROBABILITY option, with default 0.95 (i.e. 95%).

Printed output is controlled by the PRINT option. The settings model, summary and fittedvalues operate as in ordinary regression. The estimates setting produces the parameter estimates as usual, and also the confidence interval for the x-value of the intersection of the lines. There is also a setting intercepts, which prints the values at which the model intercepts the x-axis and y-axis.

The PLOT option has settings to produce the following plots:

breakpoint	displays a partial likelihood plot, displaying the approximate F ratio for the model for a range of positions of the breakpoint between the two lines;
lines	plots the fitted lines;
residuals	produces the four standard model-checking plots of residuals – histograms, Normal and half-Normal plots, and plots of residuals against fitted values.

The TITLE parameter can supply a title for the plots; the default is to use the identifier of the Y variate. The PARTIALLIKELIHOOD parameter can save the points used for the breakpoint plot, as a pointer storing a variate with the y-coordinates as its first element, and a variate with the x-coordinates as its second element.

Options: PRINT, PLOT, HORIZONTAL, CIPROBABILITY, NGRIDLINES, TERMS, METHOD.
Parameters: Y, X, TITLE, FITTEDVALUES, RESIDUALS, ESTIMATES, SE, INTERCEPTS, LOWER, UPPER, PARTIALLIKELIHOOD.

Method

A model consisting of two straight line segments is fitted by least squares. This is done by defining variables,

```
Slope_1 = (X - Breakpoint_X) * (X < Breakpoint_X)
Slope_2 = (X - Breakpoint_X) * (X > Breakpoint_X)
```

where X is the explanatory variable, and Breakpoint_X is the value of the explanatory variable where the two segments join. The response variable is then regressed on Slope_1 and Slope_2. The slopes of the lines are the regression coefficients for Slope_1 and Slope_2. If Breakpoint_X is known, there is no problem. However, if it is not known, care is needed because the residual mean square may have local minima. If one of the straight lines is assumed to be horizontal, then only one slope is fitted and the other is set to zero.

The values of X are sorted into increasing order, and a sequence of trial values for Breakpoint_X is formed, consisting of the original values X plus NGRIDLINES-1 equally spaced values between each consecutive pair of X's. The regression of Y on Slope_1 and Slope_2 is fitted for each of these trial values. The one giving the smallest residual sum of squares is then chosen as a starting value for Breakpoint_X, and the model is fitted as a nonlinear model using FITNONLINEAR.

Suppose that at the true value of Breakpoint_X the residual sum of squares is Rt, and that at the fitted value of Breakpoint_X the residual sum of squares is Rf and the residual mean square is Sf. If we assume that the observations are independently and normally distributed with common variance, the distribution of (Rt-Rf)/Sf can be approximated by an F-distribution with degrees of freedom one and number of observations minus four. Hence the set of values for Breakpoint_X for which (Rt-Rf)/Sf is less than the 95th percentile of the F-distribution defines a 95% confidence region. It is possible for this region to consist of more than one distinct interval. The confidence interval will contain the minimum and maximum values of Breakpoint_X in the region. The calculated variance ratios and the trial values of

`Breakpoint_X` are returned in `PARTIALLIKELIHOOD`.

Action with `RESTRICT`
Restrictions on `X` and `Y` are obeyed.

See also
`FITCURVE`, `FITNONLINEAR`.
Genstat Reference Manual 1 Summary section on: Regression analysis.

SAGRAPES

Produces statistics and graphs for checking sensory panel performance (D.I. Hedderley).

Options

PRINT = *string tokens*	Controls printed output (aovtables, graphs, summarystatistics, tables); default grap, tabl
TREATMENTS = *factor*	Factor defining the different treatments that are being assessed
SESSIONS = *factor*	Factor defining the sessions on which the assessments were done
ASSESSORS = *factor*	Factor defining the individual assessors
SCALING = *string token*	Equal scaling for x and y axes on Drift-Unreliability and Discrimination-Disagreement graphs (equal, none); default none
DESCRIPTION = *text*	Extra information to print on graphs

Parameter

DATA = *variates*	Variate for each attribute, containing the recorded score

Description

A trained panel of sensory assessors may test a set of products (e.g. taste a set of food samples) at several sessions, each time rating them on a range of attributes. If you have several measurements of the same samples from the same individuals, you can investigate how consistent and discriminating the individual assessors are. The scores recorded for the attributes are specified, in a list of variates, by the DATA parameter. The TREATMENTS, SESSIONS and ASSESSORS options supply factors defining the treatment, session and assessor involved with each unit of the DATA variates.

SAGRAPES presents six statistics based on analyses of variance, proposed by Schlich (1994), to describe how well individual assessors use individual attributes. These are:

Location	the assessors' overall mean score on that attribute;
Span	the mean standard deviation of the assessors' scores within a session;
Unreliability	the ratio of the root mean square residual (from a model fitting TREATMENTS and SESSIONS main effects to each assessor) to Span, i.e. what proportion of the spread in an assessor's ratings is due to changes in the relative scoring of samples in different sessions;
Drift-mood	the ratio of the root mean square for sessions (from a model fitting TREATMENTS and SESSIONS main effects to each assessor) to span, i.e. how much an assessor's average score changes from session to session, compared to the spread of scores within a session;
Discrimination	the variance ratio for TREATMENTS from a model fitting TREATMENTS and SESSIONS main effects to each assessor;
Disagreement	an estimate of how much each assessor contributes to the variance ratio of the ASSESSOR.TREATMENTS interaction (from a model fitting ASSESSORS/SESSIONS + TREATMENTS/ASSESSORS to the whole panel).

The PRINT option controls the output, with the following settings.

tables	prints a table of these statistics for each assessor for each

of the attributes in DATA.

graphs	produces a composite plot of three graphs (Location against Span, Unreliability against Drift-mood, and Discrimination against Disagreement) for each attribute. The points on the plots are labelled with the labels from the ASSESSORS factor. On the plot of Discrimination against Disagreement, a star is plotted at the 5% critical values of the relevant F distributions; so ASSESSORS to the right of the star are significantly discriminating between TREATMENTS, and ASSESSORS above the star contribute significantly to the ASSESSORS.TREATMENTS interaction.
aovtables	prints the panel ANOVA tables for each attribute.
summarystatistics	prints overall summary statistics (numbers of observations, means and standard deviations) for each attribute, across the whole panel and all samples.

Unreliability and Drift-mood are measured on the same scale (multiples of Span), as are Discrimination and Disagreement (F-ratios). Setting option SCALING=equal scales the x and y axes of the Unreliability against Drift-mood and Discrimination against Disagreement graphs equally.

The DESCRIPTION option can be used to provide additional information (for instance, the name of the study) to label the graphs.

Options: PRINT, TREATMENTS, SESSIONS, ASSESSORS, SCALING, DESCRIPTION.
Parameter: DATA.

Method

Schlich (1994) proposed the procedure, and implemented it in SAS. This procedure uses the calculations given in the article to produce graphs for individual attributes. Currently it does not produce the graphs comparing different attributes which Schlich suggests.

Action with RESTRICT

Any of the DATA variates, or the TREATMENTS, SESSIONS or ASSESSORS factors, can be restricted to analyse a subset of the data units.

Reference

Schlich, P. (1994). GRAPES: A method and a SAS program for graphical representations of assessor performances. *Journal of Sensory Studies*, **9**, 157-169.

See also

Procedure: GENPROCRUSTES.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

SAMPLE

Samples from a set of units, possibly stratified by factors (P.W. Lane).

Options

SEED = *scalar*	Seed for the random number generator; default 0 i.e. continue from previous generation
NVALUES = *scalar*	Number of units from which a simple sample is to be taken; default * i.e. as defined by UNITS statement

Parameters

NSAMPLE = *scalars* or *tables*	Number of values in simple sample, or table of numbers of values at each combination of levels of its classifying factors; no default
SAMPLE = *identifiers*	Structure to store the result; no default

Description

Procedure SAMPLE produces a random sample from a set of units. A simple sample can be obtained by setting the NSAMPLE parameter to the required number in the sample, and the NVALUES option to the number of units in the set. The NVALUES option can be omitted if the required number of units has been defined by a UNITS statement earlier in the job.

For a stratified sample, the NSAMPLE option should be set to a table containing the required number of units to be sampled at each combination of levels of the factors classifying the table. The NVALUES option is not then relevant as the set of units is determined by the values of the classifying factors.

The SAMPLE parameter must be set to an identifier, which will be formed into a variate containing a set of NSAMPLE integers in the range (1...NVALUES), obtained by random sampling without replacement. The SEED option can be set to define a starting value for the random numbers used to select the units. This can be omitted if some random numbers have already been generated during the current job; SAMPLE will then take the numbers that continue the previous sequence.

Options: SEED, NVALUES.
Parameters: NSAMPLE, SAMPLE.

Method

For a simple sample, a full set of units (1...NVALUES) is randomly ordered and the first NSAMPLE values are taken. For a stratified sample, the units are sorted according to levels of the classifying factors (after random ordering) and then the requested number of values are taken for each combination of levels.

Action with RESTRICT

The factors classifying the table must not be restricted. The procedure cannot be used on a restricted set of units.

See also

Directive: CALCULATE.
Procedures: GREJECTIONSAMPLE, GRMULTINORMAL, SVSAMPLE.
Functions: GRBETA, GRBINOMIAL, GRCHISQUARE, GRF, GRGAMMA, GRHYPERGEOMETRIC, GRLOGNORMAL, GRNORMAL, GRPOISSON, GRSAMPLE, GRSELECT, GRT, GRUNIFORM.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

SBNTEST

Calculates the sample size for binomial tests (R.W. Payne & D.A. Murray).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
PRMETHOD = *string token*	Method to be used to calculate the probabilities for the binomial test (angular, normalapproximation, exact); default norm
PROBABILITY = *scalar*	Significance level for the test; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Type of test to be done (onesided, twosided); default ones
NULL = *scalar*	Probability under the null hypothesis for the one-sample test; default 0.5
RATIOREPLICATION = *scalar*	Ratio of replication sample2:sample1 (i.e. the size of sample 2 should be RATIOREPLICATION times the size of sample 1); default 1
REPLICATION = *variate*	Replication values for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

P1 = *scalars*	Probability to detect in sample 1
P2 = *scalars*	Probability to detect in sample 2
NREPLICATES = *scalars*	Saves the required number of replicates
VREPLICATION = *variates*	Numbers of replicates for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

SBNTEST calculates the number of replicates (or sample size) required for a binomial test. A one-sample binomial test assesses the evidence that the probability of success within a sample differs from some specific value. The probability that needs to be detected is specified by the P1 parameter, and the value from which it needs to be distinguished (i.e. the value under the null hypothesis) is specified by the NULL option. If NULL is not set, the default is 0.5. Alternatively, a two-sample test assess the evidence that probabilities within two samples are different. The anticipated probability within the first sample is then specified by the P1 parameter, and the probability within the second sample (from which it must be distinguished) is specified by the P2 parameter.

The PRMETHOD option defines the type of binomial test that is to be done. The normalapproximation setting relates to a test based on the Normal approximation to the binomial distribution (see the BNTEST procedure), while the angular setting is for a test using an angular transformation of the probabilities. The final setting, exact, is available only for the one-sample test and assumes an exact test using the binomial distribution.

The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%). The required probability for detection of the difference between the probabilities (that is, the *power* of the test) is specified by the POWER option (default 0.9). It is generally assumed that the sizes of the samples in the two-sample test should be equal. However, you can set the

RATIOREPLICATION option to a scalar, R say, to indicate that the size of the second sample should be R times the size of the first sample. By default, SBNTEST assumes a one-sided test is to be used, but you can set option TMETHOD=twosided to take a two-sided test instead. The NREPLICATES parameter allows you to save the required size of the first sample.

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates in each sample (i.e. the size of each sample);
power	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding powers can also be saved, in variates, using the VREPLICATION and VPOWER parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, PRMETHOD, PROBABILITY, POWER, TMETHOD, NULL, RATIOREPLICATION, REPLICATION.
Parameters: P1, P2, NREPLICATES, VREPLICATION, VPOWER.

Method

When PRMETHOD=normalapproximation, the distribution of the probability in sample i is approximated by a Normal distribution with mean p_i and variance $p_i(1-p_i)/n_i$, where p_i is the binomial probability and n_i is the sample size. With PRMETHOD=angular, the probability is transformed to radians by an angular distribution, and the variance is then $\sqrt{(0.25/n_i)}$. For PRMETHOD=exact, the calculations are done using the CUBINOMIAL and EDBINOMIAL functions (one-sample test only).

See also

Procedure: BNTEST.
Genstat Reference Manual 1 Summary section on: Design of experiments.

SCORRELATION

Calculates the sample size to detect specified correlations (R.W. Payne).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
PROBABILITY = *scalar*	Significance level at which the correlation or difference between correlations is to be tested; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Whether to a one- or two-sided test is to be made (onesided, twosided); default ones
RATIOREPLICATION = *scalar*	Ratio of replication sample2:sample1 (i.e. the size of sample for group 2 should be RATIOREPLICATION times the size of sample for group 1); default 1
REPLICATION = *variate*	Replication values for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

COR1 = *scalars*	Anticipated correlation in group 1
COR2 = *scalars*	Anticipated correlation in group 2
NREPLICATES = *scalars*	Saves the required number of replicates
VREPLICATION = *variates*	Numbers of replicates for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

SCORRELATION may be useful when you wish to assess the correlation between two variables within a single group of subjects, or when you wish to compare the correlations between two groups of subjects. The correlation in this case is the product moment correlation coefficient, as calculated by the CORRELATION function (and so the variables are assumed to have Normal distributions).

If there is a single group of subjects the correlation is specified (in a scalar) by the COR1 parameter, and the assumption is that we wish to assess whether this is non-zero. With two groups the correlations are specified by the COR1 and COR2 parameters (again in scalars). Generally equal sample sizes are assumed for the two groups. However, you can set the RATIOREPLICATION option to a scalar, R say, to indicate that the size of the second sample should be R times the size of the first sample. The NREPLICATES parameter allows you to save the required size of the first sample.

The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%). By default this is for a one-sided test, but you can set option TMETHOD=twosided for a two-sided test. The required probabilty for detection of the correlation or difference in correlations (that is, the *power* of the test) is specified by the POWER option (default 0.9).

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates in each sample (i.e. the size of each sample);
power	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding powers can also be saved, in variates, using the VREPLICATION and VPOWER parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, PROBABILITY, POWER, TMETHOD, RATIOREPLICATION, REPLICATION.
Parameters: COR1, COR2, NREPLICATES, VREPLICATION, VPOWER.

Method

With a single group, suppose that the sample correlation is r and the number of subjects is n. SCORRELATION uses the fact that, under the null hypothesis of a zero correlation, the variable

$$t = r \times \sqrt{((n - 2) / (1 - r^2))}$$

has a t distribution on $n-2$ degrees of freedom.

With two groups, SCORRELATION uses Fisher's Z transformation:

$$z = 0.5 \times \log((1 + r)/(1 - r))$$

Provided the sample sizes are reasonably large, z can be assumed to have a Normal distribution with variance $1/(n-3)$.

See also

Directive: CORRELATE.
Procedure: FCORRELATION.
Genstat Reference Manual 1 Summary section on: Design of experiments.

SDISCRIMINATE

Selects the best set of variates to discriminate between groups (D.B. Baird, L.H. Schmitt & J.W. McNicol).

Options

PRINT = *string tokens*	Printed output from the analysis (summary, steps, validation, specificity, discrimination, monitoring); default summ, vali, spec, disc
PLOT = *string tokens*	What plots to produce (errorrate, steps, specificity, discriminant); default erro, steps, spec, disc
DDISCRIMINANT = *string tokens*	What to display on the discriminant plot (means, mlabels, scores, polygons, confidencecircle); default means, mlabels, scores, conf
METHOD = *string token*	The variable selection method to use (forward, backward); default forw
NSELECT = *scalar*	Number of variates to select; default 4
CRITERION = *string token*	Criterion to use to select variables (wilkslambda, crossvalidation, bootstrap, jackknife); default wilk
MODELCHOICE = *string token*	Which model to save (optimal, nselect); default opti
VALIDATIONMETHOD = *string token*	Validation method to use to calculate error rates (bootstrap, crossvalidation, jackknife, prediction); default cros
NSIMULATIONS = *variate*	Number of bootstraps or cross-validation sets to use for selection and for validation; default !(10,50)
NCROSSVALIDATIONGROUPS = *scalar*	Number of groups for cross-validation, default 10
SEED = *scalar*	Seed for random number generation; default 0
YROOT = *scalars*	Specifies roots for plotting on y-axes
XROOT = *scalars*	Specifies roots for plotting on x-axes

Parameters

DATA = *pointers*	Each pointer contains a set of variates that are available to be selected
GROUPS = *factors*	Define groupings for the units in each training set
FORCED = *pointers*	Variates that must be included in the model
SELECTED = *pointers*	Saves the variates in the final model
STEPS = *pointers*	Saves the criterion values for each step in the model selection
ERRORRATE = *scalars*	Saves the validation error rate for the final model
SPECIFICITY = *matrices*	Saves the specificity table for the final model
ALLOCATION = *factors*	Saves the groups allocated by the final model
LRV = *LRVs*	Saves the LRVs from the final discriminant analysis
SCORES= *matrices* or *pointers*	Saves discriminant scores for units from the final model

Description

SDISCRIMINATE uses forward selection or backwards elimination to search for the best set of variates to discriminate between groups. The variates that are available for the discrimination

must be specified, in a pointer, by the DATA parameter. The membership of the groups must be specified, in a factor, by the GROUPS parameter. If there are some variates that must always be included in the model, these can be specified, in a pointer, by the FORCED parameter.

Printed output is controlled by the option PRINT, with settings:

summary	summary of the model fitting,
steps	criterion values evaluated at each step of the model fitting,
validation	error rates at each model step,
specificity	specificity of allocation (i.e. the proportion of each group that is assigned correctly),
discrimination	the standard discriminant analysis output for the final model, and
monitoring	criterion values for each model tried.

The default is PRINT=summ,vali,spec,disc.

The PLOT option controls what plots are displayed, with settings:

errorrate	error rate at each selection step,
steps	criterion values at each step of the model fitting,
specificity	specificity at each selection step, and
discriminant	the standard discriminant plot from the final model.

By default these are all plotted. The DDISCRIMINANT option allows group means, labels for group means, unit scores, group polygons enclosing units, and 95% confidence circles around group means to be included on the discriminant plot. The YROOT and XROOT options specify the roots for the axes.

The selection method is defined by the METHOD option. The forward setting starts with the FORCED model and then, at each step, looks to see which of DATA variates not already in the model gives the best improvement; this is the default. The backward setting starts with the model, and looks to see which variate in model (other than those in FORCED) gives the least reduction in the criterion when eliminated at that step.

The criterion for evaluating the model is defined by the CRITERION option, with settings:

wilkslambda	uses the ratio of the determinant of the within-group sums of squares and products to the determinants of the total sums of squares and products (default),
crossvalidation	uses the cross-validation error rate,
bootstrap	uses the bootstrap error rate, and
jackknife	uses jackknifing.

Cross validation, bootstrapping and jackknifing take much longer than the use of Wilks' lambda.

The number of variates in the final model (excluding those in the FORCED model) is set by NSELECT option. The MODELCHOICE option indicates how to choose the final model. The default setting optimal takes the model from the step with the minimum validation error. Alternatively, the nselect setting takes the model with the number of variates specified by the NSELECT option.

The VALIDATIONMETHOD option specifies the validation method, with settings for prediction, cross-validation, jackknife and bootstrap. Cross-validation works by randomly splitting the units into a number of groups specified by the NCROSSVALIDATIONGROUPS option (default 10). It then omits each of the groups, in turn, and predicts how the the omitted units are allocated to the discrimination groups. Jackknifing leaves the units out one at a time, and uses the rest of the data to predict the group of the omitted unit. The bootstrap method works by drawing a bootstrap sample of units (a random sample of units with replacement of the same size as the original sample), and predicting the units that are not present in the random sample. The resulting bootstrap error rate is then calculated as a weighted average of the error rate of the omitted observations and the predictive error rate of the bootstrap sample. The weights used are 0.632 and 0.368 respectively, and so this is known as the *632 rule*.

The NSIMULATIONS option sets the number of simulations for cross-validation or bootstrapping. It should be set to a variate with two values: the first value defines the number of simulations to use during selection (default 10), and the second sets the number to use in the estimation of the error rates (default 50).

The SEED option provides the seed for the random numbers used for the randomizations during in the simulations. The default value of 0 continues an existing sequence of random numbers, if none have been used in the current Genstat job, it initializes the seed automatically using the computer clock.

The SELECTED parameter can save the contents of the chosen model, in a pointer. The STEPS parameter can save a pointer with a variate for each step of the selection, containing the criterion evaluated for each DATA variate at then step. The variates contain a missing value if the DATA variate had already been included or excluded from the model. The ERRORRATE parameter can save a variate with the minimum value of the validation error rate after each step. The SPECIFICITY parameter can save a matrix containing the specificity table for the final model. The LRV parameter can save the latent roots, vectors and trace from the final discriminant analysis, and the ALLOCATION and SCORES parameters can save the assigned groups and discriminant scores.

Options: PRINT, PLOT, DDISCRIMINANT, METHOD, NSELECT, CRITERION, MODELCHOICE, VALIDATIONMETHOD, NSIMULATIONS, NCROSSVALIDATIONGROUPS, SEED, YROOT, XROOT.
Parameters: DATA, GROUPS, FORCED, SELECTED, STEPS, ERRORRATE, SPECIFICITY, ALLOCATION, LRV, SCORES.

Method

The procedure steps through the models using FSSPM to calculate Wilks' Lambda, and subsidiary procedures _SDISCROSSVALIDATE and _SDISBOOTSTRAP to calculate the other selection criteria. DISCRIMINATE is called to provide the output for the final model.

Action with RESTRICT

The input variates and factor may be restricted (but any restrictions must be identical). The restricted units are omitted from the analysis.

See also

Directive: CVA.
Procedures: CVAPLOT, DBIPLOT, DISCRIMINATE, QDISCRIMINATE.
Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

SEDLSI

Calculates least significant intervals (M.C. Hannah).

Options

PRINT = *string tokens*	What to print (delta, lsi, fittedsed, discrepancy, maxdiscrepancy, %discrepancy); default delta, lsi, maxd
METHOD = *string token*	Selects the method for computing the deltas (leastsquares, max, maxpse); default leas
PLOT = *string tokens*	What to plot (sed, lsi); default sed, lsi
CHECKFIT = *string token*	Which pairwise contrasts to use in printed output or plots involving the fitted SEDs (specified, all); default spec
PROBABILITY = *scalar*	Significance level for the least significent intervals; default 0.05.
DF = *scalar*	Degrees of freedom for the t-distribution use in calculation of the least significent intervals; default * assumes an infinite number of degrees of freedom (i.e. a Normal rather than a t-distribution)
WINDOW = *scalar*	Window in which to plot the graphs
TITLE = *text*	Title for the graphs; default 'Estimates with LSIs by Treatment'
YTITLE = *text*	Title for the y-axis; default 'Estimates'

Parameters

ESTIMATES = *tables* or *variates*	Parameter estimates; if these are not supplied SEDLSI can calculate the parameters $\{\delta_i\}$ but not the LSIs
SED = *symmetric matrices*	Matrix containing standard errors of (pairwise) differences between estimates
VCOVARIANCE = *symmetric matrices*	Matrix containing variances and covariances of estimates
WEIGHTS = *symmetric matrices*	Weight (or importance) to be used for each pairwise difference; default is a matrix of ones (i.e. all pairwise differences of equal interest)
LABELS = *texts*	Text vector (e.g. treatment labels) for labelling output; default takes the labels of levels of the factor classifying an ESTIMATES table or (if ESTIMATES is a variate or unset) row labels from SED or VCOVARIANCE
DELTA = *variates*	Saves the estimated parameters $\{\delta_i\}$
LSI = *pointers*	Saves details of the least significant intervals
FITTEDSED = *symmetric matrices*	Saves the fitted SED matrices

Description

Least significant intervals (LSIs) are used for comparing a set of estimates (e.g. predicted means from ANOVA or regression) graphically, especially when their SEDs differ. LSIs are intervals (or error bars) that are designed to overlap where there is no significant difference between estimates, and to be disjoint (i.e. not to overlap) where there are significant differences.

Presentation of results can be problematic when standard errors of differences vary appreciably due to unequal replication or an unbalanced design. LSIs attempt to address this difficulty, and are suitable for graphical presentation (Snee 1981). They can also be useful for

presentation of results following a transformation of scale. Intervals can be formed on the scale on which the analysis of data took place (or the scale of a linear predictor for a generalized linear model) and be back-transformed, along with point estimates, to the original measurement scale for graphical presentation (see e.g. Hannah & Quigley 1996).

The SEDs can be supplied, in a symmetric matrix, using the SED parameter. Alternatively, you can provide a (symmetric) variance-covariance matrix, using the VCOVARIANCE parameter. SEDLSI uses these to compute parameters $\{\delta_i\}$ such that $\delta_i + \delta_j$ is approximately equal to SED_{ij}. The delta values can be saved, in a variate, using the DELTA parameter.

You can also supply parameter estimates (e.g. treatment means), in either a variate or a table, using the ESTIMATES parameter. If ESTIMATES is a variate, you may want to use the LABELS option to supply a text of labels. If estimates are available, SEDLSI can also construct least significant intervals (LSIs)

```
lower_LSI = ESTIMATES - EDT(1 - PROBABILITY/2; DF) * DELTA
upper_LSI = ESTIMATES + EDT(1 - PROBABILITY/2; DF) * DELTA
```

where the significance probability is specified by the PROBABILITY option (default 0.05), and the degrees of freedon are specified by the DF option. If DF is not set, the number of degrees of freedom is assumed to be infinite (and so SEDLSI uses a Normal rather than a t-distribution).

When the SEDs are all equal the calculation is trivial; $\delta = SED/2$ (Snee 1981). When SEDs depend on the treatment pair, estimation of δ is more difficult and there may not be an exact solution. However, there is usually an adequately approximate or a conservative solution. SEDLSI offers three methods for estimating delta, requested using the METHOD option. The first method (leastsquares, the default) provides least-squares estimates such that $\delta_i + \delta_j$ is approximately equal to SED_{ij}. The second method (max) provides estimates such that $\delta_i + \delta_j$ is greater than or equal to SED_{ij}. For METHOD settings leastsquares and max at least one of the SED or the VCOVARIANCE parameters must be set. The third method (maxpse) is similar to the max method but the δ's are constrained to be proportional to the standard errors of the estimates, SQRT(DIAG(VCOVARIANCE)). For this method, the VCOVARIANCE parameter must be set. This method may be considered desirable as it apparently constrains the width of resulting LSIs to reflect the relative precisions of the estimates more faithfully. However, it is often highly conservative, with some $\delta_i + \delta_j$ values much greater than SED_{ij}, and it often neglects an exact solution.

Usually only comparisons between certain pairs of means are of genuine interest. To restrict attention just to these pairwise differences, a symmetric matrix corresponding to the SED or VCOVARIANCE matrix can be supplied using the WEIGHTS parameter. This should contain zero in the positions of the contrasts that are not of interest, and one elsewhere. This then weights-out irrelevant SEDs from the calculation and thus avoids the δ's being unnecessarily large (conservative) for the purpose at hand. For example, it could be that the only contrasts of interest are those between each treatment and a control treatment. This is specifed by a weights matrix with the row and column corresponding to the control containing ones, and with zeros elsewhere. By default all the weights are one (signifying all pairwise comparisons of interest). For the leastsquares or max methods, the weights can be any non-negative numeric values to reflect the (subjective) importance of particular pairwise contrasts.

Printed output is controlled by the PRINT option, with settings:

delta	prints the parameters $\{\delta_i\}$,
lsi	prints the least significant intervals,
fittedsed	prints the matrix $[\delta_i + \delta_j]$ of fitted SEDs,
discrepancy	prints the difference between $[\delta_i + \delta_j]$ and $[SED_{ij}]$,
maxdiscrepancy	prints the maximum difference between $[\delta_i + \delta_j]$ and $[SED_{ij}]$,
%discrepancy	prints the difference as a percentage.

The default is PRINT=delta,lsi,maxd.

The PLOT option produces graphs:

lsi	plots the least significant interval for each estimate, and
sed	plots the difference between $[\delta_i + \delta_j]$ and $[SED_{ij}]$.

By default PLOT=lsi,sed. The WINDOW option allows you to specify the window in which to plot the LSIs. By default a window is defined internally, within SEDLSI, to fill the whole screen. The TITLE option supplies the title for the plot (default 'Estimates with LSIs by Treatment'), and the YTITLE option supplies a title for the y-axis (default 'Estimates').

If the δ's do not reproduce the SEDs exactly, it is recommended that the success of the approximation be checked, by examining the fitted SEDs, the differences, or the percent differences. By default, these outputs are produced only for differences of interest (indicated by non-zero weights in the WEIGHTS matrix). If you also wish to check how well the solution applies to contrasts that had weight zero, you can set option CHECKFIT=all to retain all the fitted SED values, provided their corresponding SED_i values were non-missing. (Note, though, that CHECKFIT controls only what contrasts are printed or plotted, not the ones that are used to estimate the deltas.)

The information defining the LSIs can be saved, in a pointer, using the LSI parameter. The components of the pointer are 'Label', 'lowLSI', 'estimate' and 'upLSI'; each is a variate except for 'Label' which is a text. The LSI pointer can be used as input to the LSIPLOT procedure, to plot the LSIs on a later occasion.

Options: PRINT, METHOD, PLOT, CHECKFIT, PROBABILITY, DF, WINDOW, TITLE, YTITLE.
Parameters: ESTIMATES, SED, VCOVARIANCE, WEIGHTS, LABELS, DELTA, LSI, FITTEDSED.

Method

Approximate least significant intervals are calculated as

```
lower_LSI = ESTIMATES + EDT(1 - PROBABILITY/2; DF) * DELTA
upper_LSI = ESTIMATES + EDT(1 - PROBABILITY/2; DF) * DELTA
```

where

```
EDT(1 - PROBABILITY/2; DF)
```

is the

```
1 - PROBABILITY/2
```

quantile of the t-distribution with DF degrees of freedom.

SEDLSI offers three methods of estimating δ. The first method (leastsquares, the default) provides weighted least squares estimates based on the model

$$SED_{ij} = \delta_i + \delta_j$$

with weights optionally provided in the WEIGHTS parameter.

The second method (max) described in Hannah & Quigley (1996) uses

$$\delta_i = \max \{SED_{ij} / \delta_{oi} + \delta_{oj}: j\}\ \delta_{oi}$$

where the parameters $\{\delta_{oi}\}$ are those obtained from the ordinary least-squares method.

The third method (maxpse) is the same as the second method but with the parameters $\{\delta_{oi}\}$ being standard errors of estimates, obtained as square roots of the diagonal of the variance-covariance matrix for the estimates.

The leastsquares method generally gives closer approximations to SEDs, but may be anti-conservative for some comparisons and conservative for others. Maximum SED methods are never anti-conservative but can be excessively conservative. If an exact solution to $\delta_i + \delta_j = SED_{ij}$ exists, the leastsquares and max methods should find it.

If there is no contrast of interest for a particular estimate, due either to missing values in the SED or VCOVARIANCE matrix (zeros are interpreted as missing values here), or zeros specified in the WEIGHTS matrix, the corresponding δ is not estimated. SEDLSI also checks for missing values in the ESTIMATES parameter and sets SED elements corresponding to these as missing.

If the only contrasts of interest are those between each treatment and a control treatment, the number of relevant SEDs is one fewer than the number of delta values requiring estimation. `SEDLSI` detects this treatments-verses-control scenario and, if `METHOD=leastsquares`, it imposes the arbitrary constraint $\delta_{control} = SE_{control}$ if `VCOVARIANCE` is set, or $\delta_{control} = \min(SE_{control,i})$ otherwise.

References

Snee, R.D. (1981). Graphical display and assessment of means. *Biometrics*, **37**, 835-836.

Hannah, M.C. & Quigley, P. (1996). Presentation of ordinal regression analysis on the original scale. *Biometrics*, **52**, 771-775.

Hannah, M.C. (1999). Usefully combining a series of unreplicated cheesemaking experiments. *Journal of Dairy Research*, **66**, 365-374.

See also

Procedures: `LSIPLOT`, `SED2ESE`.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

SED2ESE

Calculates effective standard errors that give good approximate standard errors of differences (R.W. Payne).

Option

PRINT = *string token*	Controls printed output (ese, discrepancy, maxdiscrepancy, %discrepancy, %accounted); default * i.e. none

Parameters

SED = *symmetric matrices*	Standard errors of differences to be approximated
ESE = *variates* or *tables*	Saves the effective standard errors
DISCREPANCY = *symmetric matrices*	Saves the discrepancies between the standard errors of differences and the approximate values calculated from the effective standard errors
%ACCOUNTED = *scalars*	Percentage of variation amongst the standard errors of differences accounted for by the approximate values calculated from the effective standard errors
TEMPLATE = *tables*	Table that can be duplicated to provide a table to store the effective standard errors

Description

In the analysis of variance of many balanced designs it is possible to provide a succinct description of the variability of a table of means, by giving an effective standard error (ese) for each mean. This can be used to calculate the standard error for the difference (sed) between any pair of means (i and j) using the usual formula:

$sed_{ij} = \sqrt{(ese_i^2 + ese_j^2)}$

In unbalanced designs, however, the standard errors may not possess such a simple structure. So it may be necessary to present the full symmetric matrice of sed's. This matrix has as many rows (and columns) as the number of means, and can be too large for many reports. The temptation therefore is to print just an average sed, but this can be very misleading. An alternative, provided by the SED2ESE procedure, is to estimate approximate ese's that allow good approximations to the sed's to be calculated using the usual formula.

The sed's to be approximated are supplied using the SED parameter, in a symmetric matrix. (This is the form in which they are saved from AKEEP, AUKEEP or PREDICT). The ese's can be saved using the ESE parameter. If no further information is supplied, they will be formed as a variate, with unit labels taken from the row labels of the SED symmetric matrix. Alternatively, you can predefine ESE as a table (which should have exactly the same form as the table of means to which the sed's refer). Or you can use the TEMPLATE parameter to provide a table (which could be the table of means itself) to act as a template for an ESE table. The DUPLICATE directive is then used to form ESE as a table with the same attributes as the template. The DISCREPANCY parameter can save a symmetric matrix containing the discrepancies between the sed's and the approximate values calculated from the ese's, and the %ACCOUNTED parameter can save a scalar indicating the percentage of the variation amongst the sed's accounted for by the approximate values calculated from the ese's.

Printed output is controlled by the PRINT option, with settings:

ese	the approximate effective standard errors;
discrepancy	discrepancies between the sed's and the approximate values calculated from the ese's;
maxdiscrepancy	maximum discrepancy;

`%discrepancy`	maximum discrepancy between any sed and the approximate value calculated from the corresponding ese's, expressed as a percentage of the sed;
`%accounted`	percentage of the variation amongst the sed's accounted for by the approximate values calculated from the ese's.

By default, nothing is printed.

Option: `PRINT`.
Parameters: `BLOCKFACTORS`, `TREATMENTFACTORS`, `LEVELS`.

Method
The ses's are estimated by fitting the equation in the formula by least squares using the standard Genstat regression facilities (see Menezes & Firth 1998).

Reference
Menezes, R. & Firth, D. (1998). More useful standard errors for group and factor effects. In: *COMPSTAT 1998, Proceedings in Computational Statistics, Short Communications and Posters* (ed. R. Payne & P. Lane), 79-80. IACR-Rothamsted, Harpenden.

See also
Procedures: `AUNBALANCED`, `SEDLSI`.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

SETDEVICE

Opens a graphical file and specifies the device number on basis of its extension (M.P. Boer & J.T.N.M. Thissen).

No options

Parameters

FILENAME = *texts*	Name of the graphical file including one of the possible extensions .bmp, .emf, .eps, .gmf, .jpg, .jpeg, .pdf, .png, .tif or .tiff; must be set
NUMBER = *scalars*	Saves the device number corresponding to the graphical format specified by parameter FILENAME
ACTION = *string token*	How to create graphs for file types such as .emf, .jpg, .tif or .png (asynchronous, synchronous); default asyn

Description

SETDEVICE opens a graphical file and specifies the device number on the basis of the extension of the given filename. The table below shows the correspondence between extension and device number.

Extension	Device number	Explanation
.bmp	2	Bitmap
.pdf	4	Portable Document Format
.eps	5	Encapsulated PostScript file
.emf	6	Windows Enhanced Metafile
.jpg or .jpeg	7	JPEG file
.tif or .tiff	8	TIFF file
.png	9	Portable Network Graphics file
.gmf	10	Genstat Metafile

The ACTION parameter controls how graphs are created for the file types .emf, .jpg, .jpeg, .tif, .tiff, .png, .gmf and .bmp. The setting synchronous creates the graph before executing another command, whereas the setting asynchronous allows subsequent commands to be executed whilst the graph is created.

Options: none.
Parameters: FILENAME, NUMBER, ACTION.

See also

Directive: DEVICE.
Genstat Reference Manual 1 Summary section on: Graphics.

SETNAME

Sets the identifier of a data structure to be one specified in a text (R.W. Payne).

No options

Parameters

DATA = *identifiers*	Specifies the data structures to be given new names (i.e. identifiers)
NAME =*texts*	Text for each data structure containing its new identifier

Description

SETNAME allows you to set a new identifier (or name) for a data structure. It differs from the similar RENAME directive in that the new identifier is supplied in a text structure. In RENAME the new identifier itself is specified.

The data structure to be given a new name is specified by the DATA parameter. If DATA is set to a dummy, SETNAME will operate on the data structure to which it points, not on the dummy itself. The text containing the new identifier is specified by the NAME parameter. This must be a simple identifier; identifiers with suffixes are not allowed. Also, note that the scope of the identifier will always be global (i.e. in your main program), even if SETNAME is called from inside another procedure.

For example, if you put

```
SETNAME A; NAME='B'
```

the data structure previously known as A would be renamed to have the identifier B, and the data structure previously known as B would lose its identifier and become unnamed. The identifier A would then no longer belong to anyone (and could if required be reused).

In the simplest situations, the first appearance of the new identifier will be in the SETNAME command. So there will be no consequences from the fact that the "orphan" data structure that it previously identified becomes unnamed. If the identifier has already been used, the orphan data structure will be deleted, unless it is found to belong to another (named) data structure. So, for example, if the full program was

```
SCALAR B; VALUE=1
POINTER [VALUES=B] Q
SETNAME A; NAME='B'
```

the scalar 1 would survive as the first element of the pointer Q. So it could still be referred to as Q[1], although of course no longer as B.

Options: none.
Parameters: DATA, NAME.

See also

Directive: RENAME.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Data structures.

SIGNTEST

Performs a one or two sample sign test (E. Stephens & P.W. Goedhart).

Options

PRINT = *string token*	Whether to print the test statistic with the associated probability and sample size (test); default test
METHOD = *string token*	Type of test (twosided, greaterthan, lessthan); default twos
GROUPS = *factor*	Defines the groups for a two-sample test if only the Y1 parameter is specified
NULL = *scalar*	Median value or difference in medians under the null hypothesis; default 0

Parameters

Y1 = *variates*	Data values for a one-sample sign test (neither Y2 nor GROUPS specified), or for the first sample of a two-sample test (Y2 also specified) or the values in both samples of a two-sample test (GROUPS specified but not Y2)
Y2 = *variates*	Data values for the second sample of a two-sample test
STATISTIC = *scalars*	To save the sign test statistic
NBINOMIAL = *scalars*	To save the effective sample size
PROBABILITY = *scalars*	To save the probability level of the test

Description

The sign test is a nonparametric test for difference in location between two related samples, or for testing the location of a single sample. The data values are specified by the parameters Y1 and Y2 and the option GROUPS. For a one-sample test, the Y1 parameter should be set to a variates containing the data. The data for a two-sample test can either be specified in two separate variates using the parameters Y1 and Y2. Alternatively, they can be given in a single variate, with the GROUPS option set to a factor to identify the two samples; the units are then assumed to be specified in the same order within each group. The GROUPS option is ignored when the Y2 parameter is set. The NULL option defines the size of the median under the null hypothesis for a one-sample test, or the difference between the two medians in a two-sample test. By default NULL=0.

The test is assumed to be two-sided unless otherwise requested by the METHOD option. Settings greaterthan or lessthan will give one-sided tests for the median or the difference between medians greater than, or less than, the null hypothesis value respectively.

In a one-sample test, units that are equal to the null hypothesis median are excluded and the effective sample-size is reduced. Similarly, in a two-sample test, units are excluded where the differences between the pairs of values are equal to that required by the null hypothesis. Units with missing values are also excluded.

By default, SIGNTEST prints the test statistic, the effective sample size and the (exact) probability level. This information can also be saved in named scalars using the STATISTIC, NBINOMIAL and PROBABILITY parameters repectively, and printing can be suppressed by setting option PRINT=*.

Options: PRINT, METHOD, GROUPS, NULL.
Parameters: Y1, Y2, STATISTIC, NBINOMIAL, PROBABILITY.

Method
The procedure uses standard Genstat directives for calculation and manipulation.

Action with `RESTRICT`
If the variates or the factor are restricted, the test is calculated using only the units not excluded by the restriction. In a two-sample test, the two variates or the variate and factor should be restricted in the same way. `RESTRICT` can be used for example to limit the data to only one or two groups when the `GROUPS` factor has more than two levels.

Reference
Siegel, S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also
Procedure: `SSIGNTEST`, `MANNWHITNEY`, `TTEST`, `WILCOXON`.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

SIMPLEX

Searches for the minimum of a function using the Nelder-Mead simplex algorithm (J.A. Nelder & W. van den Berg).

Options

PRINT = *string tokens*	Controls printed output (results, monitoring); default resu
CALCULATION = *expression structures*	Expressions to calculate the target function
FUNCTIONVALUE = *scalar*	Identifier of the scalar, calculated by CALCULATION, whose value is to be minimized
DATA = any type	Data to be used with procedure _SIMPLEXFUNCTION
POINTS = *pointer*	Saves the points of the final simplex
FVALUES = *pointer*	Saves the function values at the points
MAXCYCLE = *scalar*	Maximum number of iterations; default 500
TOLERANCE = *scalar*	Convergence criterion; when standard deviation of function values is lower than TOLERANCE convergence is assumed to be reached; default 1.E-9

Parameters

PARAMETER = *scalars*	Parameters to be estimated
LOWERINITIAL = *scalars*	Lower starting values for the parameters
UPPERINITIAL = *scalars*	Upper starting values for the parameters

Description

SIMPLEX uses the simplex algorithm devised by John Nelder & Roger Mead (1965) to search for the minimum of a function. The parameters to be estimated by the minimization are listed by the PARAMETER parameter of SIMPLEX. The optimum function value is found by constructing, and then moving, a "simplex" of points. The lower values for the parameters on the initial simplex are specified using the LOWERINITIAL parameter, and the upper values by the UPPERINITIAL parameter. All of these parameters must be set.

The function can be defined by specifying a list of Genstat calculation structures with the CALCULATION option, similarly to the way in which functions for optimization are specified for the FITNONLINEAR directive (see the *Guide to the Genstat Command Language, Part 2 Statistics*, Section 3.8). For example, in Section 2.9 of the *Guide to Regression, Nonlinear and Generalized Linear Models in Genstat*, an exponential model is fitted to a set crop yields. The model is

```
Yield = A + B * R ** Nitrogen + Residual
```

where Yield is the yield recorded on a plot, and Nitrogen is the amount of nitrogen fertiliser that was applied. To fit the model using the standard least-squares criterion, we need to calculate the sum of squares of the residual values. This can be done using the expressions E[1] and E[2], defined as follows:

```
EXPRESSION [VALUE= Fittedvalue = A + B * R**Nitrogen] E[1]
&          [VALUE= RSS = SUM ((Yield - Fittedvalue)**2)] E[2]
```

The FUNCTIONVALUE option defines the function value that is calculated by the expression (similarly to the FUNCTIONVALUE option of the MODEL directive). So, we can now estimate the parameters using the command

```
SIMPLEX    [PRINT=monitoring,results;\
           CALCULATION=E[1,2]; FUNCTIONVALUE=RSS]\
           A,B,R; LOWER=200,-140,0.98; UPPER=210,-120,0.99
```

Alternatively, more complicated functions can be specified by defining a procedure

_SIMPLEXFUNCTION, which operates similarly to the RESAMPLE procedure which is called by procedures BOOTSTRAP and JACKKNIFE. This is more complicated to specify, but it has the advantage that you can use any Genstat command to obtain the function value (e.g. ANOVA, FIT, SVD and so on). The DATA option is then used to list any data structures that are needed by _SIMPLEXFUNCTION to calculate the value of the function. Details are given in the Methods Section.

The PRINT option controls printed output with the settings:

results	to print numbers of iterations and function evaluations and the parameter estimates for the final simplex, and
monitoring	to print to monitor information showing the progress of the fit.

By default, PRINT=results.

The scalars specified by the PARAMETER parameter save the estimated values of the parameters. You can also use the POINTS option to save the points of the final simplex (in a pointer containing a variate for each point), and the function values at these points can be saved using the FVALUES option (as a pointer to a set of scalars).

The MAXCYCLE option sets a limit on the number of iterations; by default this is 500. The TOLERANCE option controls the convergence criterion (default 1.E-10). When the standard deviation of the function values around the simplex is less than or equal to *Limit* the algorithm stops. *Limit* is defined as TOLERANCE multiplied by the standard deviation of the function values on the initial simplex, or 1.E-10 if the initial variance is zero.

Options: PRINT, CALCULATION, FUNCTIONVALUE, DATA, POINTS, FVALUES, MAXCYCLE, TOLERANCE.
Parameters: PARAMETER, LOWERINITIAL, UPPERINITIAL.

Method

The simplex method of Nelder & Mead (1965) searches for the minimum function value by constructing, and then moving, a "simplex" of points around the parameter space. SIMPLEX uses a revised version of the algorithm which deals with the failed contraction position. This is important if the process is not to become stuck when there are steep curved valleys in the surface. The following steps are possible at each iteration:

E	worst point replaced by expanded point;
FE	worst point replaced by reflected point;
C	contracted point on better side when reflected point worst;
R	replace worst point by initial expanded point.

If, after step C, the new point is still the worst point, the points are shrunk to best point, and FC is printed in the monitoring output.

The algorithm thus searches directly for a minimum, rather than for zeros of the derivative of the function. It does not assume knowledge of derivatives, and it generally works rather well when there is some noise in the pointwise evaluation of the function itself. However, it is not fast, particularly in higher dimensions. (See Thompson 1998.)

The procedure _SIMPLEXFUNCTION, which you can use to calculate the function instead of the CALCULATION and FUNCTIONVALUE options, has two options. DATA supplies a pointer containing the data structures specified by the DATA option of SIMPLEX (so, DATA[1] is the first of these structures, DATA[2] is the second, and so on). FUNCTIONVALUE is a scalar, which should be set to the function value. There is one parameter, called PARAMETER. The PROCEDURE statement that defines _SIMPLEXFUNCTION should set option PARAMETER=pointer. The parameters of the function can then be referred to as PARAMETER[1], PARAMETER[2], and so on (and these will be in the same order as in the PARAMETER parameter of SIMPLEX). The definition below has the same effect as the two expressions

```
EXPRESSION [VALUE= Fittedvalue = A + B * R**Nitrogen] E[1]
&          [VALUE= RSS = SUM ((Yield - Fittedvalue)**2)] E[2]
```

shown in the description.

```
PROCEDURE [PARAMETER=pointer] '_SIMPLEXFUNCTION'
" calculates the function for SIMPLEX "
OPTION NAME=\
  'DATA',     "(I: any type) data to calculate the function"\
  'FUNCTIONVALUE'; "(O: scalar) returns the function value" \
  MODE = p; \
  TYPE = *, 'scalar'; \
  LIST = yes, no
PARAMETER NAME=\
  'PARAMETER';  "(I: scalar) parameter values" \
  MODE = p; \
  TYPE = 'scalar'; \
  SET = yes; \
  DECLARED = yes; \
  PRESENT = yes
CALCULATE   Fittedvalue   =   PARAMETER[1]   +   PARAMETER[2]   *
PARAMETER[3]**DATA[2]
&         FUNCTIONVALUE = SUM((DATA[1] - Fittedvalue)**2)
ENDPROCEDURE
```

The parameters can then be estimated by the statement

```
SIMPLEX    [PRINT= monitoring,results; DATA=Yield,Nitrogen]\
           A,B,R; LOWER=200,-140,0.98; UPPER=210,-120,0.99
```

Action with `RESTRICT`

The effects of restrictions on the data variables will depend on how the calculation is defined (by the `CALCULATION` option or within the `_SIMPLEXFUNCTION` procedure).

References

Nelder, J.A. & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, **7**, 303-333.

Thompson, J.R. (2000). *Simulation: a Modeler's Approach*. Wiley, New York.

See also

Directive: `FITNONLINEAR`.

Procedures: `FPARETOSET`, `MINIMIZE`, `MIN1DIMENSION`.

Genstat Reference Manual 1 Summary section on: Regression analysis.

SKEWSYMMETRY

Provides an analysis of skew-symmetry for an asymmetric matrix (P.G.N. Digby).

Option

PRINT = *string tokens*	Printed output from the analysis (roots, scores); default * i.e. no output

Parameters

DATA = *matrices*	Asymmetric (square) matrices to be analysed
ROOTS = *diagonal matrices*	Stores the squared singular values from the analysis; the structure has one value for each plane fitted in the analysis (e.g. if the DATA matrix has 11 rows and columns, the ROOTS diagonal matrix will have 5 values)
SCORES = *matrices*	Stores the coordinates of the points from the analysis; each matrix has the same number of rows as the corresponding DATA matrix, and has 2 columns for each plane fitted in the analysis (e.g. if the DATA matrix has 11 rows and columns, the SCORES matrix will have 11 rows and 10 columns)

Description

Procedure SKEWSYMMETRY provides the canonical analysis of skew-symmetry described by Gower (1977). The input to the procedure, specified by the parameter DATA, is a (square) asymmetric matrix of associations, A say. The rows and columns of A usually represent the same set of objects, but in different modes. For example, with migration data, the rows may represent the Countries or States being departed from, and the columns the same locations but being arrived at. The DATA matrix must not contain any missing values.

The results of the analysis are a set of coordinates (SCORES) for points representing the entities labelling the rows or columns of the DATA matrix. In pairs, these coordinates give positions on a series of planes, also called bimensions. So there is an even number of coordinates for each point; if the DATA matrix has an odd number of rows/columns, there will be one fewer coordinate than the number of rows or columns of the DATA matrix. Also, the "importance" of each plane can be assessed from a set of values (ROOTS) that give the amount of (squared) skew-symmetry explained in each pair of dimensions.

The results are interpreted in terms of the areas of triangles. The skew symmetry between the entities in rows (or columns) p and q is proportional to the area of the triangle OPQ, where O is the origin, and P and Q are the points representing p and q respectively. (For further details see either Gower 1977, or Digby & Kempton 1987.) Within each plane the coordinates are arranged so that their centroid is at $(0, y)$, for $y \geq 0$, and so that positive row-to-column skew symmetry is represented in a clockwise direction. (Note that in planes other than the first it is residual skew symmetry, after fitting the preceding planes, that is being modelled).

Printed output is controlled by the PRINT option: roots prints the roots, also the roots expressed as percentages and cumulative percentages, and scores prints the scores.

Results from the analysis can be saved using the parameters ROOTS and SCORES. The structures specified for these parameters need not be declared in advance. Column labels are provided automatically for the SCORES matrix, but any row labels (useful to identify the entities) are left unchanged.

Option: PRINT.
Parameters: DATA, ROOTS, SCORES.

Method

Procedure SKEWSYMMETRY provides the analysis of skew-symmetry of Gower (1977). If A is an asymmetric matrix of associations, then $S = A - A'$ is skew-symmetric; this matrix is analysed using a singular value decomposition, followed by a reflection and rotation, to provide the necessary roots and scores. For further details see Gower (1977) or Digby & Kempton (1987).

References

Digby, P.G.N. & Kempton. R.A. (1987) *Multivariate Analysis of Ecological Communities*. Chapman & Hall, London.

Gower, J.C. (1977) The analysis of asymmetry and orthogonality. In: *Recent Developments in Statistics* (ed. J. Barra, F. Brodeau, G. Romier & B. van Cutsen), 109-123. North Holland, Amsterdam.

See also

Genstat Reference Manual 1 Summary section on: Multivariate and cluster analysis.

SLCONCORDANCE

Calculates the sample size for Lin's concordance correlation coefficient (R.W. Payne).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
PROBABILITY = *scalar*	Significance level at which the non-reproducibility is to be tested; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
REPLICATION = *variate*	Replication values for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

CORRELATION = *scalars*	Correlation for two samples with the smallest amount of non-reproducibility required to be detected
CONCORDANCE = *scalars*	Value of Lin's concordance for two samples with the smallest amount of non-reproducibility required to be detected
MEANSHIFT = *scalars*	Value of the shift in means (divided by the harmonic mean of the standard deviations) for two samples with the smallest amount of non-reproducibility required to be detected
SDRATIO = *scalars*	Value of the ratio of the standard deviations for two samples with the smallest amount of non-reproducibility required to be detected
NREPLICATES = *scalars*	Saves the required number of replicates
VREPLICATION = *variates*	Numbers of replicates for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

Lin's concordance correlation coefficient can be used to assess how well a new method of measurement reproduces the results provided by a standard method. To do this, you measure the same set of units using the two methods, and calculate the concordance between the resulting two sets of measurements. The methods are regarded as equivalent if the coefficient is greater than some threshold. SLCONCORDANCE helps you to decide how many units need to be measured to make a reliable assessment.

The concordance coefficient is defined by the equation

$$\rho_c = \rho \times C_b$$

(see Lin 1989, 2000 or procedure LCONCORDANCE). The term ρ is the standard Pearson product-moment correlation coefficient, while C_b is a bias correction factor which is calculated by

$$C_b = 2 / (v + 1/v + u^2)$$
$$v = s_1 / s_2$$
$$u = (m_1 - m_2) / \sqrt{(s_1 \times s_2)}$$

where m_i and s_i $(i = 1,2)$ are the mean and standard deviation of the i^{th} set of measurements. The quantity u represents the shift in the mean between the two sets of measurements divided by the harmonic mean of their standard deviations, while v is the ratio of the two standard deviations.

If the coefficient is given a Z-transformation, the result has an approximate Normal

distribution, with a standard deviation that depends on ρ_c, ρ and u (see Lin 1989, 2000). So, to calculate the sample size, SLCONCORDANCE needs to know the values of these quantities for two sets of measurements displaying the smallest amount of non-reproducibility that is required to be detected. The correlation coefficient (ρ) is specified by the CORRELATION parameter, the concordance coefficient by the CONCORDANCE parameter, and u by the MEANSHIFT parameter. Alternatively, you can omit either CONCORDANCE or MEANSHIFT provided you specify the ratio of the standard deviations, v, using the SDRATIO parameter. (SLCONCORDANCE can then calculate the omitted quantity using the equations in the previous paragraph.)

The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%). This is for a one-sided test, on the basis that you would not reject a new method for being too similar to the standard method. (Note, this also corresponds to a test for non-inferiority; see the Methods section of the documentation for procedure STTEST.) The required probability for detecting non-reproducibility (that is the *power* of the test) is specified by the POWER option (default 0.9).

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates to measure using each method (i.e. the sample size);
power	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding powers can also be saved, in variates, using the VREPLICATION and VPOWER parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, PROBABILITY, POWER, RATIOREPLICATION, REPLICATION.
Parameters: CORRELATION, CONCORDANCE, MEANSHIFT, SDRATIO, NREPLICATES, VREPLICATION, VPOWER.

Method

The calculation uses the fact that the transformation

$$Z = 0.5 \times (\log(1 + \rho_c) / \log(1 - \rho_c))$$

has an approximate Normal distribution, with a standard deviation defined by Lin (2000). Note, the results produced by SLCONCORDANCE do not match those of Lin (1992) firstly because of the correction to the equation for the standard deviation noted by Lin (2000), and secondly because the equation for n on page 601 of Lin (1992) should read

$$n = ((\text{EDNORMAL}(1-\beta) + \text{EDNORMAL}(1-\alpha)) * S / (Z - Z_{c,a}))^2 + 2.$$

References

Lin, L.I. (1989). A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, **45**, 255-268.

Lin, L.I. (1992). Assay validation using the concordance correlation coefficient. *Biometrics*, **48**, 599-604.

Lin, L.I. (2000). A note on the concordance correlation coefficient. *Biometrics*, **56**, 324-325.

See also

Procedure: LCONCORDANCE.
Genstat Reference Manual 1 Summary section on: Design of experiments.

SMANNWHITNEY

Calculates the sample sizes for the Mann-Whitney test (R.W. Payne).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
PROBABILITY = *scalar*	Significance level at which the test is to be made; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Whether to a one- or two-sided test is to be made (onesided, twosided); default twos
RATIOREPLICATION = *scalar*	Ratio of replication sample2:sample1 (i.e. the size of sample 2 should be RATIOREPLICATION times the size of sample 1); default 1
REPLICATION = *variate*	Sample sizes for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

NULLPROBABILITIES = *variates*	Probabilities under null hypothesis
ODDSRATIO = *scalars*	Odds ratio for test group vs. control
NREPLICATES = *scalars*	Saves the required sample size
VREPLICATION = *variates*	Sample sizes for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

The Mann-Whitney U test is a nonparametric test for differences in location between two samples (see procedure MANNWHITNEY). This procedure, SMANNWHITNEY, allows you to calculate the sample sizes required for the test, provided you can supply some information about the probability distributions from which the samples are likely to be generated. For simplicity, the data are assumed to be classified into ordered categories. These may be natural categories (such as "very good", "good", "moderate" and "poor") or they may be formed by splitting a continuous scale intervals (e.g. "under 18", "18-25", "25-40", "40-60" and "over 60"). You then use the NULLPROBABILITIES parameter to specify a variate containing the probability value for each category. This indicates the probability distribution which you feel would generate the data of both samples under the null hypothesis. The accuracy of the subsequent calculations will depend on how many categories you take for a continuous variate. However, Whitehead (1993) suggests that there is little to gain in taking more than five.

To assess the power of the test, you next need to indicate how small a difference between the sample distributions the test should be able to detect. The assumption now is that there will be a control sample, with probability distribution as supplied, and a test sample for which the distribution is shifted by multiplying the odds (i.e. $p/(1-p)$) of the cumulative distribution by a constant amount. (This corresponds to the proportional-odds model of McCullagh 1980.) This constant is supplied by the ODDSRATIO parameter. An example, with odds-ratio 2, is show below.

Null hypothesis			Alternative hypothesis		
probability	cumulative probability	odds	probability	cumulative probability	odds
0.20	0.20	0.25	0.33	0.33	0.50
0.40	0.60	1.50	0.42	0.75	3.00
0.30	0.90	9.00	0.20	0.95	18.00
0.10	1.00	*	0.05	1.00	*

The cumulative probabilities are produced as part of the information generated by setting the PRINT option to power. So you can evaluate possible ratios to check that they generate plausible distributions.

By default the calculations are done for a one-sided test, but you can set option TMETHOD=twosided for a two-sided test instead. The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%). The required probability for detection of the change (that is, the *power* of the test) is specified by the POWER option (default 0.9). It is generally assumed that the sizes of the samples in the two-sample test should be equal. However, you can set the RATIOREPLICATION option to a scalar, R say, to indicate that the size of the second sample should be R times the size of the first sample. The sample size can be saved using the NREPLICATES parameter.

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates in each sample (i.e. the size of each sample);
power	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding powers can also be saved, in variates, using the VREPLICATION and VPOWER parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, PROBABILITY, POWER, TMETHOD, RATIOREPLICATION, REPLICATION.
Parameters: NULLPROBABILITIES, ODDSRATIO, NREPLICATES, VREPLICATION, VPOWER.

Method

The method is based on the equations given by Whitehead (1993), except the Genstat implementation omits the approximation of taking $n/(n+1)$ as equal to one.

References

McCullagh, P. (1980). Regression models for ordinal data. *Journal of the Royal Statistical Society Series B*, **43**, 109-142.

Whitehead, J. (1993). Sample size calculations for ordered categorical data. *Statistics in Medicine*, **12**, 2257-2271.

See also

Procedure: MANNWHITNEY.

Genstat Reference Manual 1 Summary section on: Design of experiments.

SMCNEMAR

Calculates sample sizes for McNemar's test (R.W. Payne).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
PRMETHOD = *string token*	Method to be used to calculate the power of the McNemar test (normalapproximation, exact); default exac
PROBABILITY = *scalar*	Significance level at which the test is to be made; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Whether a one- or two-sided test is to be made (onesided, twosided); default twos
REPLICATION = *variate*	Sample sizes for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

CHANGEPROBABILITY = *scalars*	Probability of any sort of change
RATIOPROBABILITIES = *scalars*	Ratio of the two probabilities of change
NREPLICATES = *scalars*	Saves the required sample size
VREPLICATION = *variates*	Sample sizes for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

The McNemar test is useful for analysing studies where subjects are assessed before and after a treatment. The response on each occasion is assumed to be categorized by a factor with two levels, with level 1 usually representing a *negative* response, and level 2 a *positive* response. The test is based on a table giving the numbers of subjects giving each combination of responses over the two occasions. Suppose that the table contains the values A, B, C and D as below:

	Second occasion	
First occasion	negative	positive
positive	A	B
negative	C	D

The test statistic assesses the equality of A and D, which represent the changes from positive to negative, and negative to positive, respectively. See procedure MCNEMAR or Siegel (1956), pages 63-67.

In its original form, the test leads to a chi-square test. However, this may be inaccurate when there are small numbers of subjects. Consequently procedure MCNEMAR also provides an exact probability (based on the binomial distribution). Similarly SMCNEMAR has an option, PRMETHOD, to select whether you want to calculate the power of the test by approximating the probabilities by a Normal distribution, or using the binomial distribution as in the exact calculation (settings normalapproximation and exact, respectively). The default is exact.

To calculate the sample size, SMCNEMAR needs to know the overall probability of change (i.e.

the probability of a subject being amongst those in either A or D), and the ratio of the probabilities of the two types of change (A versus D). These are specified by parameters CHANGEPROBABILITY and RATIOPROBABILITIES, respectively. By default the calculations are done for a one-sided test (testing for evidence that the change is in a specific direction (e.g. negative to positive). However, you can set option TMETHOD=twosided for a two-sided test (testing for either type of change). The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%). The required probability for detection of the change (that is, the *power* of the test) is specified by the POWER option (default 0.9). The sample size can be saved using the NREPLICATES parameter.

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates in each sample (i.e. the size of each sample);
power	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding powers can also be saved, in variates, using the VREPLICATION and VPOWER parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, PRMETHOD, PROBABILITY, POWER, TMETHOD, REPLICATION.
Parameters: CHANGEPROBABILITY, RATIOPROBABILITIES, NREPLICATES, VREPLICATION, VPOWER.

Method

The sample size is first calculated by taking a Normal approximation to the probabilities:

```
NREPLICATES = CEILING( ( \
        ( ( EDNORMAL(POWER) * SQRT(prob * (1 - prob))\
          - EDNORMAL(alpha) * 0.5 )\
          / (prob - 0.5) )**2 ) / CHANGEPROBABILITY)
```

where alpha is the significance level for the null hypothesis, and prob is the minimum of

```
1 / (1 + RATIOPROBABILITIES)
```

and

```
RATIOPROBABILITIES / (1 + RATIOPROBABILITIES)
```

With the exact calculation, this provides an initial estimate for a search for the required size, with probabilities calculated using the binomial distribution. Note: the exact calculation generally leads to sample sizes about 10% larger than those derived using the Normal approximation.

Reference

Siegel S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Procedure: MCNEMAR.
Genstat Reference Manual 1 Summary section on: Design of experiments.

SMOOTHSPECTRUM

Forms smoothed spectrum estimates for univariate time series (G. Tunnicliffe Wilson & S.J. Welham).

Options

PRINT = *string token*	Controls printed output (description); default desc
METHOD = *string token*	Method to be used for smoothing (lagwindow, direct, YuleWalker, exactautoregressive); default lagw
BANDWIDTH = *scalar*	Frequency domain bandwidth for the smoothing window; must be set if METHOD=dire
MAXLAG = *scalar*	Specifies the cut-off lag (i.e. the maximum lag of autocovariance used in the spectrum calculation) for METHOD=lagw, or the order of the autoregression for METHOD=Yule or exac; if this option is not set then BANDWIDTH must be set, and will be used to determine an appropriate value of MAXLAG
DIVISIONS = *scalar*	Determines the number of frequency divisions into which the range [0.0, 0.5] is divided for calculating the spectrum; the default is chosen so that the bandwidth covers about four intervals
PROBABILITY = *scalar*	Probability value used for confidence limits; default 0.9
TAPER = *scalar*	The proportion of data to be tapered (applied for all settings of METHOD except exac); default 0.0
SHAPE = *scalar*	The shape of the trapezium window (a value of 1.0 specifies a rectangular, and 0.0 a triangular window); default 0.5
YLOG = *string token*	Whether to plot with a log-transformed Y-axis (yes, no); default no
XLOG = *string token*	Whether to plot with a log-transformed X-axis (yes, no); default no
GRAPHICS = *string token*	What sort of graphics to use (lineprinter, highresolution); default high
WINDOW = *scalar*	Window to be used for plotting; default 1
PENS = *variate*	The two pens to be used (after being defined appropriately) for drawing the plots; default !(1,2)

Parameters

SERIES = *variates*	The series for which the spectrum is to be calculated
LENGTH = *scalars* or *variates*	Scalar specifying that the first *N* units of the series are to be used, or a variate specifying the first and last units of the series to be used
SPECTRUM = *variates*	Saves the smoothed spectrum; need not be declared in advance, but will be set up as a variate of the appropriate length within the procedure
LOWER = *scalars* or *variates*	Scalar to save the multiplier of the spectrum used to calculate the lower limit, or a variate to save the values of the lower limit
UPPER = *scalars* or *variates*	Scalar to save the multiplier of the spectrum used to calculate the upper limit, or a variate to save the values of the upper limit
FREQUENCY = *variates*	Saves the frequency values at which the spectrum is

calculated

Description

SMOOTHSPECTRUM calculates smoothed spectrum estimates for a univariate time series. The series is specified in a variate by the SERIES parameter. The parameter LENGTH can be used to specify that only part of the series is to be used: if LENGTH is set to a scalar N, then only units 1...N are used; alternatively, it can define a sub-series by being set to a variate of length 2 holding the numbers of the first and last units to be used. The spectrum can be saved by the SPECTRUM parameter. The method to be used for the smoothing is controlled by the METHOD option, with settings lagwindow for Parzen lag window smoothing, direct for frequency domain smoothing using a trapezium window, YuleWalker for autoregressive spectrum estimation based on Yule-Walker coefficients, and exactautoregressive for autoregressive estimation based on exact likelihood estimation of the coefficients.

For frequency domain smoothing (METHOD=direct), option BANDWIDTH specifies the bandwidth of the smoothing window and option SHAPE the shape of the trapezium window. The BANDWIDTH option is also used to determine an appropriate default for the MAXLAG option if this is not specified with other METHOD settings: for METHOD=lagwindow, MAXLAG specifies the cut-off lag (i.e. the maximum lag of autocovariance used in the spectrum calculation), while for METHOD=YuleWalker or exactautoregressive, it specifies the order of the autoregression.

The DIVISIONS option can define the number of frequency divisions into which the range [0.0, 0.5] is divided for calculating the spectrum; if this is omitted a default is chosen so that the bandwidth covers about four intervals. The frequency values at which the spectrum is calculated can be saved, in a variate, by the FREQUENCY parameter. The proportion of data to be tapered (relevant to all settings of METHOD except exactautoregressive) is controlled by the TAPER option; by default there is no tapering.

The LOWER and UPPER parameters can be set to scalars to save the scaling factor used to calculate the upper and lower bounds, or to variates to save the upper and lower bounds for the SPECTRUM variate.

Printed output can be suppressed by setting the option PRINT=*; by default, PRINT=description. The PROBABILITY option indicates the probability value used for confidence limits; 0.9 is used as the default.

The procedure will also plot the spectrum: option GRAPHICS controls whether this is done for line printer or on a high-resolution device. With high-resolution graphics, the plot will be produced using the current settings of the window specifed by the WINDOW option; by default WINDOW=1. The FRAME directive can be used to set the attributes of the window prior to calling the procedure. The PENS option controls which pens are to be used for the plots; the attributes of these pens are modified within the procedure. By default pens 1 and 2 are used, but these can be changed by setting option PENS to a variate of length 2 containing the numbers of the two pens required. Options YLOG and XLOG allow the X- and Y-axes to be represented on a logarithmic scale.

Options: PRINT, METHOD, BANDWIDTH, MAXLAG, DIVISIONS, PROBABILITY, TAPER, SHAPE, YLOG, XLOG, GRAPHICS, WINDOW, PENS.
Parameters: SERIES, LENGTH, SPECTRUM, LOWER, UPPER, FREQUENCY.

Method

A cosine bell window is used for the taper, with lag window and direct spectral smoothing carried out esentially as described in Bloomfield (1976). The autoregressive spectrum estimation uses the standard Yule-Walker equations, as presented for example in Box & Jenkins (1970). These are optionally refined by exact maximum likelihood estimation. The theoretical spectrum of the autoregressive model is then calculated. The error limits are calculated using scaled chi-

square distributions. These are quite good for the case of lag window and direct smoothing, but in small samples are only very approximate for the autoregressive estimates. The series values are mean corrected before spectrum estimation, but not trend corrected.

Action with `RESTRICT`
Input and output structures must not be restricted; restriction of the input series to a contiguous set of units can be achieved by use of the `LENGTH` parameter.

References
Bloomfield, P. (1976). *Fourier Analysis of Time Series: an Introduction*. Wiley, New York.
Box, G.E.P. & Jenkins, G.M. (1970). *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco.

See also
Directive: `FOURIER`.
Procedures: `DFOURIER`, `MCROSSPECTRUM`, `PERIODTEST`, `PREWHITEN`, `REPPERIODOGRAM`.
Genstat Reference Manual 1 Summary section on: Time series.

SOM

Declares a self-organizing map (R.W. Payne).

No options

Parameters

IDENTIFIER = *identifiers*	Identifiers of the SOMs
VARIABLENAMES = *texts*	Names of variables corresponding to the weights of each SOM
ROWS = *scalars* or *variates*	Number of rows or row coordinates for the map
COLUMNS = *scalars* or *variates*	Number of columns or column coordinates for the map
DMETHOD = *string tokens*	Method for calculating the distances of data points from the modes (euclidean, cityblock); default eucl
WMETHOD = *string tokens*	Method for calculating the contribution of a data point to each node when revising the weights (gaussian, neighbour); default gaus

Description

A self-organizing map is a two dimensional grid of nodes, used to classify vectors of observations on p variables. Each node is characterized by a vector of p weights (one for each variable). SOM defines the Genstat data structures used to represent self-organizing maps. These are compound data structures similar, for example, to the LRV structure used to store latent roots and vectors (see the LRV directive). Compound data structures are like Genstat pointers in that they point to a set of other structures. However, the set has a fixed size, its elements must be of the correct types, and must form a consistent set (in terms of their sizes and so on). You can refer to the elements of an SOM in exactly the same way as the elements of a pointers, but the suffixes and their labels are fixed. Unlike pointers, the labels are not case sensitive, so Genstat will recognize the label in either upper-case or lower-case letters or in any mixture of the two.

The elements of an SOM are as follows:

[1] or ['variablenames']	text containing the names of the variables;
[2] or ['rows']	factor giving the row position of each node;
[3] or ['columns']	factor giving the column position of each node;
[4] or ['dmethod']	text containing either 'EUCLIDEAN' or 'CITYBLOCK' indicating the method used to measure distance on the map;
[5] or ['wmethod']	text containing either 'GAUSSIAN' or 'NEIGHBOUR' indicating the method used to adjust the weights at each iteration during their estimation;
[6] or ['weights']	matrix of weights (variables × nodes);
[7] or ['summaries']	pointer to store variates of summaries of variables at the modes of the map;
[8] or ['smethods']	text indicating the method used to summarize the variable in each variate of summaries;
[9] or ['svariablenames']	text indicating the variable that was summarized in each variate of summaries.

The SOM procedure defines the SOM, and forms its first five elements. The weights (element 6) can be estimated and stored in the SOM by the SOMESTIMATE procedure, and the summary information (elements 7-9) can then be formed and added by the SOMDESCRIBE procedure. Once this has been done, the SOMPREDICT procedure can be used to generate predicted values of the

summary variables for new or hypothetical observations.

The identifier for the SOM is specified by the IDENTIFIER parameter. The names of variables corresponding to the weights are provided in a text specified by the VARIABLENAMES parameter. The row and column positions of the nodes are specified by the ROWS and COLUMNS options. These can be set to scalars, specifying the numbers of rows and columns in a rectangular grid. The row and column coordinates are then positive integers starting at one. Alternatively, you can define your own row and column coordinates (which then need not be in a rectangular grid), by setting ROWS and COLUMNS to variates. By default, ROWS is 5 and COLUMNS is 6. The distance and weighting methods are specified by the DMETHOD and WMETHOD options, respectively.

Options: none.
Parameters: IDENTIFIER, VARIABLENAMES, ROWS, COLUMNS, DMETHOD, WMETHOD.

Method
For further information, see Hastie, Tibshirani & Friedman (2001) Section 14.4.

Reference
Hastie, T., Tibshirani, R, & Friedman, J. (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, New York.

See also
Procedures: SOMADJUST, SOMDESCRIBE, SOMESTIMATE, SOMIDENTIFY, SOMPREDICT.
Genstat Reference Manual 1 Summary section on: Data mining.

SOMADJUST

Performs adjustments to the weights of a self-organizing map (R.W. Payne).

Options

SOM = *pointer*	Self-organizing map
DATA = *matrix* or *pointer*	Data values for training the map
DMETHOD = *string token*	Method for calculating the distances of data points from the modes (euclidean, cityblock); default eucl
WMETHOD = *string token*	Method for calculating the contribution of a data point to each node when revising the weights (gaussian, neighbour); default gaus

Parameters

ALPHA = *scalars*	Alpha value for each iteration
SIGMA = *scalars*	Sigma value for each iteration when WMETHOD=gaussian
THRESHOLD = *scalars*	Threshold for each iteration when WMETHOD=neighbour
ERRORS = *matrices*	Saves the reconstruction errors at the nodes of the map after each iteration
TOTALERROR = *scalars*	Saves the total reconstruction error after each iteration
FITNODES = *factors*	Saves the nodes allocated to the data points after each iteration

Description

A self-organizing map is a two dimensional grid of nodes, used to classify vectors of observations on p variables. Each node is characterized by a vector of p weights (one for each variable). Genstat has a special SOM data structure to represent a map. This is declared using the SOM procedure, which also defines the row and column positions of the nodes on the grid. In addition, SOM stores the names of the weight variables, and information about how distances are to be measured on the grid and how the weights should be adjusted during their estimation.

The training dataset to estimate the weights is specified by the DATA option, either as a matrix with *n* rows and *p* columns (where *n* is the number of observations in the training set) or as a pointer containing *p* variates each with *n* units. SOMADJUST gives a warning if the row names of a DATA matrix or the names of the variates in a DATA pointer differ from the names stored for the weight variables in the SOM structure.

The weights are estimated by a sequence of iterations. In each iteraction, the training observations are taken in turn. Each observation *i* is assessed to find its closest node. The method to use to measure distance on the map will have been specified, by the DMETHOD option of SOM, and stored with the SOM structure when it was declared. However, SOMADJUST also has a DMETHOD option in case you want to override the stored setting. The default setting for the DMETHOD option of SOM is euclidean. If X_i is a variate containing the values of the variables for observation *i* and W_k is the variate of weights at node *j*, the distance is then given by

```
d_ij = SQRT(SUM((X_i - W_j)**2))
```

The alternative setting, cityblock, calculates the distance as

```
d_ij = SUM(ABS(X_i - W_j))
```

Once the closest node, *k*, has been found, the weights at that node and other nodes are adjusted. The method to use will have been specified when the SOM structure was declared, by the WMETHOD option of SOM. However, SOMADJUST again has its own WMETHOD option, that you can use to override the stored setting. The default setting for the DMETHOD option of SOM is gaussian. This adjusts the weights W_j at every node *j* to become

```
W_j + alpha * EXP( -0.5 * (d_jk / sigma)**2) * (X_i - W_j)
```

where `d_jk` is the distance between nodes *j* and *k*. With the alternative setting, `neighbour`, the weights at node *j* are adjusted to become

```
W_j + alpha * (X_i - W_j)
```

but only if `d_jk` is less than a threshold `r`.

The values of `alpha`, `sigma` and `r` for the iterations are listed by the `ALPHA`, `SIGMA` and `THRESHOLD` parameters of `SOMADJUST`. Each of these supplies a list of scalars (one for each iteration). The `ERRORS` parameter can save a list of matrices containing reconstruction error at the nodes of the map after each iteration. The `TOTALERROR` parameter can save a list of scalars with the total reconstruction error after each iteration. Finally, the `FITNODES` parameter can save a list of factors indicating how the observations are allocated to the nodes by each iteration.

`SOMADJUST` thus allows you define your own sequence of adjustment iteractions leading to the estimation of the weights. An alternative is to use procedure `SOMESTIMATE`, which initializes the weights and runs through an automatic sequence of iterations (each performed using `SOMADJUST`).

Options: `SOM`, `DATA`, `DMETHOD`, `WMETHOD`.
Parameters: `ALPHA`, `SIGMA`, `THRESHOLD`, `ERRORS`, `TOTALERROR`, `FITNODES`.

Action with `RESTRICT`

`SOMADJUST` takes account of any restrictions defined on the `DATA` variates.

See also

Procedures: `SOM`, `SOMDESCRIBE`, `SOMESTIMATE`, `SOMIDENTIFY`, `SOMPREDICT`.
Genstat Reference Manual 1 Summary section on: Data mining.

SOMDESCRIBE

Summarizes values of variables at nodes of a self-organizing map (R.W. Payne).

Options

PRINT = *string token*	Controls whether or not the summaries are printed (summaries); default summ
DATA = *matrix* or *pointer*	Data values to identify the positions of the samples on the map
SOM = *pointer*	Specifies the map
NEWSOM = *pointer*	Saves the map, augmented by the summary information

Parameters

Y = *variates* or *factors*	Data values to be summarized
METHOD = *string tokens*	How to summarize each Y (mean, mode, median, minimum, maximum, sd, variance); default mode for factors, mean for variates

Description

A self-organizing map is a two dimensional grid of nodes, used to classify vectors of observations on *p* variables. Each node is characterized by a vector of *p* weights (one for each variable); these can be estimated, from a training dataset, by procedure SOMESTIMATE. This procedure, SOMDESCRIBE, allows you to allocate observations to the nodes of a map and form summaries of their values.

The SOM option supplies the information about the self-organizing map, which will have been saved in a pointer using the SOM parameter of SOMESTIMATE. The DATA option supplies the variables required to identify the positions of the samples on the map, either as a matrix with *n* rows and *p* columns (where *n* is the number of samples) or as a pointer containing *p* variates each with *n* units. The SOMIDENTIFY procedure, called by SOMDESCRIBE to identify the positions, will issue a warning if the variables have different names to those in the data set used by SOMESTIMATE to form the map. The NEWSOM option can be used to save an extended form of the self-organizing map which also contains the summary information. This extended map can then be used by SOMPREDICT to form predictions for future observations.

The information to be summarized at each node is supplied by the Y parameter, as a list of variates and/or factors. The METHOD parameter supplies a list of strings, defining how each one is to be summarized: either mean, mode, median, minimum, maximum, sd (i.e. standard deviation) or variance.

The PRINT option controls whether or not the summaries are printed (by default they will be printed).

Options: PRINT, DATA, SOM, NEWSOM.
Parameters: Y, METHOD.

Method

The SOMIDENTIFY procedure is used to allocate the samples to the nodes of the map. The TABMODE procedure is used to form modes, and the TABULATE directive to form the other types of summary.

Action with RESTRICT

SOMDESCRIBE takes account of any restrictions defined on the Y variates or factors.

See also
Procedures: SOM, SOMADJUST, SOMESTIMATE, SOMIDENTIFY, SOMPREDICT.
Genstat Reference Manual 1 Summary section on: Data mining.

SOMESTIMATE

Estimates the weights for self-organizing maps (R.W. Payne).

Options

PRINT = *string tokens*	Controls output (weights, errors, monitoring, report); default weig, repo
PLOT = *string token*	Controls what to plot (fit, totalerror); default fit
DMETHOD = *string token*	Method for calculating the distances of data points from the modes (euclidean, cityblock); default eucl
WMETHOD = *string token*	Method for calculating the contribution of a data point to each node when revising the weights (gaussian, neighbour); default gaus
ALPHA = *scalar* or *variate*	Initial alpha value for each set of iterations; default !(1, 0.1)
SIGMA = *scalar* or *variate*	Initial sigma value for each set of iterations when WMETHOD=gaussian; default !(1, 0.01) multiplied by the maximum distance between nodes
THRESHOLD = *scalar* or *variate*	Initial distance threshold for each set of iterations when WMETHOD=neighbour; default !(0.5, 0.1) multiplied by the maximum distance between nodes
NCYCLE = *scalar* or *variate*	Number of cycles in each set of iterations; default 500
NSTOP = *scalar*	Number of consecutive cycles with no changes required for convergence; default 10

Parameters

SOM = *pointers*	Save the information about each map
DATA = *matrices* or *pointers*	Data values for training each map
ERRORS = *matrices*	Reconstruction errors at the nodes of each map
FITROWS = *factors*	Save the positions of the rows allocated to the data points
FITCOLUMNS = *factors*	Save the positions of the columns allocated to the data points
Y = *variates*	Save y-values used to plot the data points
X = *variates*	Save x-values used to plot the data points
PEN = *scalars*, *variates* or *factors*	Pens used to plot the maps
SEED = *scalars*	Seed for the random numbers used to initialize the weights in each map

Description

A self-organizing map is a two dimensional grid of nodes, used to classify vectors of observations on *p* variables. Each node is characterized by a vector of *p* weights (one for each variable).

Before estimating the weights, you first need to declare a SOM structure to store the map. The SOM procedure, which does this, defines the row and column positions of the nodes on the grid. It also stores the names of the weight variables and information about how distances are to be measured on the grid and how the weights should be adjusted during their estimation. The SOM structure is then input to SOMESTIMATE by the SOM parameter.

The training dataset to estimate the weights is specified by the DATA parameter, either as a matrix with *n* rows and *p* columns (where *n* is the number of observations in the training set) or as a pointer containing *p* variates each with *n* units. SOMESTIMATE gives a warning if the row names of a DATA matrix or the names of the variates in a DATA pointer differ from the names

stored for the weight variables in the SOM structure.

The weights are estimated by a sequence of iterations, which are performed by the SOMADJUST procedure. In an iteration, the training observations are taken in turn. Each observation *i* is assessed to find its closest node. The method to use to measure distance on the map will have been specified, by the DMETHOD option of SOM, and stored with the SOM structure when it was declared. However, SOMESTIMATE also has a DMETHOD option in case you want to override the stored setting. The default setting for the DMETHOD option of SOM is euclidean. If X_i is a variate containing the values of the variables for observation *i* and W_j is the variate of weights at node *j*, the distance is then given by

```
d_ij = SQRT(SUM((X_i - W_j)**2))
```

The alternative setting, cityblock, calculates the distance as

```
d_ij = SUM(ABS(X_i - W_j)))
```

Once the closest node, *k*, has been found, the weights at that node and other nodes are adjusted. The method to use will have been specified when the SOM structure was declared, by the WMETHOD option of SOM. However, SOMESTIMATE again has its own WMETHOD option, that you can use to override the stored setting. The default setting for the DMETHOD option of SOM is gaussian. This adjusts the weights W_j at every node *j* to become

```
W_j + alpha * EXP( -0.5 * (d_jk / sigma)**2) * (X_i - W_j)
```

where d_jk is the distance between nodes *j* and *k*. With the alternative setting, neighbour, the weights at node *j* are adjusted to become

```
W_j + alpha * (X_i - W_j)
```

but only if d_jk is less than a threshold r.

The values of alpha, sigma and r change at each iteration. By default, SOMESTIMATE runs two sequences of iterations. At the start of the first set, the parameters have initial values

```
alpha = 1
sigma = dmax
r = dmax / 2
```

where dmax is the maximum distance between any two nodes in the network. At the end of the first set, they have final values

```
alpha = 0.1
sigma = dmax / 10
r = dmax / 10
```

There are 500 iterations in the first set, and the parameters decrease in equal steps from their initial to their final values. There are also 500 cycles in the second set of iterations, and the parameters now decrease in equal steps to to final values

```
alpha = 1
sigma = 0
r = dmin
```

where dmin is the minimum distance between any two nodes in the network. If dmax/10 is less than dmin, then the value of r at the end of the first set will be dmin too.

You can define your own sequence of iterations using the ALPHA, SIGMA, THRESHOLD and NCYCLE options (where SIGMA is relevant only when WMETHOD=gaussian, and THRESHOLD only when WMETHOD=neighbour). Setting all the relevant options to scalars, defines a single set of iterations where the parameters decrease from initial values set by the options to the final values specified above. Alternatively, you can set ALPHA and either SIGMA or THRESHOLD to variates to specify initial values for several sets of iterations. NCYCLE can be set to a scalar if all the sets are to contain the same number of iterations, or to a variate of the same length as ALPHA if you want each set to contain a different number.

The weights are initialized to have random positions within the plane of the first two principal components for the DATA matrix. The SEED parameter supplies a seed for the random numbers

used to define the positions. The default value of zero initializes the random number generator automatically if this is the first time that it has been used in the current job, or continues the existing sequence of random numbers.

By default SOMESTIMATE will stop the estimation process if there are more than ten successive iterations in which no observation changes its closest node. Different numbers of successive iterations with no changes can be specified using the NSTOP option.

Printed output is controlled by the PRINT option, with settings:

weights	to print the weights at each node of the map;
errors	to print the reconstruction errors at each node of the map;
monitoring	to provide monitoring about each iteration; and
report	to print a report at the end of the estimation process.

By default PRINT=weights,report.

The PLOT option controls which plots are produced, with settings:

fit	for a plot showing how the data observations are allocated to the nodes of the map; and
totalerror	for a plot showing how the total reconstruction error changes at each iteration.

By default, the map is plotted. The PEN parameter can be used to define the pen or pens to be used to plot the points on the map. If PEN is set to a scalar, the same pen will be used for every point, so you would simply be able to assess the density of points around the map. Alternatively, you can supply a variate or factor to distinguish different groups of observations.

The ERRORS parameter can save a matrix with the reconstruction error at each node of the map. The Y and X parameters can save the coordinates used to plot the points on the map. These are formed by adding a small amount of random variation to the row and column of the nodes, to ensure that points allocated to the same node are not all plotted in the same position.

Options: PRINT, PLOT, DMETHOD, WMETHOD, ALPHA, SIGMA, THRESHOLD, NCYCLE, NSTOP.
Parameters: SOM, DATA, ERRORS, FITROWS, FITCOLUMNS, Y, X, PEN, SEED.

Method

The individual iterations involved in the estimation are carried out by the SOMADJUST procedure.

Action with RESTRICT

SOMESTIMATE takes account of any restrictions defined on the DATA variates.

See also

Procedures: SOM, SOMADJUST, SOMDESCRIBE, SOMIDENTIFY, SOMPREDICT.
Genstat Reference Manual 1 Summary section on: Data mining.

SOMIDENTIFY

Allocates samples to nodes of a self-organizing map (R.W. Payne).

No options

Parameters

DATA = *matrices* or *pointers*	Data values used to allocate the samples to the nodes of the map
SOM = *pointers*	Save the information about each map
FITNODES = *factors*	Save nodes allocated to the data points
FITROWS = *factors*	Save the positions of the rows allocated to the data points
FITCOLUMNS = *factors*	Save the positions of the columns allocated to the data points

Description

A self-organizing map is a two dimensional grid of nodes, used to classify vectors of observations on *p* variables. Each node is characterized by a vector of *p* weights (one for each variable); these can be estimated, from a training dataset, by procedure SOMESTIMATE. This procedure, SOMIDENTIFY, allows you to allocate samples in a new dataset to the nodes of a map.

The new dataset is specified by the DATA parameter, either as a matrix with *n* rows and *p* columns (where *n* is the number of samples) or as a pointer containing *p* variates each with *n* units. The SOM parameter supplies the information about the self-organizing map, saved in a pointer using the SOM parameter of SOMESTIMATE. The FITNODES parameter saves a factor containing the number of the node to which each sample has been allocated. The FITROWS and FITCOLUMNS parameters save factors containing the row and column positions of the nodes.

Options: none.
Parameters: DATA, SOM, FITNODES, FITROWS, FITCOLUMNS.

Action with RESTRICT

SOMIDENTIFY takes account of any restrictions defined on the DATA variates.

See also

Procedures: SOM, SOMADJUST, SOMDESCRIBE, SOMESTIMATE, SOMPREDICT.
Genstat Reference Manual 1 Summary section on: Data mining.

SOMPREDICT

Makes predictions using a self-organizing map (R.W. Payne).

Options

PRINT = *string token*	Controls whether or not the predictions are printed (predictions); default pred
SOM = *pointer*	Specifies the map
YNAMES = *text*	Names of variables to predict; default * gives predictions for all the variables
METHODS = *string tokens*	Types of predictions to give (mean, mode, median, minimum, maximum, sd, variance); default mean, mode, medi, mini, maxi, sd, vari
YSAVE = *text*	Saves a text with a unit for each set of predictions giving the name of the corresponding y-variable
MSAVE = *text*	Saves a text with a unit for each set of predictions giving the name of the corresponding method

Parameters

DATA = *matrices* or *pointers*	Data values to identify the positions of the new samples on the map
UNITLABELS = *variates* or *texts*	Labels for the predictions (to identify the samples); default takes the row labels if DATA is a matrix or any unit labels if DATA is a pointer to a set of variates
PREDICTIONS = *variates* or *pointers*	Save the predictions

Description

A self-organizing map is a two dimensional grid of nodes, used to classify vectors of observations on *p* variables. Each node is characterized by a vector of *p* weights (one for each variable); these can be estimated, from a training dataset, by procedure SOMESTIMATE. You can then use procedure SOMDESCRIBE to associate, with each node, predictions of various types for a set of variables (and these variables need not be amongst those used to form the map). This information can now be used by SOMPREDICT to supply predictions for some new or hypothetical samples.

The SOM option supplies the information about the self-organizing map, which will have been saved in a pointer using the NEWSOM parameter of SOMDESCRIBE. The DATA parameter supplies the variables required to identify the positions of the new samples on the map, either as a matrix with *n* rows and *p* columns (where *n* is the number of samples) or as a pointer containing p variates each with *n* units. The SOMIDENTIFY procedure, called by SOMPREDICT to identify the positions, will issue a warning if the variables have different names to those in the data set used by SOMESTIMATE to form the map. The YNAMES option supplies a text containing the names of the variables for which predictions are required. (These correspond to the identifiers of the variates and/or factors specified by the Y parameter of SOMDESCRIBE to form the predicted values currently associated with the map.) If YNAMES is not set, predictions will be given for all those variables. If more than one type of prediction was requested for a Y variable, using the METHOD parameter of SOMDESCRIBE, you can use the METHODS option of SOMPREDICT to specify a list of strings to indicate which ones you want. By default all are given.

The PREDICTIONS parameter can save the predictions formed for each matrix or pointer supplied by the DATA parameter. If the YNAMES and METHODS options have requested several sets of predictions (for different variables and/or using different methods), PREDICTIONS will save a pointer containing a variate for each set. Alternatively if only one set has been requested (i.e

only one variable using only one method), PREDICTIONS will save a variate. To identify the variates within each pointer, the YSAVE option can save a text with a unit for each set of predictions, giving the name of the corresponding y-variable. Similarly, the MSAVE option can save a text whose units contain the names (in lower-case letters) of the corresponding methods. Each PREDICTIONS variate will have a unit for every sample. You can use the UNITLABELS parameter to supply a variate or text to label the units; otherwise SOMPREDICT uses the any row or unit labels defined on the matrix or variates supplied by the DATA parameter.

The PRINT option controls whether or not the predictions are printed (by default they will be printed).

Options: PRINT, SOM, YNAMES, METHODS, YSAVE, MSAVE.
Parameters: DATA, UNITLABELS, PREDICTIONS.

Method
The SOMIDENTIFY procedure is used to allocate the samples to the nodes of the map. The variates of predictions are then formed, from the information stored with the map, using ordinary Genstat declarations and calculations.

Action with RESTRICT
SOMPREDICT takes account of any restrictions defined on the variates in a DATA pointer.

See also
Procedures: SOM, SOMADJUST, SOMDESCRIBE, SOMESTIMATE, SOMIDENTIFY.
Genstat Reference Manual 1 Summary section on: Data mining.

SPCAPABILITY

Calculates capability statistics (R.W. Payne).

Option

PRINT = *string tokens*	Controls output (cpk, ppk, histogram); default cpk, ppk

Parameters

DATA = *variates* or *pointers*	Data measurements
SAMPLES = *factors* or *scalars*	Factor identifying samples or scalar indicating the size of each sample
LOWERLIMIT = *scalars*	Specifies the lower specification limit for each set of data
UPPERLIMIT = *scalars*	Specifies the upper specification limit for each set of data
CPK = *scalars*	Saves the index C_{pk}
PPK = *scalars*	Saves the index P_{pk}

Description

SPCAPABILITY calculates capability statistics. These are used to assess the extent to which the output of a process lies within its specification limits. The data values consist of samples of measurements made on successive occasions, which are specified by the DATA and SAMPLES parameters. DATA can be set to a variate containing the measurement and SAMPLES to a factor identifying the samples. Alternatively, if the samples are all of the same size and occur in the DATA variate one sample at a time, you can set SAMPLES to a scalar indicating the size of each sample. Finally, if the samples are in separate variates, you can set DATA to a pointer containing the variates (SAMPLES is then unset). The LOWERLIMIT parameter supplies the lower specification limit of the process, and the UPPERLIMIT parameter supplies the upper limit.

There are two indexes that can be calculated. The index C_{pk} is the minimum of the two quantities C_{pl} and C_{pu}. These are defined as

C_{pl} = (LOWERLIMIT - *mean*) / (3 × *sigma*)
C_{pu} = (UPPERLIMIT - *mean*) / (3 × *sigma*)

where *sigma* is the within-sample standard deviation (see for example Ryan 1989, Chapter 7). The alternative index, P_{pk}, is the minimum of the two quantities P_{pl} and P_{pu}. These have similar definitions to C_{pl} and C_{pu}, except that *sigma* now also includes the between-sample variation.

The PRINT option controls which of these are printed, with settings cpk and ppk. There is also a setting histogram, which plots a histogram of the data together with vertical lines indicating the lower and upper limits. By default PRINT=cpk,ppk. Alternatively, the indexes can be saved, in scalars, using the parameters CPK and PPK.

Option: PRINT.
Parameters: DATA, SAMPLES, LOWERLIMIT, UPPERLIMIT, CPK, PPK.

Method

SPCAPABILITY estimates the within-sample standard deviation (for C_{pk}) by the average of the standard deviations of the samples, each divided by a bias correction constant c_4:

$c_4 = \sqrt{(2/n)} \times \text{GAMMA}(n/2) / \text{GAMMA}((n-1)/2)$

where n is the sample size. Similarly the standard deviation for P_{pk} is the bias-corrected standard deviation of the samples, all pooled together.

Action with `RESTRICT`
Neither the `DATA` variates nor the `SAMPLE` factors may be restricted.

Reference
Ryan, T.P. (1989). *Statistical Methods for Quality Improvement*. Wiley, New York.

See also
Procedures: `SPCCHART`, `SPCUSUM`, `SPEWMA`, `SPPCHART`, `SPSHEWHART`.
Genstat Reference Manual 1 Summary section on: Six sigma.

SPCCHART

Plots *c* or *u* charts representing numbers of defective items (A.F. Kane & R.W. Payne).

Options

PRINT = *string token*	What to print (warnings); default * i.e. nothing
PLOT = *string token*	Type of chart to plot (c, u); default c
METHOD = *string token*	Method to use to obtain the control limits (given, loglinear, untransformed); default untr
TOLERANCEMULTIPLIER = *scalar*	Multiplier to use to test whether to use mean sample size for control limits; default 1
WINDOW = *scalar*	Which high-resolution graphics window to use; default 3
SCREEN = *string token*	Whether or not to clear the graphics screen before plotting (clear, keep); default clea

Parameters

NDEFECTIVE = *variates*	Number of defective items
NTESTED = *scalars* or *variates*	Number of items tested
CENTRELINE = *scalars*	Sets or saves centre line
LOWERCONTROLLIMIT = *scalars* or *variates*	Sets or saves lower control limit
UPPERCONTROLLIMIT = *scalars* or *variates*	Sets or saves upper control limit

Description

The *c* and *u* charts are used in statistical process control to evaluate testing schemes in which numbers of defects are measured in successive batches of items. The number of defects per batch is specified, in a variate, by the NDEFECTIVE parameter. The NTESTED parameter supplies the number of items in each batch – this can be a scalar if the batches are all of the same size, otherwise it is a variate.

The PLOT option controls the type of chart: the *c* chart plots number of defects per batch, while the *u* chart plots the number of defects per item.

The charts contain not only the observed numbers of defects but also a centre line (indicating a target value) and lines showing upper and lower control limits (bounding the zone outside which the process is said to be out of control). The control limits relevant to each batch in a *u* chart will depend on the batch sizes. The TOLERANCE option determines whether an average sample size is used if the individual sizes are not exactly equal: this will happen unless either

```
MIN(NTESTED) * TOLERANCE < MEAN(TESTED)
```

or

```
MEAN(TESTED) * TOLERANCE < MAX(NTESTED)
```

The METHOD option specifies how the various lines are to be defined, with the following settings.

untransformed	this is the default setting, and requests the method conventionally used in SPC. For a *c* chart, the centre line is at *c* = (total number defects) / (number of batches) and the limits are at $c \pm 3 \times \sqrt{(c)}$. For a *u* chart, the centre line is at *u* = (total number defects) / (total number of items) and the limits are at $u \pm 3 \times \sqrt{(u/n)}$.
given	specifies that the values are supplied by the CENTRELINE, LOWERCONTROLLIMIT and UPPERCONTROLLIMIT

parameters.

`loglinear` obtains the values by fitting a generalized linear model with Poisson distribution and log link.

For settings of `METHOD` other than `given`, the `CENTRELINE`, `LOWERCONTROLLIMIT` and `UPPERCONTROLLIMIT` parameters can be used to save the centre line and limits.

You can set `PRINT=warnings` to list any batches that are outside the control limits; by default these are suppressed. As usual, the `WINDOW` option specifies which high-resolution graphics window to use for the plot, and the `SCREEN` option controls whether or not to clear the graphics screen before plotting.

Options: `PRINT`, `PLOT`, `METHOD`, `TOLERANCEMULTIPLIER`, `WINDOW`, `SCREEN`.
Parameters: `NDEFECTIVE`, `NTESTED`, `CENTRELINE`, `LOWERCONTROLLIMIT`, `UPPERCONTROLLIMIT`.

Method
For further information about the standard SPC methods see for example Chapter 5 of Montgomery (1985). Section 3.5 of the *Guide to the Genstat Command Language, Part 2 Statistics* gives more details about generalized linear models.

Action with `RESTRICT`
Any restrictions are ignored.

Reference
Montgomery, D.C. (1985). *Introduction to Statistical Process Control*. Wiley, New York.

See also
Procedures: `SPCAPABILITY`, `SPCUSUM`, `SPEWMA`, `SPPCHART`, `SPSHEWHART`.
Genstat Reference Manual 1 Summary section on: Six sigma.

SPCOMBINE

Combines spreadsheet and data files, without reading them into Genstat (D.B. Baird).

Options

OUTFILENAME = *text*	Name of the output file
METHOD = *string token*	How to add the new data from the files specified by the FILENAME parameter (add, append, concatenate, merge); default appe
COLMATCH = *string token*	How to match columns when appending (name, position); default posi
GROUPS = *factor*	Factor to identify sections of appended files
OLDGLABEL = *texts*	Label to use in the GROUPS factor for the original data if GROUPS has not already been defined
MATCH = *text* or *pointer*	Up to four columns in the files specified by the FILENAME parameter to use as keys when merging files; default * uses the first column in the file
WITH = *text* or *pointer*	Columns in the OUTFILENAME file to use as keys when merging files; default * uses as many columns of the initial columns in OUTFILENAME as are needed to give a column for each MATCH column
UPDATE = *string token*	Whether to use columns with matching names to replace existing columns when concatenating or merging files (yes, no); default no changes the names of columns with the same name as existing columns so that they become unique
EXTRAROWS = *string token*	What to do with extra rows when merging files (all, matched, none); default all merges in all the extra rows into the data, matched merges in just the extra rows which have matching ids into the data, and none does not merge the extra rows into the data.

Parameters

FILENAME = *texts*	Names of files containing new data to be combined with the data in the OUTFILENAME file
SHEETNAME = *texts*	Name of a worksheet or a named range within an Excel, Quattro, 123 or Open Office spreadsheet file; default takes the first sheet
CELLRANGE = *texts*	Cell range giving the top left and bottom right cells within a worksheet; default takes all the data that it contains
ROWSELECTION = *variates*	Row numbers of the units of data to be included into the OUTFILENAME file; default takes all the rows
COLSELECTION = *variates*	Numbers of the columns of data to be combined with the OUTFILENAME file; default takes all the columns
PAGENAME = *texts*	Page name for each new sheet when METHOD=add; default 'SHEET<n>' where n is the number of the sheet in the OUTFILENAME file, unless the sheet is already named in the FILENAME file
GLABEL = *texts*	Label to use in the GROUPS factor to identify the data from each FILENAME file; if this is unset, GROUPS is defined with only levels

Description

SPCOMBINE combines spreadsheet files into a single file, without reading all the data into Genstat. This is intended for use in particular with very large data sets.

The names of the files containing the new data values to be combined with an original dataset are specified using the FILENAME parameter. The file name can also be an internet URL prefixed with http://, https:// ftp:// or file://, in which case the data source is downloaded and then imported. The name of the output file is specified by the OUTFILENAME option. This may already contain a set of data. Alternatively, if it does not exist already, it will be created.

The following file types are supported for input: Excel 2-5, 95, 97, 2000, XP, 2003, 2007, Open Office, Lotus WK1, Quattro (WQ1, WB*, QPW), dBase 2-5, Paradox 3-9, Genstat GSH and GWB, SAS PC 6.03-12, 7-9, SAS Transport, SAS JMP, Minitab 8-14, Statistica 5 and 6, Systat, MStat, Instat, Epi-Info, SPSS/Win, Gauss Data/Matrix (PC/Win/Unix), MatLab, S+ (PC/Unix), Stata 4-8, StatGraphics, R data frames, Weka Attribute files, SigmaPlot 7-9, OSIRIS, Limdep, RATS, EViews, GRETL panel files, Comma delimited text files (*.CSV), Cornell Ecology format, MapQTL trait files (.QUA), ArcView/Info Shapefiles, MapInfo Exchange files, Windows Bitmap (*.BMP), Windows Sound (*.WAV), NMR Binary files and image files (JPG, GIF, TIF, PNG). The file type is worked out from the file contents, so the usual extension need not be used with the exception of the following file types which do not contain a unique signature: Epi-Info (.REC), S+ (.SDD) and Paradox (.DB). Any files not containing a unique file signature, but ending in these extensions will be classified as above. Any other file extensions will be attempted to be read as a comma, space or tab delimited text file. A subset of these file types is supported for output in the OUTFILENAME file: Genstat, Excel, Open Office, R, dBase, Lotus, Weka, SAS Transport, CSV and TXT. If the OUTFILENAME file does not exist, its format is determined by its extension.

You can use the SHEETNAME and CELLRANGE parameters to define a specified section of data to take from a spreadsheet file (Excel, Quattro, 123, Open Office). In addition, the ROWSELECTION and COLSELECTION parameters can specify that a subset of the rows or columns, respectively, is to be included. They can be set to a variate containing the numbers of the rows or columns. With COLSELECTION, you can also supply a text containing column names. So, for example, to import only rows where the variate Year is greater than 2005, you could put

```
ROWSELECTION = WHERE(Year > 2005)
```

(the WHERE function gives the unit numbers where a logical expression has the value one i.e. true). Note that the variate Year must already have been imported into Genstat, in order to do the calculation – this can be done using the IMPORT procedure.

The METHOD option controls how the files are combined. The add setting can be used to include the new data from each FILENAME file as a new sheet in the OUTFILENAME file, provided this is an Excel or Genstat GWB file. The PAGENAME parameter specifies the name to use for the page of the added sheet. If this is not set, SPCOMBINE looks to see whether the sheet is already named in the FILENAME file. If so, it will use that name, adding a number at the end, if necessary, to make the name unique. Otherwise, the name will be 'SHEET<n>' where n is the number of the sheet in the OUTFILENAME file.

The append setting of METHOD appends the new data values at the end of those in the OUTFILENAME file. The COLMATCH option specifies whether the columns are matched by name or position. If a matching column is not found, a new empty column is created and the data are appended to that column. The append will fail if the types of the original and appended data do not match (e.g. if you attempt to append a text onto a variate). The GROUPS option can specify a factor to indicate the source of the data. If the factor is not already present in the OUTFILENAME file, the OLDGLABEL option can supply a label to use to identify the original rows of data. The GLABEL parameter can specify the label to use for the new rows appended from the FILENAME file. If these are not supplied, the factor will have only levels.

If METHOD=concatenate, the new data from the FILENAME file are added as new columns on the right-hand side of the sheet in the OUTFILENAME file. However, the types of the sheets (vectors, matrix or scalar) and their lengths must match. The new data can also be added as new columns on the right-hand side of the OUTFILENAME file by setting METHOD=merge. The rows of data from the FILENAME file are now merged with the original rows using up to four key columns specified by the MATCH and WITH options (for the new and original rows, respectively). If MATCH is not specified, the first column of new data is used. If WITH is not specified, the MATCH columns are matched with the same number of initial columns of the OUTFILENAME file. If a column with the same name already exists in the OUTFILENAME file when concatenating or merging, the default action is to rename the new column by adding a number to the end of the name to make it unique. Alternatively, if you set option UPDATE=yes, the new column will replace the existing column. When METHOD=merge, the EXTRAROWS option controls what happens to rows in the data being merged that do not appear already in the OUTFILENAME file: these rows can be added (all), added only if the id is already in the file (matched) or omitted (none).

Options: OUTFILENAME, METHOD, COLMATCH, GROUPS, OLDGLABEL, MATCH, WITH, UPDATE, EXTRAROWS.

Parameters: FILENAME, SHEETNAME, CELLRANGE, ROWSELECTION, COLSELECTION, PAGENAME, GLABEL.

Method

The data files are combined by passing a request to the Dataload.dll library. This reads the data from OUTFILENAME and each FILENAME, includes the new data, and then rewrites OUTFILENAME.

Action with RESTRICT

When METHOD=concatenate, restrictions from FILENAME are added to the new rows of OUTFILENAME. When METHOD=add, restrictions from FILENAME are retained on the new pages in OUTFILENAME. Otherwise restrictions are ignored.

See also

Directive: SPLOAD.

Procedure: IMPORT.

Genstat Reference Manual 1 Summary section on: Input and output.

SPCUSUM

Prints CUSUM tables for controlling a process mean (A.F. Kane & R.W. Payne).

Options

REFERENCEVALUE = *scalars*	Specifies the upper and then the lower reference values, or just one of these if they are both the same; default 0.5
THRESHOLD = *scalars*	Detection thresholds, upper and then the lower, or just one of these if they are both the same; default 5
HEADSTART = *scalars*	Headstart values, upper and then the lower, or just one of these if they are both the same; default 0

Parameters

DATA = *variates* or *pointers*	Data measurements
SAMPLES = *factors* or *scalars*	Factor identifying samples or scalar indicating the size of each sample
MEANTARGET = *scalars*	Specifies the target value for the sample means
SIGMA = *scalars*	Specifies or saves the standard deviation of the observations

Description

SPCUSUM prints cumulative sum (or *CUSUM*) charts, as described for example by Ryan (1989). The data values consist of samples of measurements made on successive occasions, which are specified by the DATA and SAMPLES parameters. DATA can be set to a variate containing the measurement and SAMPLES to a factor identifying the samples. Alternatively, if the samples are all of the same size and occur in the DATA variate one sample at a time, you can set SAMPLES to a scalar indicating the size of each sample. Finally, if the samples are in separate variates, you can set DATA to a pointer containing the variates (SAMPLES is then unset).

The chart displays columns containing:

1) the sample number;
2) the sample mean;
3) z, the devation of the mean from a target value, divided by its standard deviation;
4) SH, the upper *CUSUM*;
5) SL, the lower *CUSUM*.

An asterisk is printed alongside any values SH and SL that exceed a threshold value, indicating that the process is out of control.

The *CUSUM* values SH_i and SL_i for each sample i are calculated as

$$SH_i = z_i - k_u + SH_{i-1}$$
or $$= 0 \quad \text{if } z_i - k_u + SH_{i-1} < 0$$
$$SL_i = -z_i - k_l + SL_{i-1}$$
or $$= 0 \quad \text{if } -z_i - k_l + SL_{i-1} < 0$$

The target value is specified by the MEANTARGET parameter. The SIGMA parameter can be used to specify the standard deviation of the individual observations (which is required to calculate the standard deviation of the deviations of the sample means from the target value). It this is not set or if it is set to a missing value, the standard deviation is calculated using the within-sample replication, as the average of the standard deviations of the samples, divided by a bias correction constant c_4:

$$c_4 = \sqrt{(2/n)} \times \text{GAMMA}(n/2) \,/\, \text{GAMMA}((n-1)/2)$$

where n is the sample size. You can thus save the calculated standard deviation by setting SIGMA to a scalar containing a missing value.

The *reference values* k_u and k_l are specified by the REFERENCEVALUE option. If they are both the same, you need specify this only once. Their default is 0.5. Similarly the threshold value, or

values, are specified by the THRESHOLD option; by default these take the value 5. The *CUSUM*s usually start at 0, but you can specify another value or values using the HEADSTART option.

Options: REFERENCEVALUE, THRESHOLD, HEADSTART.
Parameters: DATA, SAMPLES, MEANTARGET, SIGMA.

Reference
For further details of the method, and advice on the setting of thresholds, reference values and so on, see for example Ryan (1989) Section 5.3.

Action with RESTRICT
Neither the DATA variates nor the SAMPLE factors may be restricted.

Reference
Ryan, T.P. (1989). *Statistical Methods for Quality Improvement*. Wiley, New York.

See also
Procedures: SPCAPABILITY, SPCCHART, SPEWMA, SPPCHART, SPSHEWHART.
Genstat Reference Manual 1 Summary section on: Six sigma.

SPEARMAN

Calculates Spearman's rank correlation coefficient (S.J. Welham, N.M. Maclaren & H.R. Simpson).

Options

PRINT = *string tokens*	Output required (test, correlations, ranks): test produces the correlation coefficient/matrix and relevant test statistics, correlations prints out just the correlation coefficients for each pair of variates; ranks produces the vectors of ranks for each sample; default test
GROUPS = *factor*	Defines the sample membership if only one variate is specified by DATA
CORRELATION = *scalar* or *symmetric matrix*	Scalar to save the rank correlation coefficient if there are two samples, or symmetric matrix to save the coefficients between all pairs of samples if there are several
T = *scalar* or *symmetric matrix*	Scalar to save the Student's t approximation to the correlation coefficient if there are two samples, or symmetric matrix to save the t approximations for all pairs of samples if there are several (calculated only if the sample size is 8 or more)
DF = *scalars*	Scalar to save the degrees of freedom for each t-statistic

Parameters

DATA = *variates*	List of variates containing the data for each sample, or a single variate containing the data from all the samples (the GROUPS option must then be set to indicate the sample to which each unit belongs)
RANKS = *variates*	Saves the ranks

Description

SPEARMAN calculates Spearman's rank correlation coefficient between pairs of samples. The samples can be stored in different variates and supplied in a list with the DATA parameter. Alternatively, they can all be placed in a single variate, and the GROUPS option set to a factor to indicate the sample to which each unit belongs.

If the sample size is less than 20, an exact two-sided probability is calculated using the PRSPEARMAN procedure. Note, though, that the probability will be approximate if the variates contain ties; the probability is calculated for the adjusted correlation, but the calculation itself takes no account of the ties. SPEARMAN also calculates a Student's t approximation if the sample size is 8 or more (i.e. large enough for the approximation to be valid).

Printed output is controlled by the PRINT option, with settings:

correlation	to display correlations;
test	to display tests and correlations; and
ranks	to display the ranks for each sample.

The results can also be saved using the CORRELATION, T and DF options and the RANKS parameter.

Options: PRINT, GROUPS, CORRELATION, T, DF.
Parameters: DATA, RANKS.

Method

Spearman's rank correlation coefficient is a measure of association between the rankings of two variables measured on N individuals (i.e. two vectors of length N). The correlation coefficient is calculated from the two vectors of ranks for the samples: let $\{ X_i ; i=1...N \}$ and $\{ Y_i ; i=1...N \}$ be the vectors of ranks for sample 1 and sample 2 respectively, then the coefficient r is based on the vector of differences between ranks: $\{ D_i = X_i - Y_i ; i=1...N \}$ and is calculated by

$$r = 1 - 6 \times \sum_{i=1...N} D_i^2 / [N(N^2-1)].$$

If ties are present, then the statistic will be biased, and must be recalculated taking account of ties by:

$$r = (\sum X_i^2 + \sum Y_i^2 - \sum D_i^2) / (2 \times \sqrt{(\sum X_i^2 \times \sum Y_i^2)})$$

where $\sum X_i^2 = (N^3-N)/12 - T_x$;

$\sum Y_i^2 = (N^3-N)/12 - T_y$;

$T_k = \sum (t_j^3 - t_j)/12$

and t_j is the number of observations in the group with rank j.

The t-approximation for this statistic, T, is valid for samples of size 8 upwards, and is calculated by

$$T = r \times \sqrt{[(N-2)/(1-r^2)]}.$$

It has approximately a t-distribution on $N-2$ degrees of freedom, and can be used for a test of the null hypothesis of independance between samples. (See for example Siegel 1956, pages 202-213, or Siegel & Castellan 1988, pages 235-244.)

Action with `RESTRICT`

If any of the variates in `DATA` is restricted, the statistic is calculated only for the set of units not excluded by the restriction.

References

Siegel, S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

Siegel, S. & Castellan, N.J. (1988). *Nonparametric Statictics for the Behavioural Sciences (second edition)*. McGraw-Hill, New York.

See also

Procedures: `PRSPEARMAN`, `CMHTEST`, `FCORRELATION`, `KCONCORDANCE`, `KTAU`, `LCONCORDANCE`.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

SPEWMA

Plots exponentially weighted moving-average control charts (A.F. Kane & R.W. Payne).

Options

PRINT = *string token*	What to print (warnings); default * i.e. nothing
TOLERANCEMULTIPLIER = *scalar*	Multiplier to use to test whether to use mean sample size for control limits; default 1
WEIGHT = *scalar*	Weight parameter used in the calculation of the exponentially weighted moving-average statistic; default 0.25
NSIGMA = *scalar*	Number of multiples of sigma to use for control limits; default 3
WINDOW = *scalar*	Which high-resolution graphics window to use; default 3
SCREEN = *string token*	Whether or not to clear the graphics screen before plotting (clear, keep); default clea

Parameters

DATA = *variates* or *pointers*	Data measurements
SAMPLES = *factors* or *scalars*	Factor identifying samples or scalar indicating the size of each sample
MEAN = *scalars*	Sets or saves the sample mean value
SIGMA = *scalars*	Sets or saves the sample standard deviation

Description

SPEWMA plots exponentially weighted moving-average control charts for controlling the mean of a process. The data values consist of samples of measurements made on successive occasions, which are specified by the DATA and SAMPLES parameters. DATA can be set to a variate containing the measurement and SAMPLES to a factor identifying the samples. Alternatively, if the samples are all of the same size and occur in the DATA variate one sample at a time, you can set SAMPLES to a scalar indicating the size of each sample. Finally, if the samples are in separate variates, you can set DATA to a pointer containing the variates (SAMPLES is then unset).

The chart plots a statistic w whose value for sample t is a weighted average of the mean of sample t, and the value of the statistic for sample $t-1$:

$$w_t = r_t \times xbar_t + (1 - r) \times w_{t-1}$$

where *xbar* is the variate of sample means, and r is the weighting parameter specified by the WEIGHT option of the procedure with default 0.25. (Notice that the statistic involves all the previous means, but with exponentially decreasing weights.)

The position of the central line for the chart is specified, in a scalar, by the MEAN parameter. If this is not set, or if it is set to a scalar containing a missing value, the overall mean of the samples is used. (So you can save the calculated mean by setting MEAN to a scalar containing a missing value.) There are also control lines $-nsigma \times \text{var}(w)$ and $+nsigma \times \text{var}(w)$, where *nsigma* is specified by the NSIGMA option (default 3) and var(w) is the variance of the statistic w. For sample t, this is

$$(3 \times sigma / \sqrt{(\text{REP}_t)}) \times \sqrt{((r/(2 - r)) \times (1 - (1 - r)^{2t}))}$$

where REP is a variate containing the number of observations in each sample, and *sigma* is the standard deviation of a single observation. The SIGMA parameter can be used to supply a value for *sigma*. It this is not set or if it is set to a missing value, *sigma* is calculated using the within-sample replication as the average of the standard deviations of the samples, divided by a bias correction constant c_4:

$$c_4 = \sqrt{(2/n)} \times \text{GAMMA}(n/2) / \text{GAMMA}((n-1)/2)$$

The TOLERANCE option determines whether an average replication is used if the replication

of the individual samples is no exactly equal: this will happen unless either

```
MIN(REP) * TOLERANCE < MEAN(rep)
```

or

```
MEAN(rep) * TOLERANCE < MAX(rep)
```

You can set `PRINT=warnings` to list any batches that are outside the control limits; by default these are suppressed. As usual, the `WINDOWS` option specifies which high-resolution graphics window to use for the plot (default 3), and the `SCREEN` option controls whether or not to clear the graphics screen before plotting the charts.

Options: `PRINT`, `TOLERANCEMULTIPLIER`, `WEIGHT`, `NSIGMA`, `WINDOW`, `SCREEN`.
Parameters: `DATA`, `SAMPLES`, `MEAN`, `SIGMA`.

Method
Further details of the method, and advice on the setting of the weight parameter, can be found for example in Ryan (1989) Section 5.5.

Action with `RESTRICT`
Neither the `DATA` variates nor the `SAMPLE` factors may be restricted.

Reference
Ryan, T.P. (1989). *Statistical Methods for Quality Improvement*. Wiley, New York.

See also
Procedures: `SPCAPABILITY`, `SPCCHART`, `SPCUSUM`, `SPPCHART`, `SPSHEWHART`.
Genstat Reference Manual 1 Summary section on: Six sigma.

SPLINE

Calculates a set of basis functions for M-, B- or I-splines (P.W. Goedhart).

Options

KNOTS = *scalar* or *variate*	Defines the interior knot values; no default i.e. this option must be set
ORDER = *scalar*	Defines the order of the piecewise polynomial; default 3
TYPE = *string token*	Controls which spline basis is calculated (m, b, i); default m
LOWER = *scalar*	Left-hand limit *L* of the interval [*L*, *U*); default * i.e. the minimum of the X parameter is used
UPPER = *scalar*	Right-hand limit *U* of the interval [*L*, *U*); default * i.e. a value slightly larger than the maximum of the X parameter is used
NOMESSAGE = *string token*	Which warning messages to suppress (warning); default *

Parameters

X = *variates*	Values for which the basis spline functions are calculated
BASIS = *pointers*	Pointer to save variates containing the values of the basis spline functions
DBASIS = *pointers*	Pointer to save variates containing the values of the first order derivatives of the basis spline functions

Description

Piecewise polynomials or splines can be used for nonparametric function estimation. Splines offer a flexible way to investigate the shape of a relationship or can be used for interpolation and smoothing. There are several types of splines. Smoothing splines, implemented in Genstat by means of regression function SSPLINE, minimize a penalized residual sums of squares in which lack of smoothness of the estimated function is penalized. Smoothing splines can be less appropriate when local effects are strong or when the estimated function should be monotone, e.g. when estimating growth curves.

An alternative for smoothing splines is to use regression splines which offer more control over the characteristics of the estimated function. With regression splines the user first specifies an interval [*L*, *U*) on which the estimated function is non-trivial. This interval is then explicitly divided into segments by the user, and a polynomial, of order say *k*, is fitted in each segment. The segments are separated by a sequence of so-called knots. It is customary to force the piecewise polynomials to join smoothly at these knots. The piecewise polynomials and all their derivatives are always continuous from the right at the knots. Moreover, when there are no replicated knot values, the $(k-1)$th derivative is continuous at the knot values. The order of differentiability is lower when there are replicated knot values. The full knot sequence includes the endpoints *L* and *U* which are replicated depending on the order of the piecewise polynomial. Ramsay (1988) provides a concise introduction into regression splines, while de Boor (1978) gives a full account.

The SPLINE procedure can be used to calculate a set of so called basis functions which have all the required properties of continuity and differentiability. These basis functions can then be used to fit the regression spline. A simple basis is given by truncated polynomials but this has the disadvantage of generating considerable rounding errors. A numerically superior basis is provided by M-splines. Their main features are that any basis function is positive in a series of consecutive segments, is zero elsewhere and is normalized by having unit area. An alternative

normalization is provided by B-splines which have the property that the sum over all basis functions is 1 for values in the interval [*L*, *U*). Basis functions of M-splines and B-splines are linearly related and are 0 outside [*L*, *U*). The resulting piecewise polynomial is discontinuous at the endpoints *L* and *U*.

Monotonicity of the estimating function can be imposed by employing a basis consisting of monotone functions. Ramsay (1988) uses integrated M-splines which, when combined with nonnegative regression coefficients, yield a monotone spline. These integrated M-splines are called I-splines. The basis functions for I-splines are not linearly related and they are 0 for values smaller than *L* and 1 for values greater than or equal to *U*. The resulting piecewise polynomial is continuous but not differentiable at the endpoints. The choice of the polynomial order and of the knot values are crucial for successful usage of regression splines. Wegman & Wright (1983) summarize practical recommendations for M-splines, while Ramsay (1988) does so for I-splines. In general the knots should be chosen in regions where the relationship changes most markedly. A useful preliminary knot placement is to position a single interior knot at the median, two interior knots at the terciles, three at the quartiles, and so on. The order of the piecewice polynomials is usually taken to be 2 or 3.

The values for which the basis functions are calculated must be specified by the `X` parameter. The values of the basis functions are saved with the `BASIS` parameter, while the first order derivatives of the basis functions can be saved by setting the `DBASIS` parameter. The `BASIS` and `DBASIS` pointers are redefined in the procedure. If a value in the `X` parameter coincides with an interior knot and the basis function or its first order derivative has a discontinuity at that value, it should be remembered that the functions are continuous and differentiable from the right.

The interior knot sequence must be set with the `KNOTS` option and the `ORDER` option can be used to specify the order of the piecewise polynomials. The `TYPE` option determines which spline basis is calculated. The interval [*L*, *U*) for which the basis functions are non-trivial can be specified by the `LOWER` and `UPPER` options. If these are unset the following values are used:

```
CALCULATE LOWER = MINIMUM(X)
CALCULATE max   = MAXIMUM(X)
CALCULATE UPPER = max + ((max.EQ.0) + ABS(max))/500000
```

In this case the `UPPER` value is such that max is just in the interval [*L*, *U*). The `NOMESSAGE` option can be used to suppress warning messages which are printed when the `KNOTS` variate has replicated values and when the interval [*L*, *U*) does not overlap the range of `X` values.

Options: `KNOTS`, `ORDER`, `TYPE`, `LOWER`, `UPPER`, `NOMESSAGE`.
Parameters: `X`, `BASIS`, `DBASIS`.

Method

Basis functions for M-splines are calculated by a recurrence relation from Ramsay (1988). These basis functions are multiplied to give B-splines or summed to provide I-splines. Note that unlike Ramsay (1988), the order of the spline is here defined as the order of the piecewise polynomial.

Action with `RESTRICT`

The variates contained in the `BASIS` and `DBASIS` pointers are restricted in the same way as the `X` parameter. Values in the units excluded by the restriction are set to missing. Restrictions on the `KNOTS` variate are ignored.

References

de Boor, C. (1978). *A Practical Guide to Splines*. Springer-Verlag. New York.

Ramsay, J.O. (1988). Monotone regression splines in action (with discussion). *Statistical Science*, **3**, 425-441.

Wegman, E.J. & Wright, I.W. (1983). Splines in statistics. *Journal of the American Statistical*

Association, **78**, 351-365.

See also
Directive: VCOMPONENTS.
Procedures: SPLINE, NCSPLINE, PENSPLINE, PSPLINE, RADIALSPLINE, TENSORSPLINE.
Function: SSPLINE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

SPNTEST

Calculates the sample size for a Poisson test (R.W. Payne & D.A. Murray).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
PRMETHOD = *string token*	Method to be used to calculate the probabilities for the test (normalapproximation, exact); default norm
PROBABILITY = *scalar*	Significance level for the test; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Type of test to be done (onesided, twosided); default ones
NULL = *scalar*	Mean under the null hypothesis for the one-sample test; must be set when MU2 is unset
RATIOREPLICATION = *scalar*	Ratio of replication sample2:sample1 (i.e. the size of sample 2 should be RATIOREPLICATION times the size of sample 1); default 1
REPLICATION = *variate*	Replication values for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

MU1 = *scalars*	Mean to detect in sample 1
MU2 = *scalars*	Mean to detect in sample 2
NREPLICATES = *scalars*	Saves the required number of replicates
VREPLICATION = *variates*	Numbers of replicates for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

SPNTEST calculates the number of replicates (or sample size) required for a Poisson test. In the one-sample Poisson test, the data consist of a set of counts that are assumed to have been generated by the same Poisson distribution, and the sample size is the number of counts that have been observed. The mean that needs to be detected is specified by the MU1 parameter, and the value from which it needs to be distinguished (i.e. the value under the null hypothesis) is specified by the NULL option.

Alternatively, a two-sample test assesses the evidence that the there is a difference between the means of the Poisson distributions that have generated two separate samples of counts. The anticipated mean for the first sample is then specified by the MU1 parameter, and the mean for the second sample is specified by the MU2 parameter.

The PRMETHOD option defines the type of Poisson test that is to be done. The normalapproximation indicates that the test will be based on the Normal approximation to the Poisson distribution. The exact setting, which is available only for the one-sample test, does an exact test using the Poisson distribution. See the PNTEST procedure for more information.

The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%). The required probability for detection of the difference between the means (that is, the *power* of the test) is specified by the POWER option (default 0.9).

It is generally assumed that the sizes of the samples in the two-sample test should be equal. However, you can set the RATIOREPLICATION option to a scalar, R say, to indicate that the size

of the second sample should be R times the size of the first sample.

By default, SPNTEST assumes a one-sided test is to be used, but you can set option TMETHOD=twosided to take a two-sided test instead.

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates in each sample (i.e. the size of each sample);
power	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The NREPLICATES parameter allows you to save the required size of the first sample. The replications and powers in the table can also be saved, in variates, using the VREPLICATION and VPOWER parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, PRMETHOD, PROBABILITY, POWER, TMETHOD, NULL, RATIOREPLICATION, REPLICATION.
Parameters: MU1, MU2, NREPLICATES, VREPLICATION, VPOWER.

See also
Procedure: PNTEST.
Genstat Reference Manual 1 Summary section on: Design of experiments.

SPPCHART

Plots *p* or *np* charts for binomial testing for defective items (A.F. Kane & R.W. Payne).

Options

PRINT = *string token*	What to print (warnings); default * i.e. nothing
PLOT = *string token*	Type of chart to plot (p, np); default p
METHOD = *string token*	Method to use to obtain the control limits (complementaryloglog, given, logit, probit, untransformed); default untr
TOLERANCEMULTIPLIER = *scalar*	Multiplier to use to test whether to use mean sample size for control limits; default 1
WINDOW = *scalar*	Which high-resolution graphics window to use; default 3
SCREEN = *string token*	Whether or not to clear the graphics screen before plotting (clear, keep); default clea

Parameters

NDEFECTIVE = *variates*	Number of defective items
NTESTED = *scalars* or *variates*	Number of items tested
CENTRELINE = *scalars*	Sets or saves centre line
LOWERCONTROLLIMIT = *scalars* or *variates*	Sets or saves lower control limit
UPPERCONTROLLIMIT = *scalars* or *variates*	Sets or saves upper control limit

Description

The *p* and *np* charts are used in statistical process control to evaluate testing schemes in which successive batches of items are classified as either good or defective. The number of defective items in each batch is specified, in a variate, by the NDEFECTIVE parameter. The NTESTED parameter supplies the number of items in each batch – this can be a scalar if the batches are all of the same size, otherwise it is a variate.

The PLOT option controls the type of chart: the *p* chart plots the proportion of defective items while the *np* chart (which is most useful each batch of items has the same total size) plots the number of defective items.

The charts contain not only the observed numbers or proportions but also a centre line (indicating a target value) and lines showing upper and lower control limits (bounding the zone outside which the process is said to be out of control). The control limits relevant to each batch will depend on the batch sizes. The TOLERANCE option determines whether an average total size is used if the individual totals are not exactly equal: this will happen unless either

```
MIN(NTESTED) * TOLERANCE < MEAN(TESTED)
```

or

```
MEAN(TESTED) * TOLERANCE < MAX(NTESTED)
```

The METHOD option specifies how the various lines are to be defined, with the following settings. They are defined below for a *p* chart. For an *np* chart, the values are simple multiplied by the batch size(s).

untransformed	this is the default setting, and requests the method conventionally used in SPC. The centre line is at p = (total number defective) / (total number tested) and the limits are at $p \pm 3 \times \sqrt{(p / (1-p))}$
given	specifies that the values are supplied by the CENTRELINE, LOWERCONTROLLIMIT and UPPERCONTROLLIMIT

parameters.

`logit`	obtains the values as the batch mean +/− three times its standard error as estimated on the logit scale of a generalized linear model (with binomial distribution).
`probit`	obtains the values as the batch mean +/− three times its standard error as estimated on the probit scale of a generalized linear model
`complementaryloglog`	obtains the values as the batch mean +/− three times its standard error as estimated on the complementary-log-log scale of a generalized linear model.

For settings of `METHOD` other than `given`, the `CENTRELINE`, `LOWERCONTROLLIMIT` and `UPPERCONTROLLIMIT` parameters can be used to save the centre line and limits.

You can set `PRINT=warnings` to list any batches that are outside the control limits; by default these are suppressed. As usual, the `WINDOW` option specifies which high-resolution graphics window to use for the plot, and the `SCREEN` option controls whether or not to clear the graphics screen before plotting.

Options: `PRINT`, `PLOT`, `METHOD`, `TOLERANCEMULTIPLIER`, `WINDOW`, `SCREEN`.
Parameters: `NDEFECTIVE`, `NTESTED`, `CENTRELINE`, `LOWERCONTROLLIMIT`, `UPPERCONTROLLIMIT`.

Method
For further information about the standard SPC methods see for example Chapter 5 of Montgomery (1985). Section 3.5 of the *Guide to the Genstat Command Language, Part 2 Statistics* gives more details about generalized linear models.

Action with `RESTRICT`
Any restrictions are ignored.

Reference
Montgomery, D.C. (1985). *Introduction to Statistical Process Control*. Wiley, New York.

See also
Procedures: `SPCAPABILITY`, `SPCCHART`, `SPCUSUM`, `SPEWMA`, `SPSHEWHART`.
Genstat Reference Manual 1 Summary section on: Six sigma.

SPRECISION

Calculates the sample size to obtain a specified precision (R.W. Payne).

Options

PRINT = *string token*	What to print (replication, precision); default repl, prec
NSAMPLES = *scalar*	Number of samples (1 or 2); default 2
CIPROBABILITY = *scalar*	Probability level for the confidence interval to indicate the precision; default 0.95
RATIOREPLICATION = *scalar*	Ratio of replication sample2:sample1 (i.e. the size of sample 2 should have be RATIOREPLICATION times the size of sample 1); default 1
REPLICATION = *variate*	Replication values for which to calculate and print or save the precision; default * takes 11 replication values centred around the required number of replicates

Parameters

PRECISION = *scalars*	Required precision
VAR1 = *scalars*	Anticipated variance of sample 1
VAR2 = *scalars*	Anticipated variance of sample 2; default * assumes the same variance as sample 1
NREPLICATES = *scalars*	Saves the required number of replicates
VREPLICATION = *variates*	Numbers of replicates for which precisions have been calculated
VPRDETECTION = *variates*	Precision for the various numbers of replicates

Description

SPRECISION calculates the number of replicates (or sample size) required to estimate a sample mean, or the difference between the means of two samples to a specified precision. The number of samples is specified by the NSAMPLES option (default 2). The precision is obtained by calculating a confidence interval around the sample mean or difference of means, and represents half the width of the interval. (The interval is generated by a t distribution, so this represents the distance of the mean or difference between means and the lower or the upper limits of the interval.) The probability level for the interval is specified by the CIPROBABILITY option (default 0.95 i.e. 95%).

The required precision is supplied by the PRECISION parameter. The variances of the first and second samples are supplied by the VAR1 and VAR2 parameters. VAR2 can be omitted if there is only one sample, or the two samples have equal variances. It is generally assumed that the second sample (if present) should be the same size as the first sample. However, you can set the RATIOREPLICATION option to a scalar, R say, to indicate that the size of the second sample should be R times the size of the first sample. The NREPLICATES parameter allows you to save the required size of the first sample.

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates in each sample (i.e. the size of each sample);
precision	to print a table giving the precision provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding detection probabilities in the table can also be saved, in variates, using the VREPLICATION and VPRDETECTION parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the probabilities of

detection; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, NSAMPLES, CIPROBABILITY, RATIOREPLICATION, REPLICATION.
Parameters: PRECISION, VAR1, VAR2, NREPLICATES, VREPLICATION, VPRECISION.

Method
An approximate number of replicates is calculated initially assuming a Normal approximation. This is then refined by calculating powers for a range of replications centred around that approximation.

See also
Procedures: ADETECTION, ASAMPLESIZE.
Genstat Reference Manual 1 Summary section on: Design of experiments.

SPSHEWHART

Plots control charts for mean and standard deviation or range (A.F. Kane & R.W. Payne).

Options

PRINT = *string token*	What to print (warnings); default * i.e. nothing
PLOT = *string token*	Type of chart to plot to accompany the chart of sample means (range, standarddeviation); default stan
METHOD = *string token*	Type of control limits (probability, sigma); default sigm
TOLERANCEMULTIPLIER = *scalar*	Multiplier to use to test whether to use mean sample size for control limits; default 1
PROBABILITY = *scalars*	Probability value(s) to use to calculate control limits when METHOD=probability; default 0.01, 0.025
WINDOWS = *scalar*	Which high-resolution graphics windows to use; if unset SPSHEWHART automatically sets up two windows containing the upper and lower halves of the screen
SCREEN = *string token*	Whether or not to clear the graphics screen before plotting (clear, keep); default clea

Parameters

DATA = *variates* or *pointers*	Data measurements
SAMPLES = *factors* or *scalars*	Factor identifying samples or scalar indicating the size of each sample
MEAN = *scalars*	Sets or saves the sample mean value
SIGMA = *scalars*	Sets or saves the sample standard deviation

Description

SPSHEWHART plots the standard charts devised by Shewhart (1931) for the control of manufacturing processes. The data values consist of samples of measurements made on successive occasions, which are specified by the DATA and SAMPLES parameters. DATA can be set to a variate containing the measurement and SAMPLES to a factor identifying the samples. Alternatively, if the samples are all of the same size and occur in the DATA variate one sample at a time, you can set SAMPLES to a scalar indicating the size of each sample. Finally, if the samples are in separate variates, you can set DATA to a pointer containing the variates (SAMPLES is then unset).

Two charts are produced. The first chart plots the mean of each sample. It also contains a centre line (indicating a target value) and lines representing upper and lower control limits (bounding the zone outside which the process is said to be out of control). The MEAN and SIGMA parameters allow you to supply values for the process mean and standard deviation if these are available either as targets or from previous observations. If they are unset, or if they are set to scalars containing missing values, the values are calculated from the data values (see the *Methods* Section). The traditional chart (and the one that is most popular in the USA) sets the centre line at the mean, and the control limits at 3 × SIGMA and −3 × SIGMA from the mean. The alternative (often used in the UK and requested by setting option METHOD to probability) sets control limits according to probability values. Usually the lower control limit is at the equivalent deviate value for a probability of 0.01, and the upper limit is at the value for 0.99 (see the *Methods* Section). There may also be intermediate warning limits, usually at 0.025 and 0.975. These are the default probabilities used by SPSHEWHART, but you can set the PROBABILITY option to a variate containing one or two values to define other limits. (If the values are p_1 and p_2, the limits are then for probabilities p_1, p_2, $100-p_2$, $100-p_1$.)

The control limits relevant to each batch will depend on the sample sizes. The TOLERANCE

option determines whether an average sample size is used if the individual sizes are not exactly equal: this will happen unless either

```
MIN(sample_size) * TOLERANCE < MEAN(sample_size)
```

or

```
MEAN(sample_size) * TOLERANCE < MAX(sample_size)
```

The second chart is either for the standard deviation of values in each sample or for their range, according to the setting of the PLOT option (by default PLOT=standarddeviation). Traditionally, before computers were available, the range chart was more popular. However, it is less sensitive than the standard deviation, particularly for larger samples, and SPSHEWHART does not permit range charts if any sample size is greater than 25.

If the number in each sample is one, the chart of the means is known as an individuals chart. There is now no within-sample replication, so the range chart instead presents a moving range displaying the range between each sample and the previous sample. Similarly, the standard deviations are calculated between each sample and its previous sample.

You can set PRINT=warnings to list any batches that are outside the control limits; by default these are suppressed. As usual, the WINDOWS option specifies which high-resolution graphics windows to use for the plots. If this is unset, SPSHEWHART automatically sets up and uses two windows containing the upper and lower halves of the screen. The SCREEN option controls whether or not to clear the graphics screen before plotting the charts.

Options: PRINT, PLOT, METHOD, TOLERANCEMULTIPLIER, PROBABILITY, WINDOWS, SCREEN.
Parameters: DATA, SAMPLES, MEAN, SIGMA.

Method

SPSHEWHART follows the standard methods as described for example by Nelson (1982), Montgomery (1985) or Ryan (1989). If required, the mean is estimated in the usual way by the average of the sample values. Likewise, the standard deviation is estimated by the average of the standard deviations of the samples, divided by a bias correction constant c_4:

$c_4 = \sqrt{(2/n)} \times$ GAMMA$(n/2)$ / GAMMA$((n-1)/2)$

where n is the sample size.

First of all we describe the calculations with METHOD=sigma. In the mean chart, the centre line is at the mean (i.e. MEAN), and the control limits at MEAN + 3 × SIGMA and MEAN − 3 × SIGMA. In the range chart, if the standard deviation has been supplied, the centre line is at d_2 × SIGMA and the control limits at D_1 × SIGMA and D_2 × SIGMA; if the standard deviation has not been supplied, the centre line is at the mean of the ranges observed in the samples, and the control limits are at D_3 × SIGMA and D_4 × SIGMA. (See Appendix VI of Montgomery, or Nelson 1982 Table 1 for values of the constants d_2, and $D1-D_4$.) In the standard-deviation chart, the centre line is at SIGMA × c_4 (so that it exhibits the same bias as the sample standard deviations) and the control limits are at 3 × SIGMA × $\sqrt{(1 - c_4^2)}$ above and below the centre line.

For METHOD=probability, the centre lines are unaffected. However, the control limits for the means chart are now at

```
EDNORMAL(PROBABILITY) * SIGMA / SQRT(N)
```

above and below the centre line. For the range chart, the control limits are at

```
SIGMA * EDSRANGE(PROBABILITY; 1000; N)
```

and

```
SIGMA * EDSRANGE(1-PROBABILITY; 1000; N)
```

(where the high value 1000 used for the degrees of freedom of the Studentized range is to obtain the value for the Normal range). For the standard-deviation chart, the control limits are at

```
SQRT(EDCHI(PROBABILITY; N-1) / (N-1))
```

and

```
SQRT(EDCHI(1-PROBABILITY; N-1) / (N-1))
```

Action with `RESTRICT`

Neither the `DATA` variates nor the `SAMPLE` factors may be restricted.

References

Montgomery, D.C. (1985). *Introduction to Statistical Process Control*. Wiley, New York.

Nelson, L.S. (1982). Control charts. In: *Encyclopedia of Statistical Sciences* (ed. S. Kotz, N.L. Johnson & C.B. Read), Volume 2, 176-183. Wiley, New York.

Ryan, T.P. (1989). *Statistical Methods for Quality Improvement*. Wiley, New York.

Shewhart, W.A. (1931). *Economic Control of Quality of Manufactured Product*. Van Nostrand, New York.

See also

Procedures: `SPCAPABILITY`, `SPCCHART`, `SPCUSUM`, `SPEWMA`, `SPPCHART`.

Genstat Reference Manual 1 Summary section on: Six sigma.

SPSYNTAX

Puts details about the syntax of commands into a spreadsheet (R.W. Payne).

Option

OUTFILENAME = *texts*	Name of Genstat file (.gsh or .gwb) or Excel (.xls or .xlsx) file to create

Parameter

COMMAND = *texts*	Single-line texts specifying the commands

Description

SPSYNTAX forms a spreadsheet containing details about the syntax of commands into a spreadsheet. By default, the spreadsheet is a Genstat workbook, opened in the Genstat client. Alternatively, the OUTFILENAME can specify the name of a spreadsheet file where the details are to be saved. This can be a Genstat workbook (.gwb), or a Genstat spreadsheet (.gsh), or an Excel spreadsheet (.xls or .xlsx) If the name is specified without a suffix, '.gwb' is added (so that a Genstat workbook is saved).

The COMMAND parameter lists the names of the directives and procedures whose syntax is to be saved. If the output is to a Genstat workbook, or to an Excel file, the spreadsheet file is created with a page for each command. An ordinary Genstat spreadsheet (.gsh) can save details for only one command. This will be the last command in the list.

Each page is a vector sheet, with columns for each aspect of the syntax. (See parameters NAME, MODE, NVALUES, VALUES, DEFAULT, SET, DECLARED, TYPE, COMPATIBLE, PRESENT, LIST and INPUT of the SYNTAX directive.) The aspects that can have multiple settings (i.e. those that save pointers in SYNTAX) are represented by pointers in the spreadsheet so that they have several columns to save these multiple settings. The pointers have a zero element to specify the number of columns used by each option or parameter. The options and parameters that can have both string and number settings have two pointers, one containing texts for the strings, and another containing variates for the numbers.

Option: OUTFILENAME.
Parameter: COMMAND.

See also

Directives: COMMANDINFORMATION, OPTION, PROCEDURE, SYNTAX.
Genstat Reference Manual 1 Summary section on: Program control.

SSIGNTEST

Calculates the sample size for a sign test (R.W. Payne).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
PROBABILITY = *scalar*	Significance level at which the response is to be tested; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Whether to a one- or two-sided test is to be made (onesided, twosided); default twos
REPLICATION = *variate*	Replication values for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

RESPONSE = *scalars*	Probability of response (i.e. the probability that an observation in one sample will be greater than the equivalent observation in the other sample) that should be detectable
NREPLICATES = *scalars*	Saves the required number of replicates
VREPLICATION = *variates*	Numbers of replicates for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

SSIGNTEST calculates the number of replicates (or sample size) required for a sign test (see procedure SIGNTEST). By default the calculations are done for a one-sided test (testing for evidence that the location of one sample is greater than the other, but you can set option TMETHOD=onesided for a two-sided test (testing that locations of the samples are different). The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%).

The probability of response (i.e. the probability that an observation in one sample will be greater than the equivalent observation in the other sample) that should be detectable is supplied by the RESPONSE parameter. The required probability for detection of the response (that is, the *power* of the test) is specified by the POWER option (default 0.9). The sample size can be saved using the NREPLICATES parameter.

The PRINT option controls printed output, with settings:

replication	to print the required number of replicates in each sample (i.e. the size of each sample);
power	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding powers can also be saved, in variates, using the VREPLICATION and VPOWER parameters. The REPLICATION option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: PRINT, PROBABILITY, POWER, TMETHOD, REPLICATION.
Parameters: RESPONSE, NREPLICATES, VREPLICATION, VPOWER.

Method
An approximate number of replicates is calculated initially assuming a Normal approximation. This is then refined by calculating powers for a range of replications centred around that approximation.

See also
Procedure: `SIGNTEST`.
Genstat Reference Manual 1 Summary section on: Design of experiments.

STACK

Combines several data sets by "stacking" the corresponding vectors (R.W. Payne).

Option

`DATASET` = *factor*	Factor to indicate the data set to which each unit originally belonged

Parameters

`STACKEDVECTOR` = *variates*, *factors* or *texts*	New vectors combining the corresponding members of the data sets specified by parameter `V1`, or parameters `V1`-`V100`
`V1` = *pointers*, *variates*, *factors*, *texts* or *scalars*	Pointers defining (all) the components to be stacked into each `STACKEDVECTOR`, or contents of the first data set
`V2` - `V100` = *variates*, *factors*, *texts* or *scalars*	Data sets 2 - 100
`FREPRESENTATION` = *string token*	How to match the values of factors (`levels`, `labels`, `ordinals`, `renumbered`); default `leve`

Description

`STACK` allows you to combine vectors (variates, factors or texts) from several data sets into a single data set. Each vector in the new data set is formed by "stacking" the corresponding vectors from the original data sets. So, the new vector first has all the units from the first data set, then those from the second data set, and so on.

The identifiers of the new vectors are specified by the first parameter, `STACKEDVECTOR`. The original vectors of up to 100 data sets can be specified one data set at a time using the subsequent parameters: `V1`, `V2`, ... `V100`. Alternatively, `V1` can specify a list of pointers, each one containing all the vectors that are to be stacked together to form the equivalent `STACKEDVECTOR` (allowing vectors from more than 100 data sets to be specified). So, these two statements would be equivalent

```
STACK [DATASET=Month] Rainfall,Temperature;\
  V1=MarchRain,MarchTemp; V2=AprilRain,AprilTemp
```

and

```
STACK [DATASET=Month] Rainfall,Temperature;\
  V1=!p(MarchRain,AprilRain),!p(MarchTemp,AprilTemp)
```

The vectors in each data set must generally all be of the same length. The exception is that you can specify a scalar instead of a variate of identical values (the number of values is then deduced from the lengths of the corresponding vectors of the other data sets). Likewise you can specify a single-valued text instead of a text with duplicates of that value, and either a scalar or a single-valued text instead of a factor with the same level or label duplicated throughout.

The `FREPRESENTATION` option indicates how the levels are to be matched amongst factors. If this is set to `labels` and the levels of the original factors are compatible (that is if each label corresponds to the same level in all the original factors), then the level definitions are transferred to the new factor; if not, the levels are defined to be the default values 1, 2... and a warning is printed by the `APPEND` procedure which is called by `STACK`. Similarly, with the default setting `levels`, the labels are retained if they are compatible, but no warning is printed if they are not. For the `ordinals` setting, the levels of all the factors are taken as the ordinal values 1, 2... (and no labels are defined). Finally, the `renumbered` setting assumes that the original factors all have independent sets of levels, and renumbers these from one upwards for the first factor, from number of levels of the first factor plus one upwards for the second factor, and so on; the new

factor will thus have a different level for every level of the original factors.

The DATASET option allows a factor to be formed indicating the number if the data set to which each unit of the stacked vectors originally belonged. This factor could be used in the DATASET parameter of the UNSTACK procedure subsequently to recover the original vectors.

Option: DATASET.
Parameter: STACKEDVECTOR, V1, V2, ... V100, FREPRESENTATION.

Method
The vectors are stacked together using the APPEND procedure.

Action with RESTRICT
Any restrictions on the vectors are ignored.

See also
Directive: EQUATE.
Procedures: APPEND, JOIN, RESHAPE, UNSTACK, VEQUATE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

STANDARDIZE

Standardizes columns of a data matrix to have mean zero and variance one (S.A. Harding & D.A. Murray).

No options

Parameters

OLD = *variates* or *matrices*	Structures containing data to be standardized
NEW = *variates* or *matrices*	Structures to contain output; by default the OLD structures are overwritten

Description

The parameter OLD lists the variates or matrices to be standardized, and the NEW parameter specifies a list of variates or matrices to store the standardized values. If NEW is not set, the transformed data values overwrite the contents of the OLD structures. If NEW is set, it should be to structures of the same type (variate or matrix) as the corresponding OLD structures.

Options: none. Parameters: OLD, NEW.

Method

The standardized values are calculated as $(x - \text{mean}(x)) / \sqrt{(\text{var}(x))}$. If there are any missing values in the data these are omitted from the calculation.

Action with RESTRICT

Restrict is irrelevant with matrix structures. It should work as expected with variates.

See also

Function: STANDARDIZE .
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

STEEL

Performs Steel's many-one rank test (R.W. Payne).

Options

PRINT = *string token*	Controls printed output (description, sumranks, critical, permutationtest); default desc, sumr, crit
METHOD = *string token*	Form of the alternative hypothesis (twosided, greaterthan, lessthan); default twos
TREATMENTS = *factor*	Defines the treatments
CONTROL = *scalar* or *text*	Treatment level corresponding to the control; default takes the reference level of TREATMENTS
NTIMES = *scalar*	Number of permutations for the permutation test; default 999
SEED = *scalar*	Seed to use to generate the random numbers for the permutation test; default 0

Parameters

DATA = *variates*	Data values for the tests
SUMRANKS = *tables*	Saves the sum of the ranks within the treatments from each test
RANKS = *variates*	Saves the ranks of the data values for each test

Description

Steel's test (Steel 1959) is a multiple-comparison test for comparing several treatments with a control treatment. The data are assumed to come from a one-way classification where all the treatments (and the control) have equal replication. The data values are specified, in a variate, using the DATA parameter. The TREATMENTS option species a factor to indicate the allocation of data values to treatments. The CONTROL option indicates which level of the TREATMENTS factor is the control; if this is not set, the reference level of TREATMENTS is used.

The METHOD option defines the type of test that is done. By default STEEL does a two-sided test, so the test is against the alternative hypothesis that the treatments may be either less than or greater than the control. If you set METHOD=lowerthan, STEEL does a one-sided test of the null hypothesis that the treatment values are not lower than the control. Alternatively, you can set METHOD=greaterthan, to do a one-sided test of the null hypothesis that the treatment values are not greater than the control.

The test operates by comparing the data values from each treatment in turn with the control. The comparison is made by pooling the data values from the treatment and control, forming their ranks, and calculating the sum of the ranks for the treatment data values. For METHOD=greaterthan, the test statistic for each treatment is simply the sum of the ranks for each treatment. For METHOD=lessthan, each rank sum must be subtracted from the total sum of ranks $(2n + 1) \times n$, where n is the replication of the treatments. For METHOD=twosided, the statistic is the minimum of the greaterthan and the lessthan statistics.

The PRINT option controls printed output, with settings:

description	description of the data and test;
sumranks	the test statistics (sums of ranks for each treatment);
critical	critical value as provided by Steel (1959);
permutationtest	uses a random permutation test to forms critical values and the probability that any treatment differs from control (according to the test specified by METHOD).

By default these are all produced.

By default, when PRINT=perm, STEEL makes 999 random allocations of the data to the treatment and control groups (using a default seed), and determines critical values for the test from the distribution of the minimum rank sum over these randomly generated datasets. The NTIMES option allows you to request another number of allocations, and the SEED option allows you to specify another seed. STEEL checks whether NTIMES is greater than the number of possible ways in which the data values can be allocated. If so, it does an exact test instead, which takes each possible allocation once. The results should be more reliable than Steel's critical values, which are based on a multivariate Normal approximation.

The rank sums can be saved using the SUMRANKS parameter, and the ranks of the individual treatment data values can be saved using the RANKS parameter.

Options: PRINT, METHOD, TREATMENTS, CONTROL, NTIMES, SEED.
Parameters: DATA, SUMRANKS, RANKS.

Action with RESTRICT

DATA or TREATMENTS can be restricted to analyse a subset of the data values.

Reference

Steel, R.G.D. (1959). A multiple comparison rank sum test: treatments versus control. *Biometrics*, **15**, 560-572.

See also

Procedures: AMCOMPARISON, AUMCOMPARISON, AMDUNNETT, CONFIDENCE, VMCOMPARISON.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

STEM

Produces a simple stem-and-leaf chart (J. Ollerton & S.A. Harding).

No options

Parameters

DATA = *variates*	Data values for each plot
NDIGITS = *scalars*	Number of digits in the leaves of each plot
STEMUNITS = *scalars*	Scale units for the stem values in each plot

Description

STEM produces a simple stem-and-leaf chart of a variate of data. The stems indicate leading digits and the leaves indicate subsequent digits. By default, the leaves are formed from single digits; the parameter NDIGITS can be used to specify the number of digits in each leaf if more than one is required. The STEMUNITS parameter can be used to specify the units represented by the stem values. By default, this is determined from the data so that the display will fit within a single screen or page of output. Small values of STEMUNITS (in comparison to the range of the data) should be avoided as they may generate far too many lines of output. The display produced by STEM is restricted to the current output width; any lines that have to be truncated at the right-hand margin are terminated by >, indicating their continuation.

Options: none. Parameters: DATA, NDIGITS, STEMUNITS.

Method

The variate of data is scaled appropriately and printed into a text. Using CONCATENATE the individual stem and leaf digits are extracted and then formed into another text structure which is printed out as the stem-and-leaf display. Missing values are excluded from the data.

Action with RESTRICT

STEM takes account of any restriction on the data variate.

Reference

Tukey, J.W. (1977). *Exploratory Data Analysis*. Addison-Wesley.

See also

Directive: DHISTOGRAM.
Procedures: BOXPLOT, DOTPLOT, RUGPLOT.
Genstat Reference Manual 1 Summary section on: Graphics.

STTEST

Calculates the sample size for t-tests, including equivalence tests (R.W. Payne).

Options

PRINT = *string token*	What to print (replication, power); default repl, powe
NSAMPLES = *scalar*	Number of samples for the t-test (1 or 2); default 2
PROBABILITY = *scalar*	Significance level at which the response is to be tested; default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Type of test to be done (onesided, twosided, equivalance, noninferiority); default ones
RATIOREPLICATION = *scalar*	Ratio of replication sample2:sample1 (i.e. the size of sample 2 should be RATIOREPLICATION times the size of sample 1); default 1
REPLICATION = *variate*	Replication values for which to calculate and print or save the power; default * takes 11 replication values centred around the required number of replicates

Parameters

RESPONSE = *scalars*	Response to be detected
VAR1 = *scalars*	Anticipated variance of sample 1
VAR2 = *scalars*	Anticipated variance of sample 2; default * assumes the same variance as sample 1
NREPLICATES = *scalars*	Saves the required number of replicates
VREPLICATION = *variates*	Numbers of replicates for which powers have been calculated
VPOWER = *variates*	Power (i.e. probability of detection) for the various numbers of replicates

Description

STTEST calculates the number of replicates (or sample size) required for various types of t-test. The calculations can be done for a one-sample t-test (testing for evidence that the mean of the sample differs from a specific value) or a two-sample test (testing that means of the samples are different). The number of samples is specified by the NSAMPLES option (default 2).

The size of response that should be detectable is supplied by the RESPONSE parameter. (This is difference between the sample mean of a one-sample test and the specific value, or the difference between the means of the two samples in a two-sample test.) The VAR1 parameter supplies the variance of the observations in the sample of a one-sample test or of the first sample of a two-sample test. If the second sample of a two-sample test has a different variance from the first sample, this can be supplied by the VAR2 parameter.

The significance level for the test is specified by the PROBABILITY option (default 0.05 i.e. 5%). The required probability for detection of the response (that is, the *power* of the test) is specified by the POWER option (default 0.9). It is generally assumed that the sizes of the samples in the two-sample test should be equal. However, you can set the RATIOREPLICATION option to a scalar, R say, to indicate that the size of the second sample should be R times the size of the first sample. The NREPLICATES parameter allows you to save the required size of the first sample.

By default, STTEST assumes a one-sided t-test is to be used, but you can set option TMETHOD=twosided to take a two-sided t-test instead. Other settings of TMETHOD enable you

to test for equivalence or for non-inferiority. To demonstrate equivalence of the two samples (`TMETHOD=equivalence`), their means m_1 and m_2 must differ by less than some threshold d; this is specified by `RESPONSE` and should represent a limit below which the difference can be assumed to have no physical (or clinical) importance. Statistically, equivalence implies comparing a null hypothesis that the samples are not equivalent, i.e.

$$(m_1 - m_2) \leq -d$$

or

$$(m_1 - m_2) \geq d$$

with the alternative hypothesis that they are equivalent, i.e.

$$-d < (m_1 - m_2) < d$$

A one-sample test for equivalence operates similarly, but here d specifies the threshold for the sample mean itself. To demonstrate non-inferiority of sample 1 compared to sample 2, the null hypothesis becomes

$$(m_1 - m_2) \geq -d$$

(which, in fact, represents a simple one-sided t-test).

The `PRINT` option controls printed output, with settings:

`replication`	to print the required number of replicates in each sample (i.e. the size of each sample);
`power`	to print a table giving the power (i.e. probability of detection) provided by a range of numbers of replicates.

By default both are printed.

The replications and corresponding powers can also be saved, in variates, using the `VREPLICATION` and `VPOWER` parameters. The `REPLICATION` option can specify the replication values for which to calculate and print or save the power; if this is not set, the default is to take 11 replication values centred around the required number of replicates.

Options: `PRINT`, `NSAMPLES`, `PROBABILITY`, `POWER`, `TMETHOD`, `RATIOREPLICATION`, `REPLICATION`.

Parameters: `RESPONSE`, `VAR1`, `VAR2`, `NREPLICATES`, `VREPLICATION`, `VPOWER`.

Method

An approximate number of replicates is calculated initially assuming a Normal approximation. This is then refined by calculating powers for a range of replications centred around that approximation.

In the equivalence test, comparing the null hypothesis that the samples are not equivalent, i.e.

$$(m_1 - m_2) \leq -d$$

or

$$(m_1 - m_2) \geq d$$

with the alternative hypothesis that they are equivalent, i.e.

$$-d < (m_1 - m_2) < d$$

defines an *intersection-union* test, in which each component of the null hypothesis must be rejected separately. This implies performing two one-sided t-tests (this is known as a *TOST* procedure). If the significance level for the full test is to be α, each t-test must have significance level α (see Berger & Hsu 1996). To obtain a detection probability (or power) of $(1 - \beta)$, each of the t-tests must have detection probabilities of $(1 - \beta/2)$.

Reference

Berger, M.L. & Hsu, J.C. (1996). Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Statistical Science*, **11**, 283-319.

See also
Procedure: TTEST.
Genstat Reference Manual 1 Summary section on: Design of experiments.

SUBSET

Forms vectors containing subsets of the values in other vectors (R.W. Payne).

Options

CONDITION = *expression*	Logical expression to define which units are to be included; no default – this option must be set
SETLEVELS = *string token*	Whether to reform the levels (and labels) of factors to exclude those that do not occur in the subset (yes, no); default no
NULL = *scalar*	Indicator set to 1 or 0 according to whether or not the subset contains no units

Parameters

OLDVECTOR = *vectors*	Vector from which the subset is to be formed
NEWVECTOR = *vectors*	Vector to store the subsets if none is specified, the OLDVECTOR is redefined to store the subset

Description

SUBSET forms vectors containing subsets of the values in other vectors. The subset is defined by a logical condition which must be specified by the CONDITION option; units with *true* values (non-zero and non-missing) for the condition are included in the subset, others are omitted.

Subsets can be formed for factors, texts and variates. Relevant attributes will also be transferred across to the new structures but, if the subset excludes some of the levels of a factor, a new reduced set of levels (and labels) can be requested by setting option SETLEVELS=yes.

The NULL option can specify a scalar that will be set to one if the subset contains no units; otherwise it is set to zero. Also, when NULL set, SUBSET suppresses the fault that it normally gives if the subset is empty.

The original vectors are specified by the OLDVECTOR parameter and identifiers for the vectors to contain the subsets are specified by the NEWVECTOR parameter. If NEWVECTOR is not set, the OLDVECTOR are redefined to store the subsets instead of their original values.

Options: CONDITION, SETLEVELS, NULL.
Parameters: OLDVECTOR, NEWVECTOR.

Method

RESTRICT is used to obtain a list of units included according to the CONDITION. This is then used to calculate a format for EQUATE to use to transfer the values. The DUPLICATE directive is used to transfer any relevant attributes. We thank Jac Thissen for suggestions about the redefinition of factor levels.

Action with RESTRICT

Any restriction is ignored; the subset is formed only from the CONDITION option.
OLDVECTOR is redefined to store the subset.

See also

Directive: EQUATE.
Procedure: UNSTACK.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

SVBOOT

Bootstraps data from random surveys (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (summary); default * i.e. none
SEED = *scalar*	Seed for random numbers; default 0
STRATUMFACTOR = *factor*	Stratification factor
SAMPLINGUNITS = *factor*	Sampling units (default single stage design)
WEIGHTS = *variates*	Weights variates (not required for simple bootstrap)
METHOD = *string token*	Method (simple, sarndal); default simp
POPULATION = *pointers*	Units in the population
SAVEUNITS = *variate*	Units in the bootstrapped sample
BSTRATUMFACTOR = *factor*	Bootstrapped stratification factor
BSAMPLINGUNITS = *factor*	Bootstrapped sampling units

Parameters

DATA = *variates* or *factors*	Data to bootstrap
BOOT = *variates* or *factors*	Saves bootstrap sampling units

Description

SVBOOT forms a single bootstrap sample using data from a stratified one- or two-stage survey. It is designed to be used in a FOR loop, with a new sample being formed and analysed each time that the loop is executed. The DATA parameter supplies a list of structures to be bootstrapped, whilst BOOT contains the corresponding bootstrapped structures. Alternatively, the SAVEUNITS option can be used to save the units in the bootstrapped samples, allowing the bootstrapped structures to be formed by a CALCULATE statement. Options STRATUMFACTOR and SAMPLINGUNITS supply the stratification factor and the sampling units respectively, whilst survey weights are supplied by the WEIGHTS option.

When option METHOD=simple, sampling is with replacement within each stratum. This is the correct approach for an infinite population, but will give reasonable results as long as sampling proportion is not very high. METHOD=sarndal uses the method described by Sarndal *et al*. (1992, page 442), as implemented by Grilli & Pratesi (2004), in which an artificial population is created, containing each element of the sample *w* times, where *w* is the survey weight (the inverse of the probability of inclusion), rounded to the nearest integer. Sampling is then carried out without replacement (not with replacement as Sarndal recommends). For two-stage sampling WEIGHTS should be set to a list of two variates, the first giving the overall sampling weights and the second the weights at the first stage only (typically the inverse of the probability of selection of the primary sampling units).

The Sarndal approach works well as long as either the weights are integers, or they are large enough that the effect of rounding is negligible. For surveys with high sampling fractions, METHOD=random implements a variant on the Sarndal method in which the artificial population is formed by a random process, using resampling in proportion to the weights and ensuring that each observation is present at least once in the population. Care must be taken when using this method, as means, totals and other statistics will vary slightly between the different artificial populations. With this method it may sometimes be helpful to form repeated bootstrap samples from the same pseudo-population; this can be achieved by means of the POPULATION option.

Except in simple surveys with no restrictions, the number of units in each bootstrapped sample will not be the same as the original survey and so options BSTRATUMFACTOR and BSAMPLINGUNITS save new factors for use with the bootstrapped structures.

Options: PRINT, SEED, STRATUMFACTOR, SAMPLINGUNITS, WEIGHTS, METHOD, POPULATION, SAVEUNITS, BSTRATUMFACTOR, BSAMPLINGUNITS.
Parameters: DATA, BOOT.

Method

a) simple, one-stage
A new variate is formed for each stratum containing the unit numbers associated with each stratum, indexed by a grouping factor. The new bootstrap sample is then formed by selecting from these at random with replacement. Any weights set are ignored. The new samples are in stratum order, rather than the order of the original dataset.

b) simple, two-stage
The method described above is applied twice, once to select primary sampling units at random from those in the stratum, and once to select secondary sampling units from those in the appropriate psu.

c) Sarndal, one-stage
An artificial population is generated for each stratum, with each unit being replicated w times, where w is the appropriate weight, rounded to the nearest integer. Sampling is then carried out, without replacement, using the inverse of the weights as inclusion probabilities. For reasons of computational simplicity, the bootstrap sample sizes are not fixed, and will therefore differ slightly from the one in the original sample.

d) Sarndal, two-stage
The method described above is applied twice, once to select primary sampling units at random from those in the stratum, and once to select secondary sampling units from those in the appropriate psu.

e) Random
This method is designed as an alternative to the Sarndal method when the sampling fraction is very high, so that the rounded weights are equal to one and the same sample is always generated. The pseudo-population is formed by including each of the sampled observations once and then resampling with replacement from the sampled observations to generate the remaining $N-n$ units in the pseudo-population (where N is the population size, and n is the sample size in the stratum). This method is currently only implemented for one stage sampling with equal weights in a stratum. The pseudo-population is then sampled without replacement, as in the Sarndal method.

Action with RESTRICT

Restricted units are excluded from the bootstrapping process and do not occur in the resampled dataset The restriction is defined by the first variate in the DATA list, if this is set.

References

Grilli, L. & Pratesi, M. (2004). Weighted estimation in multilevel ordinal and binary models in the presence of informative sampling designs. *Survey Methodology*, **30**, 93-103.

Sarndal, C., Swensson, B. & Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer-Verlag, New York.

See also

Procedures: BOOTSTRAP, SVCALIBRATE, SVGLM, SVHOTDECK, SVREWEIGHT, SVSAMPLE, SVSTRATIFIED, SVTABULATE, SVWEIGHT.
Genstat Reference Manual 1 Summary section on: Survey analysis.

SVCALIBRATE

Performs generalized calibration of survey data (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (summary, totals, monitoring); default summ, tota
PLOT = *string token*	Controls which high-resolution graphs are plotted (weights); default * i.e. none
STRATUMFACTOR = *factor*	Stratification factor; default * i.e. unstratified
SAMPLINGUNITS = *factor*	Factors indicating the sampling units in a two-stage design; default *, i.e. single-stage design
TCONSTRAINTS = *scalars*	Constraint totals or tables
X = *variates*	Variates corresponding to TCONSTRAINTS; * implies the equivalent constraint relates to a count
WEIGHTS = *variate*	Initial weights
OUTWEIGHTS = *variate*	Final (calibration) weights
METHOD = *string token*	Method to use (linear, truncatedlinear, logistic, fittedvalues); default line
LOWER = *scalar*	Lower bound for g-weights; default 0.1
UPPER = *scalar*	Upper bound for g-weights; default 10
MAXCYCLE = *scalar*	Maximum number of iterations; default 50
TOLERENCE = *scalar*	Tolerence for convergence; default 0.0001

Parameters

Y = *variates*	Response data for analysis
TOTALS = *scalars*	Saves estimated totals
SETOTALS = *scalars*	Saves standard errors of totals
FITTEDVALUES = *variates*	Saves fitted values from the regression

Description

SVCALIBRATE performs calibration estimation of survey data (Deville & Sarndal 1992). The sampling weights from a survey are often adjusted to ensure that they produce estimates that match known population totals. For example, if in an agricultural survey the sampling weights are applied to the areas of the sampled farms, the resulting estimate will not generally exactly equal the known total agricultural area in the population, and so an adjustment is required. Calibration calculates adjusted weights that ensure the constraints are met, while remaining as close as possible to the original sampling weights.

The TCONSTRAINTS option is used to specify the constraints, either in a scalar to provide a total for the whole population, or in a table specifying totals for subgroups defined by the classification factors of the table. The X option specifies a list of variates (in parallel) to which the constraints relate, with a null value indicating that the corresponding constraint relates to a count of units in the population. If STRATUMFACTOR is set a separate calibration is performed in each stratum and TCONSTRAINTS must be set to one or more tables, classified by the stratification factor. The SAMPLINGUNITS option can be used to specify primary sampling units in a two stage design; this information is only used for calculation of the standard error of the total and does not affect the calibration process. The WEIGHTS option specifies the initial sampling weights, which will usually be the inverse of the probability of selection of each unit, whilst OUTWEIGHTS returns the adjusted weights.

The METHOD option controls the restrictions on the range of adjustments (the "g-weights") used to convert the initial to the modified weights and has three possible settings: linear produces estimates equivalent to the usual regression estimates, the g-weights are not restricted

and may be negative; truncatedlinear restricts the g-weights to the range specified by the LOWER and UPPER options by replacing extreme values with these bounds; logistic uses a logit-like transformation to ensure that the weights remain within the specified bounds. These correspond to methods 1, 5 and 7 respectively of Singh & Mohl (1996). The last two methods use iterative calculations which are controlled by the MAXCYCLE and TOLERENCE options. Progress of the iterations can be viewed using the monitoring setting of PRINT. The default values for LOWER and UPPER are 0.1 and 10, thus allowing the adjusted weights to differ from the initial weights by a factor of ten in either direction.

The procedure can be run without setting any options, in order to produce adjusted weights for use with TABULATE or SVTABULATE. Alternatively the first parameter, Y, may be used to specify variates for which estimates are required. The estimates of totals and approximate standard errors can be saved using the TOTALS and SETOTALS parameters. More complex analyses (e.g. cross-tabulations, and two-stage analyses with a finite population correction) can be achieved by saving the OUTWEIGHTS and using them as input weights for SVTABULATE. Fitted values from the generalized regression method (METHOD=linear) are saved in FITTEDVALUES; these are needed to calculate the correct asymptotic standard errors for estimates produced using the weights by means of SVTABULATE. You can produce FITTEDVALUES without any calibration, by setting METHOD=fittedvalues; this avoids having to repeat the full calibration process when analysing additional Y variates.

Options: PRINT, PLOT, STRATUMFACTOR, SAMPLINGUNITS, TCONSTRAINTS, X, WEIGHTS, OUTWEIGHTS, METHOD, LOWER, UPPER, MAXCYCLE, TOLERENCE.
Parameters: Y, TOTALS, SETOTALS, FITTEDVALUES.

Action with RESTRICT

Any restriction on WEIGHTS, OUTWEIGHTS or Y excludes the restricted units from the calibration process, so that their values of WEIGHTS pass unchanged to OUTWEIGHTS. TCONSTRAINTS should be based only on the unrestricted units and, if Y is set, estimates of the total are for the subpopulation defined by the restrictions on WEIGHTS. Any restrictions on X are ignored.

References

Deville, J.-C. & Sarndal, C.-E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association*, **87**, 376-382.

Singh, A.C. & Mohl, C.A. (1996). Understanding calibration estimators in survey sampling. *Survey Methodology*, **22**, 107-115.

See also

Procedures: SVBOOT, SVGLM, SVHOTDECK, SVREWEIGHT, SVSAMPLE, SVSTRATIFIED, SVTABULATE, SVWEIGHT.
Genstat Reference Manual 1 Summary section on: Survey analysis.

SVGLM

Fits generalized linear models to survey data (S.D. Langton).

Options

PRINT = *string token*	What output to display (model, summary, estimates, wald, predictions, monitor); default mode, esti, wald, pred
DISTRIBUTION = *string token*	Error distribution (binomial, poisson, normal, gamma); default norm
LINK = *string token*	Link function (identity, logarithm, logit, reciprocal, probit, complementaryloglog, canonical); default cano
DISPERSION = *scalar*	Value at which to fix the residual variance, if missing the variance is estimated; default 1 for binomial or Poisson, otherwise *
TERMS = *formula*	Explanatory model
CONSTANT = *string token*	Whether to estimate or omit constant term in fixed model (omit, estimate); default esti
FACTORIAL = *scalar*	Limit on number of factors/covariates in a model term; default 3
PFACTORS = *factors* or *variates*	Variables for which predictions are to be formed; default *, or as specified in PTERMS
PLEVELS = *variates* or *scalars*	Levels or values at which predictions are to be made corresponding to PFACTORS; default (weighted) mean for variates, all levels for factors
PTERMS = *formula*	Formula specifying fixed terms for which predicted means are to be printed; default *, unless PFACTORS is set, in which case it is all main effects of and interactions between PFACTORS
STRATUMFACTOR = *factor*	Stratification factor; default *, i.e. unstratified
NUNITS = *variate* or *table*	Number of primary sampling units in each stratum
SAMPLINGUNITS = *factor*	Factor indicating the primary sampling units; default *, i.e. single stage design
WEIGHTS = *variates*	Survey weights
METHOD = *string token*	Bootstrapping method (simple, csimple, sarndal); default simp
NBOOT = *scalar*	Number of bootstrap samples to use; default 0 uses a Taylor series approximation
SEED = *scalar*	Seed for random number generator for bootstrap; default 0
CIPROBABILITY = *scalars*	The probability level for the confidence intervals; default 0.95
CIMETHOD = *string token*	Method for forming confidence intervals (automatic, tdistribution, percentile); default auto

Parameters

Y = *variates*	Dependent variates
NBINOMIAL = *scalars* or *variates*	Number of binomial trials for each unit (must be set if DISTRIBUTION=binomial)
RESIDUALS = *variates*	Variates to save residuals
FITTEDVALUES = *variates*	Variates to save fitted values

ESTIMATES = *variates*	Estimates of parameters for each Y variate
SE = *variates*	Standard errors of the estimates
VCOVARIANCE = *symmetric matrices*	Variance-covariance matrix for the estimates
LOWER = *variates*	Lower confidence limits for estimates
UPPER = *variates*	Upper confidence limits for estimates
WALD = *pointers*	Pointers to save Wald statistics for each term (pointer contains name of term, Wald statistic, F statistic, degrees of freedom, and P-value)
PREDICTIONS = *pointers*	Pointers to tables of predictions
SEPREDICTIONS = *pointers*	Pointers to tables of standard errors of predictions
LOWPREDICTIONS = *variates*	Lower confidence limits for predictions
UPPREDICTIONS = *variates*	Upper confidence limits for predictions
VCPREDICTIONS = *symmetric matrices*	Variance-covariance matrix for the predictions

Description

SVGLM fits generalized linear models to data from one- or two-stage surveys. Variance estimates reflecting the survey design are estimated by a bootstrap method or a Taylor series approximation (Korn & Graubard 1999). Survey weights, which are supplied using the WEIGHTS option and which may be calculated by SVWEIGHT, are used to ensure that unbiased estimates of the finite survey population parameters are produced. It should be noted that using a weighted analysis is not the only way to handle such data; in some circumstances it may be preferable to use an unweighted analysis, including factors reflecting the survey design (see, for example, Chapter 5 of Korn & Graubard 1999 for discussion of this subject). Mixed models, such as those fitted by the REML directive, the GLMM procedure or the HGANALYSE procedure may be another way of accounting for the correlations induced in the data by the survey design.

The DISTRIBUTION, LINK, DISPERSION, CONSTANT and FACTORIAL options are used to specify the model in exactly the same way as in the MODEL directive. Similarly the Y parameter supplies the response variable to be analysed and, for the binomial distribution, NBINOMIAL supplies the number of trials for each unit. The terms to be fitted are supplied using option TERMS as either a formula or, if no interactions are fitted, a list of variates and factors.

Information on the survey design is provided using the STRATUMFACTOR and SAMPLINGUNITS options. The option NUNITS can be used to list the number of primary sampling units per stratum, using a table or variate with one value for each stratum; this is used to calculate the appropriate degrees of freedom for test statistics and in construction of bootstrap samples.

The bootstrapping method is selected using the METHOD option. In a one-stage design the default of simple forms each bootstrap sample by sampling with replacement from the original sample within each stratum. In a two-stage design (i.e. if SAMPLINGUNITS is set), primary sampling units are first sampled with replacement, and then secondary units are sampled with replacement within the selected primary units. Variance estimates from the boostrapping process will be biased where there are very few sampling units in each stratum and so the method is not recommended in this situation. For a cluster sample the setting csimple should be used; this samples primary sampling units with replacement as for the two-stage design, but does not resampling within those secondary units. The setting METHOD=sarndal constructs a "pseudo-population" by replicating each sampled unit by the rounded value of its weight, so that, for example, an observation with weight 16.1 is represented sixteen times in the pseudo-population (see Sarndal *et al*. 1992, page 442). The bootstrap sample is formed by sampling with replacement from this pseudo-population. At present this method is only available for single-stage sampling.

The number of bootstrap samples used is set by means of the NBOOT parameter. For exploratory analyses a relatively low value (perhaps 20) may suffice, but where test statistics or confidence limits are required a value of at least 500 is recommended. For simple linear regression (i.e. DISTRIBUTION=normal), setting NBOOT to zero calculates variances of regression parameters by a linearization approach similar to that used for means and totals by SVTABULATE (Binder 1983). For other generalized linear models setting NBOOT to zero uses a simple approximation in which the weights are scaled to sum to the number of observations in the sample; this setting is only recommended for initial model fitting as variance estimates will be seriously inaccurate, particularly in two-stage designs.

Parameter estimates and their standard errors can be saved using the ESTIMATES and SE parameters, whilst VCOVARIANCE saves the full variance-covariance matrix. The LOWER and UPPER parameters save confidence limits for the estimates; by default 95% confidence limits are shown, but this may be changed by means of the CIPROBABILITY option. The CIMETHOD option controls how confidence limits are formed after bootstrapping: percentile uses simple percentiles of the bootstrapped distribution, whilst tdistribution calculates a standard error from the bootstrapped estimates and then uses the t-distribution to form intervals; the default of automatic uses the percentile method unless less than 400 bootstrap samples have been made.

Wald statistics (Korn & Graubard 1999) for terms in the model can be saved using parameter WALD, in the form of a pointer with elements corresponding to the term (as a text), the Wald statistic, the approximate F statistic, the two sets of degrees of freedom, and the probability value.

Predicted values can be formed from the analysis. These estimate the average value of the response variable that would have been expected in the population had all the units been in the specified group, or had had the specified covariate value. The averages are taken over the distribution of the other fitted variables within the population (as deduced from the weighted sample). Factors and variates for which predictions are required are specified using the PFACTORS option and particular levels or values may be specified using PLEVELS, which operates in the same way as the LEVELS parameter of PREDICT. Alternatively, PTERMS can be used to specify particular terms so that, for example, PTERMS=A.B would produce a two-way table classified by factors A and B. The parameters PREDICTIONS, SEPREDICTIONS, LOWPREDICTIONS, and UPPREDICTIONS save the tables of predictions, their standard errors, and the lower and upper confidence limits respectively. VCPREDICTIONS saves the full variance-covariance matrix of the bootstrapped predictions.

Printing is controlled by the PRINT option. The default output consists of model details, parameter estimates, Wald statistics and, if PFACTORS or PTERMS is set, predictions. The monitor setting provides progress of the bootstrap samples.

Options: PRINT, DISTRIBUTION, LINK, DISPERSION, TERMS, CONSTANT, FACTORIAL, PFACTORS, PLEVELS, PTERMS, STRATUMFACTOR, NUNITS, SAMPLINGUNITS, WEIGHTS, METHOD, NBOOT, SEED, CIPROBABILITY CIMETHOD.

Parameters: Y, NBINOMIAL, RESIDUALS, FITTEDVALUES, ESTIMATES, SE, VCOVARIANCE, LOWER, UPPER, WALD, PREDICTIONS, SEPREDICTIONS, VCPREDICTIONS, LOWPREDICTIONS, UPPREDICTIONS.

Action with RESTRICT

Restricting the response variate Y fits a model to the subpopulation defined by the restriction.

References

Binder, D.A. (1983). On the Variances of Asymptotically Normal Estimators from Complex Surveys. *International Statistical Review*, **51**, 279-292.

Sarndal, C., Swenssion, B. & Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer-Verlag, New York.

See also

Procedures: SVBOOT, SVCALIBRATE, SVHOTDECK, SVREWEIGHT, SVSAMPLE, SVSTRATIFIED, SVTABULATE, SVWEIGHT.

Genstat Reference Manual 1 Summary sections on: Survey analysis, Regression analysis, REML analysis of linear mixed models.

SVHOTDECK

Performs hot-deck and model-based imputation for survey data (S.D. Langton).

Options

`PRINT` = *string token*	Controls printed output (`summary`, `monitoring`, `check`, `list`, `regression`); default `summ`
`METHOD` = *string token*	Imputation method (`hotdeck`, `modelbased`); default `hotd`
`DMETHOD` = *string token*	Method for calculating distances (`mean`, `minimax`, `regression`); defaule `mini`
`%THRESHOLD` = *scalar*	Percentage threshold for matches
`THRESHOLD` = *scalar*	Absolute threshold for matches
`DVARIABLES` = *variates* or *factors*	Variables to use for distance calculation or factors
`DRANGES` = *scalars*	Ranges to use for distance calculations with each of the `DVARIABLES`; default `*` uses the observed range
`LABELS` = *variate*, *factor* or *text*	Provides labels for the cases
`SEED` = *scalar*	Seed for random numbers; default 0
`IMPUTE` = *variate* or *scalar*	The variate provides logical (0 or 1) values to indicate whether each unit is to be imputed, alternatively the scalar specifies a number of rows to be selected at random to be imputed to allow the effectiveness of the imputation process to be studied; default `*` imputes values for any units where an `OLDSTRUCTURE` contains a missing value
`DONORS` = *variate*	Logical variate indicating whether each unit can be used as a donor; default `*` implies that all units are used with complete data for each `OLDSTRUCTURE`
`RSAVE` = *rsave*	Regression analysis to use for `METHOD=model` or `DMETHOD=regression`
`URECEPTORS` = *variate*	Saves unit numbers of receptor (imputed) cases
`UDONORS` = *variate*	Saves unit numbers of donor cases
`DISTANCES` = *variate*	Saves the distances for the chosen receptor-donor pairs

Parameters

`OLDSTRUCTURE` = *variates* or *factors*	Structure containing missing values
`NEWSTRUCTURE` = *variates* or *factors*	New structures with imputed values
`OVERWRITE` = *string tokens*	Whether to overwrite any existing data for imputed cases (`yes`, `no`); default `no`

Description

Survey data frequently contain missing values. When all the information is missing for a sample unit it is generally appropriate to allow for this by modifying the weights, but when only certain variables are missing (item non-response) imputation is often used to fill in the missing values. `SVHOTDECK` performs "hot-deck" imputation (see for example Korn & Graubard 1998) whereby replacement values are taken from another unit, chosen at random, usually from a list of suitable matches determined on the basis of a suitable distance metric. The procedure can also be used for model-based imputation; in this case the imputed value is taken as the sum of the fitted value from a regression model and a residual chosen at random from another unit. In the description below "donor" is used to mean a unit supplying data to a "receptor" that has a missing value

initially.

The data are usually supplied by the OLDSTRUCTURE parameter, in variates and/or factors, containing missing values. The NEWSTRUCTURE parameter supplies new variates or factors to contain the values of each OLDSTRUCTURE variate or factor, but with the missing values replaced by the imputed values. By default, imputation is carried out for any row of data where an OLDSTRUCTURE contains missing values. Alternatively, the rows to be imputed can be specified by setting option IMPUTE. This can supply a logical variate, containing the value one in the units whose values are to be imputed, and zero elsewhere, or it can supply a scalar specifying a number of rows to be selected at random to be imputed. The scalar setting is useful if you want to study the effectiveness of the imputation process.

By default, imputed values will be used only to replace the missing values in each OLDSTRUCTURE, unless the corresponding setting of the OVERWRITE parameter is yes. Imputed values are then inserted even if the original value is not missing. This would allow you, for example, to compare real and imputed data in order to check the efficiency of the imputation process. Alternatively, you might set OVERWRITE=yes for every OLDSTRUCTURE in order to preserve the correlations between the variables by taking all the values from each donor.

By default, any row of OLDSTRUCTURE with no missing values may be used as a donor, unless option DONORS is used to specify a logical variate to indicate the rows that are to act as potential donors.

The DVARIABLES option is used to supply one of more variables to use to determine the matching between donors and receptors. In the simplest case, if you set DVARIABLES to a single factor, the donors are selected at random from receptors with the same factor value (e.g. to replace observations by others from the same stratum). For more complex matching, DVARIABLES can be set to a list of variates or factors which are then used to determine a distance between each receptor and the potential donors. By default the distance for a DVARIABLES variate is calculated as

$$d = |x_i - x_j| / r$$

where r is the observed range of the data, but an alternative value of r may be supplied using the DRANGES option. DRANGES should be set to 1 if no scaling of the distances is required. For a DVARIABLES factor a simple matching criterion is used, so $d = 0$ if x_i and x_j are the same, and $d = 1$ if they are not.

Matches are then determined using these distances according to a "minimax" approach, where the best match is the one with the minimum value of the maximum absolute difference between any of the DVARIABLES. Alternatively you can set the DMETHOD option to mean to use the mean of the absolute differences, or to regression to request that the distances are determined on the basis of predictions from a regression.

The RSAVE option specifies the regression analysis to use when DMETHOD=regression. The terms in the model must include the DVARIABLES. If RSAVE is not specified, the most recent regression analysis is used. The calculation of the distances between units is then weighted by the appropriate regression coefficients: for example, if the slope of x1 is 0.24 and two units have x1 values of 10 and 20, the distance is

$$(20 - 10) \times 0.24 = 2.4.$$

DRANGES are ignored when DMETHOD=regression.

Conventional hot-deck imputation is the default method. Alternatively, if you set option METHOD=modelbased, SVHOTDECK will do model-based imputation. Note, though, that this cannot be used if DMETHOD=regression. Model-based imputation uses a regression analysis, specified by the RSAVE option. If RSAVE is not specified, the most recent regression analysis is used. The method creates an imputed value by adding a random residual to the fitted value of the selected donor. This method can be used only if the OLDSTRUCTURE is the same as the y-variate in the regression. DVARIABLES will frequently be left unset in this situation, so that the residuals are chosen totally at random. However, in some situations it may be preferable to select residuals

from similar units, in which case DVARIABLES can be used to determine the matching, as with the hot-deck method.

By default, SVHOTDECK will determine the single best match for each unit, where possible. In many cases (e.g. when doing multiple imputation), it is required to select one at random from the closest matches. The %THRESHOLD option specifies the tolerance to use in these situations: for example, setting %THRESHOLD to 10 requests that the match is selected at random from amongst the donors with distance up to 10% greater than the minimum distance. The SEED option specifies the seed for the random numbers that are used for this operation (default 0). Alternatively, if it is desired to specify the distance relative to the minimum in absolute terms, the THRESHOLD option should be used instead. If both THRESHOLD and %THRESHOLD are set, both criteria must be met. The THRESHOLD value is normally set relative to the minimum distance, but, if it is set to a negative value this is taken to mean that a match is selected at random from those with a distance less than the absolute value of the THRESHOLD. Thus, for example, if THRESHOLD is set to −0.2 and METHOD=mean, any units with a mean distance of less than 0.2 (after taking into account settings of DRANGES) from the unit to be imputed are considered matches, and one of these is selected at random. Alternatively, if THESHOLD is set to 0.2 and the best match is for example 0.18, any units with a mean distance of less than 0.18 + 0.2 = 0.38 are considered matches, and one of these is selected at random.

The URECEPTORS and UDONORS options can be used to save the unit numbers of the receptor (imputed) cases and the donor cases, respectively. Note that, if the IMPUTE option is set, the OLDSTRUCTURE and NEWSTRUCTURE parameters need not be set. The use of URECEPTORS and UDONORS then allows more complicated methods of replacement to be used than those provided directly by SVHOTDECK.

Printed output and plots are controlled by the PRINT option, with the settings:

monitoring	provides information about each match,
summary	provides a summary,
list	produces a list of recipients and donors,
check	prints correlations as well as giving a scatter plot of the predictions against the actual data, and
regression	gives details of the model used when DMETHOD is set to regression.

To use check it is necessary to impute for data values that are present. This can be achieved either by specifying these units using IMPUTE, or by setting IMPUTE to a scalar, in which case the appropriate number of rows will be selected at random.

Options: PRINT, METHOD, DMETHOD, %THRESHOLD, THRESHOLD, DVARIABLES, DRANGES, LABELS, SEED, IMPUTE, DONORS, RSAVE, URECEPTORS, UDONORS, DISTANCE.
Parameters: OLDSTRUCTURE, NEWSTRUCTURE, OVERWRITE.

Action with RESTRICT

SVHOTDECK takes restrictions from any OLDSTRUCTURE or DVARIABLES vectors. Only unrestricted units are used as either donors or receptors. However, restrictions on IMPUTE and DONORS are ignored.

References

Korn, E.L. & Graubard, B.I. (1999). *Analysis of Health Surveys*. Wiley, New York.

See also

Procedures: SVBOOT, SVCALIBRATE, SVGLM, SVREWEIGHT, SVSAMPLE, SVSTRATIFIED, SVTABULATE, SVWEIGHT, MULTMISSING, QMVREPLACE.

Genstat Reference Manual 1 Summary section on: Survey analysis.

SVMERGE

Merges strata prior to survey analysis (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (summary, intable, outtable, twowaytable); default summ
OLDFACTOR = *factor*	Factor defining the original strata
NEWFACTOR = *factor*	Factor to save the merged strata

Parameters

MERGELABELS = *texts*	Labels of strata to merge
NEWLABEL = *text*s	Label for merged stratum

Description

In survey analysis it is often necessary to combine a number of strata for analysis, for example where less data are collected than anticipated. SVMERGE does this, identifying strata using their labels, avoiding the errors that can arise when using other methods, such as the NEWLEVELS function.

The original stratum factor is specified using the OLDFACTOR option, and the corresponding new factor is saved using the NEWFACTOR option. The MERGELABELS parameter supplies a text containing the two or more stratum labels to merge from the OLDFACTOR. The NEWLABEL parameter supplies the name of the merged stratum for the NEWFACTOR. The NEWLABEL can be set to an existing label of OLDFACTOR, but only if it is also included in the list of MERGELABELS.

Printed output is controlled by the PRINT option, using the following settings:

summary	summary of merges,
intable	table of counts for original factor,
outtable	table of counts for merged factor, and
twoway	two-way table of counts for original and merged factors.

By default, the summary is printed.

Options: PRINT, OLDFACTOR, NEWFACTOR.
Parameters: MERGELABELS, NEWLABEL.

Method

SVMERGE uses the FACMERGE procedure.

Action with RESTRICT

The merging process ignores any restrictions but, when this has been completed, any restriction on OLDFACTOR is applied to NEWFACTOR.

See also

Procedures: SVSTRATIFIED, SVTABULATE, FACMERGE.
Genstat Reference Manual 1 Summary section on: Survey analysis.

SVMFIT

Fits a support vector machine (D. B. Baird).

Options

PRINT = *string tokens*	Printed output from the analysis (summary, predictions, allocations, debug); default summ, alloc
SVMTYPE = *string token*	Type of support vector machine to fit (svc, svr, nusvc, nusvr, lsvc, lsvr, lcs, svm1); default svc
KERNEL = *string token*	Type of kernel to use (linear, polynomial, radialbasis, sigmoid); default radi
PENALTY = *scalar* or *variate*	Penalty or cost for points on the wrong side of the boundary; default 1
GAMMA = *scalar* or *variate*	Gamma parameter for types with non-linear kernels; default 1
NU = *scalar* or *variate*	Nu parameter for types nusvc, nusvr, and svm1; default 0.5
EPSILON = *scalar* or *variate*	Epsilon parameter for types svr and lsvr; default 0.1
BIAS = *scalar*	Bias for allocations to groups for types lsvc and lsvr; default –1 i.e. no bias
DEGREE = *scalar*	Degree for polynomial kernel; default 3
CONSTANTVALUE = *scalar*	Constant for polynomial or sigmoid kernel; default 0
LOWER = *scalar* or *variate*	Lower limit for scaling data variates; default –1
UPPER = *scalar* or *variate*	Upper limit for scaling data variates; default 1
SCALING = *string token*	Type of scaling to use (none, uniform, given); default unif
NOSHRINK =*string token*	Whether to suppress the shrinkage of attributes to exclude unused ones (no, yes); default no
OPTMETHOD =*string token*	Whether to optimize probabilities or allocations (allocations, probabilities); default allo
REGULARIZATIONMETHOD = *string token*	Regularization method for SVMTYPE = lsvc or lsvr (l1, l2); default l2
LOSSMETHOD = *string token*	Loss method for SVMTYPE = lsvc or lsvr (logistic, l1, l2); default logi
DUALMETHOD = *string token*	Whether to use the dual algorithm for SVMTYPE = lsvc or lsvr (yes, no); default no
NCROSSVALIDATIONGROUPS = *scalar*	Number of groups for cross-validation; default 10
SEED = *scalar*	Seed for random number generation; default 0
TOLERANCE = *scalar*	Tolerance for termination criterion; default 0.001
WORKSPACE = *scalar*	Size of workspace needed for data; default is to calculate this from the number of observations and variates

Parameters

Y = *factors or variates*	Define groupings for the units in each training set y-variate to be predicted via regression, with missing values in the units to be allocated or predicted
X = *pointers*	Each pointer contains a set of explanatory variates or factors
WEIGHTS = *variates*	Weights to multiply penalties for each group when

	SVMTYPE = svc, nusvc, lsvc or lcs
PREDICTIONS = *factors or variates*	
	Saves allocations to groups or predictions from regression
ERRORRATE = *scalars, variates* or *matrices*	
	Saves the error rate for the combinations of parameters specified for the support vector machine
OPTPENALTY = *scalars*	Saves the optimal value of penalty parameter
OPTGAMMA = *scalars*	Saves the optimal value of gamma parameter
OPTNU = *scalars*	Saves the optimal value of nu parameter
OPTEPSILON = *scalars*	Saves the optimal value of epsilon parameter
OPTERRORRATE = *scalars*	Saves the minimum error rate
SCALE = *texts* or *pointers*	Saves the scaling used for the X variates, in a file if a text is given, or otherwise in a pointer to a pair of variates
SAVEFILE = *texts*	File in which to save the model, for use by SVMPREDICT

Description

SVMFIT fits a support vector machine (Cortes & Vapnik 1995), which defines multivariate boundaries to separate groups, or predict values. It provides a Genstat interface to the libraries LIBSVM (Chang & Lin 2001) and LIBLINEAR (Fan *et al.* 2008), which are made available subject to the conditions listed in the Method section.

Unlike linear discriminant analysis, a support vector machine assumes no statistical model for the distribution of individuals within a group. The method is thus less affected by outliers. The method chooses boundaries to maximize the separation between groups. The reason why this is known as a support vector machine, is that there is a small set of data points that define the boundaries, and these are known as the support vectors. If individuals lie on the wrong side of the boundary, the distance from the boundary, multiplied by a penalty, is added to the separation criterion.

The type of support vector machine to fit is specified by the SVMTYPE option, with settings:

svc	a multi-class support vector classifier with a range of kernels for discriminating between groups;
svr	support vector regression with a range of kernels for predicting the values of a y-variate as in a regression;
nusvc	Nu classification – a multi-class support vector classifier with a range of kernels for discriminating between groups with a parameter NU that controls the fraction of support vectors used;
nusvr	Nu regression – support vector regression with a range of kernels for predicting the values of a y-variate as in a regression with a parameter NU that controls the fraction of support vectors used;
lsvc	Fast linear classification – a fast regularized linear support vector for discriminating between groups;
lsvr	Fast linear regression - a fast regularized linear support vector regression for predicting the values of a y-variate as in a regression;
lcs	a fast linear support vector machine for discriminating between groups using the approach of Cramer & Singer (2000), where a direct method for training multi-class predictors is used, rather than dividing the multi-class

	classification into a set of binary classifications; and
`svm1`	Consistent group SVM – a support vector machine which attempts to identify a consistent group of observations.

The shape of the boundary is controlled by the `KERNEL` option which specifies the metric used to measure distance between multi-dimensional points u and v. The settings are:

`linear`	the linear function $u'v$;
`polynomial`	the polynomial function $\gamma\ (u'v + c)^d$;
`radialbasis`	the radial basis function $\exp(-\gamma\ \|u - v\|^2)$; and
`sigmoid`	the sigmoid function $\tanh(\gamma\ u'v + c)$.

With a linear kernel, the boundaries are multi-dimensional planes. For the other types they are curved surfaces. The kernel is ignored for `SVMTYPE=lsvc`, `lsvr` and `lcs` as these always use a linear kernel.

The data set is supplied in a pointer of explanatory variates or factors, specified by the `X` parameter, and a response variate or factor specified by the `Y` parameter. The `Y` parameter need not be set if `SVMTYPE=svm1`, as this searches for a consistent group of individuals in the data set, ignoring the `Y` parameter. Explanatory factors are converted to variates, using the levels of the factor concerned. Any unit with a missing value in an explanatory variate takes a zero value for that attribute. With the default, uniform, scaling this puts them in the centre of the range of the variate concerned. Units can also be excluded from the analysis by restricting the factor or variates; any such restrictions must be consistent.

The response factor specifies the pre-defined groupings of the units from which the allocation is derived (the "training set"); the units to be allocated by the analysis have missing values for `Y`. A response variate supplies training values for a regression-type support vector machine. (These are requested by `SVMTYPE` settings `svr`, `nusvr` and `lsvr`.) Units to be predicted by the regression have missing values in the y-variate.

The support vector machine solutions depend on the scale of the attributes. It is usually recommended that all attributes are put on the same scale, so that they all have the same influence. This is controlled by the `SCALING` option, with settings:

`none`	the attributes are used as supplied, with no scaling;
`uniform`	all the attributes are centred, and scaled to have the same minimum and maximum (default); and
`given`	the variates are scaled using the `LOWER` and `UPPER` options.

The `LOWER` and `UPPER` options can be set to a scalar, to apply a uniform scaling, where all the variates are given the same minimum (`LOWER`) and maximum (`UPPER`) value; alternatively, they can be variates specifying the minimum and maximum value for each variate, respectively.

The `PENALTY` option defines the penalty that is applied to the sum of distances for the points on the wrong side of the boundary when calculating the optimal boundaries; default 1. Larger values apply more weight to points that are on the wrong side of the discrimination boundaries, and can be investigated to optimize performance. However, linear support vector machines are generally insensitive to the choice of the penalty. The `WEIGHTS` parameter can be used to change the penalty for mis-assigning a case to a particular group, and should be a variate with the same length as the number of levels in `Y`. The penalty for each group is then corresponding value of `PENALTY*WEIGHTS`.

The `GAMMA` option (γ in the equations for the kernels) controls the smoothness of the boundary for non-linear kernels, with larger values giving a rougher surface.

With `SVMTYPE=nusvc` and `nusvr`, the parameter `NU` controls the number of support vectors used; default 0.5. With larger values of `NU`, smaller numbers of support vectors are used, giving a sparser solution that may be more robust and thus perform better in future prediction.

With the regression cases `SVMTYPE=svr` and `lsvr`, the parameter `EPSILON` controls the sensitivity of the loss function being optimized; default 0.1. A range of parameter values for `PENALTY`, `GAMMA`, `NU` or `EPSILON` are usually tried, to optimize the discrimination between

groups or predictions of the y-variate. These parameters also accept a variate, in which case all the values in the variate are tried and the one that minimizes the error rate is selected. Up to two of these parameters can be variates at once. A grid of error rates is then calculated using every combination of the two sets of parameters, and the optimal combination is selected. If three or more of these parameters are set to variates, a warning is given, and only the first values of the third and fourth variates are selected.

When `KERNEL=polynomial`, the `DEGREE` option defines the degree of the polynomial (*d* in the equation for the polynomial kernel). The `CONSTANTVALUE` option gives the constant (*c* in the equations for the kernels), for `KERNEL=polynomial` and `sigmoid`.

The `TOLERANCE` option supplies a small positive value that controls the precision used for the termination criterion. Decreasing this may provide a better solution, but will increase the time taken until convergence.

The `NOSHRINK` option controls whether unnecessary attributes are dropped from the fitting process; by default, these are dropped, thus increasing the speed to find a solution when there are many iterations (e.g. when `TOLERANCE` has been made smaller). If few iterations are required to find a solution, it may be faster to set `NOSHRINK=yes`.

The `OPTMETHOD` option controls the criterion that is optimized when the `SVMTYPE` is set to `svc`, `svr`, `nusvc` or `nusvr`, with settings:

`allocations`	for the accuracy of allocating individuals to groups; or
`probabilities`	for sum of the probabilities of allocating an individual to the correct group.

The `SYMTYPE`s `lsvc`, `lsvr` and `lcs` fit regularized linear support vector machines using the algorithms in the LIBLINEAR library of Fan *et al.* (2008). This is much faster than the default algorithm, allowing much bigger data sets to be analysed. The `REGULARIZATIONMETHOD`, `LOSSMETHOD` and `DUALMETHOD` options specify which LIBLINEAR algorithm is used for `SYMTYPE`s `lsvc` and `lsvr`.

The `REGULARIZATIONMETHOD` option allows you to create sparser sets of support vectors, with the `L1` setting giving a smaller set of support vectors than `L2`. The `LOSSMETHOD` option controls the loss function being minimized: the `L2` setting minimizes the sum of the squared distances of points on the wrong side of the boundary, the `L1` setting minimizes the sum of the distances, and the `logistic` setting uses a logistic regression loss function. Setting option `DUALMETHOD=yes` may be faster when there are a large number of attributes. Not all combinations of `REGULARIZATIONMETHOD`, `LOSSMETHOD` and `DUALMETHOD` options are available.

When `SVMTYPE=lsvc`, you can use the `BIAS` option to attempt to achieve a more optimal discrimination between groups. When `BIAS` is set to a non-negative value, an extra constant attribute is added to the end of each individual. This extra attribute is given a weight that controls the origin of the separating hyper-plane (the origin is where all attributes have value of 0). A `BIAS` of 0 forces the separating hyper-plane to go through the origin, and a non-zero value moves the plane away from the origin. The `BIAS` thus acts as a tuning parameter, that changes the hyper-plane's origin. A range of values can be investigated, to try to improve the discrimination.

Printed output is controlled by the option `PRINT` with settings:

`summary`	tables giving the number of units in each group with a complete set of observations;
`allocations`	tables of counts of allocations; and
`debug`	details of the parameters set when calling the libraries.

The error rate is worked out by cross-validation, which works by randomly splitting the units into a number of groups specified by the `NCROSSVALIDATIONGROUPS` option. It then omits each of the groups, in turn, and predicts how the omitted units are allocated to the discrimination groups.

The `SEED` option provides the seed for the random numbers used for allocating individuals to

the cross-validation groups. The default value of 0 continues an existing sequence of random numbers. If none have been used in the current Genstat job, it initializes the seed automatically using the computer clock.

The WORKSPACE option can be set if the problem requires more memory than the default settings.

Results from the analysis can be saved using the parameters PREDICTIONS, ERRORRATE, OPTPENALTY, OPTGAMMA, OPTNU, OPTEPSILON and OPTERRORRATE. The structures specified for these parameters need not be declared in advance. If one of the options PENALTY, GAMMA, NU or EPSILON has been set to a variate, ERRORRATE will be a variate indexed by that variate. Alternatively, if two of these options have been set to variates, ERRORRATE will be a matrix with rows and columns indexed by those variates. The OPT parameters contain the values of the parameters, that give the minimum error rate (returned in OPTERRORRATE).

The support vector machine model can be saved in an external file, using the SAVEFILE parameter, so that it can be used later with SVMPREDICT. As the scaling on the attributes must be the same in future data sets, the scaling can be saved with the SCALE parameter. This can supply either a filename (ending in .gsh) to keep these permanently, or a pointer so that these can be applied to the attributes used in SVMPREDICT later in the same program. The file or pointer contains two variates, which give the slope and intercept (in that order) for the linear transform applied to each attribute.

Options: PRINT, SVMTYPE, KERNEL, PENALTY, GAMMA, NU, EPSILON, BIAS, DEGREE, CONSTANTVALUE, LOWER, UPPER, SCALING, NOSHRINK, OPTMETHOD, REGULARIZATIONMETHOD, LOSSMETHOD, DUALMETHOD, NCROSSVALIDATIONGROUPS, SEED, TOLERANCE, WORKSPACE.

Parameters: Y, X, WEIGHTS, PREDICTIONS, ERRORRATE, OPTPENALTY, OPTGAMMA, OPTNU, OPTEPSILON, OPTERRORRATE, SCALE, SAVEFILE.

Method

SVMFIT provides a Genstat interface to the C++ libraries LIBSVM (Chang & Lin 2001) and LIBLINEAR (Fan *et al.* 2008), that have been compiled into the GenSVM dynamic link library. A user guide by Hsu *et al.* (2003) gives details on their use.

LIBSVM is provided subject to the following copyright notice.

possibility of such damage.

LIBLINEAR is provided subject to the following copyright notice.

Action with RESTRICT

The input variates and factor may be restricted. The restrictions must be identical.

References

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**, 273-297. URL: http://link.springer.com/article/10.1007%2FBF00994018

Chang, C.C. & Lin, C.J. (2001). LIBSVM: A library for support vector machines. URL: http://www.csie.ntu.edu.tw/~cjlin/libsvm

Cramer, K. & Singer, Y. (2000). On learnability and design of output codes for multi-class problems. In *Computational Learning Theory*, 35-46.

Fan, R.E., Chang, K.W, Hsieh, X.R., Wang, X.R. & Lin C.J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, **9**, 1871-1874. URL: http://www.csie.ntu.edu.tw/~cjlin/papers/liblinear.pdf

Hsu, C.W., Chang, C.C. & Lin, C.J. (2003). A practical guide to support vector classification. (Technical report). Department of Computer Science and Information Engineering, National Taiwan University. URL: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

See also

Directive: CVA.

Procedures: SVMPREDICT, DISCRIMINATE, QDISCRIMINATE, SDISCRIMINATE.

SVMPREDICT

Forms the predictions using a support vector machine (D. B. Baird).

Options

SCALE = *texts* or *pointers*	Gives scaling used for the X variates
SAVEFILE = *texts*	Gives support vector machine model file; default is to use the model from the last support vector machine

Parameters

X = *pointers*	Each pointer contains a set of variates defining the attributes for the predictions
PREDICTIONS = *factors* or *variates*	Saves the classification groupings or predicted values for each observation in X
GROUPDEFINITIONS = *factors*	Supplies levels and labels for predicted groups; default uses ordinal levels

Description

SVMPREDICT forms predictions using a support vector machine (Cortes & Vapnik 1995) fitted by SVMFIT. The input for the procedure is given by a pointer specified by the X parameter. The X pointer contains a set of variates and factors defining the attributes of the units. Any unit with a missing value in any of the variates is taken as having a central value for that attribute.

The PREDICTIONS parameter returns the predictions in a factor for a classification support vector machine, or a variate for a regression support vector machine. The GROUPDEFINITIONS parameter can be used to specify the levels and labels for factor predictions. If GROUPDEFINITIONS is unset, the ordinal levels 1...*n* are used.

The SAVEFILE option specifies the model file for a support vector machine saved by SVMFIT. If this is not specified, the last fitted model created by SVMFIT will be used.

The SCALE option gives the scaling used for the attributes. This can either be in a file saved by SVMFIT, or in a pointer containing two variates with the same length as the X pointer. The variates give the *slope* and the *constant*, respectively, used to scale each X variate. The scaled variates are then

slope × X[] + *constant*

If SCALE is not specified, the X variates will not be scaled. Note, though, that the attributes in X must be on the same scale as that used in SVMFIT for the predictions to be correct.

Options: SCALE, SAVEFILE.
Parameters: X, PREDICTIONS, GROUPDEFINITIONS.

Method

The C++ libraries LIBSVM (Chang and Lin, 2001) and LIBLINEAR (Fan *et al.*, 2008) have been compiled into the library GenSVM.DLL, and this is called from SVMPREDICT. A user guide by Hsu *et al.* (2003) gives details on using this software.

Action with RESTRICT

The input variates and factor may be restricted. The restrictions must be identical.

References

Cortes, C. & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**, 273-297.
URL: http://link.springer.com/article/10.1007%2FBF00994018

Chang, C.C. & Lin, C.J. (2001). LIBSVM: A library for support vector machines.
URL: http://www.csie.ntu.edu..tw/~cjlin/libsvm

Fan, R.E., Chang, K.W., Hsieh, X.R., Wang X.R., & Lin C.J. (2008). LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, **9**, 1871-1874. URL: http://www.csie.ntu.edu.tw/~cjlin/papers/liblinear.pdf

Hsu, C.W., Chang, C.C., & Lin, C.J. (2003). A practical guide to support vector classification. (Technical report). Department of Computer Science and Information Engineering, National Taiwan University. URL: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf

See also

Directive: CVA.

Procedures: SVMFIT, DISCRIMINATE, QDISCRIMINATE, SDISCRIMINATE.

SVREWEIGHT

Modifies survey weights for particular observations, adjusting other weights in the sampling unit or stratum to ensure that the overall sum of the weights remains unchanged (S.D. Langton).

Options

PRINT = *string tokens*	Controls printed output (summary); default summ
METHOD = *string tokens*	What to reweight over (all, stratum, samplingunits, lowest); default lowe
WEIGHTS = *variate*	Initial weights
OUTWEIGHTS = *variate*	Final weights
STRATUMFACTOR = *factor*	Stratification factor; default * i.e. unstratified
OUTSTRATUMFACTOR = *factor*	Saves a modified stratification factor with the reweighted observations in their own stratum
SAMPLINGUNITS = *factor*	Factor indicating the primary sampling units; default *, i.e. single stage design
LABELS = *variate*, *text* or *factor*	Labels for each unit

Parameters

OBSERVATIONS = *scalars*, *variates* or *texts*	Observations to reweight
NEWWEIGHTS = *scalars* or *variates*	New weights (default inserts a missing value, indicating that the observation should be removed)

Description

Item non-response (i.e. a missing value for one question although valid responses are present for others) and outliers are two common problems in survey data. If the item response occurs entirely at random, one method of dealing with it is to analyse the question with a modified set of weights with the weight for the missing observation redistributed over the rest of the units in the stratum or sampling unit. This could be achieved by calculating the weights again from scratch, but it is often preferable to modify the existing weights variable. Similarly if the influence of outliers is reduced by giving them a reduced weight (see Lee 1995 for a discussion of this subject), the weights for the remaining observations must be adjusted to maintain the same sum of weights.

The units whose weights are to be adjusted are specified by the OBSERVATIONS parameter. They may be specified in two different ways:

1) A list of observations whose weights need modifying may be supplied in one or more variates, scalars or texts. By default the units are identified by the unit number of the observation, but, if the LABELS option is set to a variate, factor or text, the values are matched against the labels. Multiple observations can be specified either as a list of scalars (or single-valued texts if appropriate), or by variates or texts with multiple values.
2) A variate of the same length as WEIGHTS may be supplied, with the value one in the units whose weights need to be modifed. Other units should contain zeros.

By default the procedure assumes that the observations should have their weight set to missing so that they are excluded from analysis by TABULATE or SVTABULATE. Alternatively NEWWEIGHTS can be used to specify the required weights to insert. This can be set to a scalar if the same weight is to be used for every unit specified by the corresponding OBSERVATIONS variate, text or scalar. Alternatively, it can be set to a variate of the same length as the corresponding OBSERVATIONS setting.

The METHOD option specifies the level at which the weights are redistributed, so that, for example, setting METHOD=stratum changes the other weights in the stratum containing the

observation so that their total remains unchanged. If METHOD is unset the procedure works to the lowest specified level, i.e. sampling units if these are specified, or otherwise the strata. If the stratification factor is also unspecified the redistribution takes place over all other observations.

Where reduced weights (typically 1.0) are allocated to outliers because they are genuine but not representative of the wider population, these units are often placed in their own stratum; the OUTSTRATUMFACTOR option can be used to create such a suitable stratification factor.

Options: PRINT, METHOD, WEIGHTS, OUTWEIGHTS, STRATUMFACTOR, OUTSTRATUMFACTOR, SAMPLINGUNITS, LABELS.
Parameters: OBSERVATIONS, NEWWEIGHTS.

Action with RESTRICT
Any restrictions are ignored.

References
Lee, H. (1995). Outliers in Business Surveys. Chapter 26 of *Business Survey Methods* (ed. Cox, Binder, Hinnappa, Christianson, Colledge & Kott). Wiley, New York.

See also
Procedures: SVBOOT, SVCALIBRATE, SVGLM, SVHOTDECK, SVSAMPLE, SVSTRATIFIED, SVTABULATE, SVWEIGHT.
Genstat Reference Manual 1 Summary section on: Survey analysis.

SVSAMPLE

Constructs stratified random samples (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (list, summary); default summ
SAMPLE = *variate*	Saves the sample, as unit numbers of sampled units when METHOD=sample, or as a logical (1 or 0) variable indicating sampled or unsampled units when METHOD=population
STRATUMFACTOR = *factor*	Saves the stratification factor
CLUSTERS = *factor*	Specifies a factor indicating groupings of units for a cluster sample; default * i.e. sample individual rows
NUNITS = *table*, *scalar* or *variate*	Numbers of units in the full data set for each level of the STRATUMFACTOR
NSAMPLE = *table*, *scalar* or *variate*	Numbers, or proportions, of units to sample for each level of the STRATUMFACTOR
SFLEVELS = *variate*	Levels for the stratum factor, if it has not already been declared
SFLABELS = *text*	Labels for the stratum factor, if it has not already been declared
METHOD = *string token*	Whether SAMPLE should contain the numbers of the units sampled from the population, or be a variate with a value for every unit of the full population containing 0 or 1 for unsampled and sampled units respectively (population, sample); default samp
NUMBERING = *string token*	Whether to number units within each stratum, or across the whole population (withinstratum, population); default with
SEED = *scalar*	Seed for the random number generator; default 0 i.e. continue from previous generation

Parameters

OLDVECTOR = *variates*, *factors* or *texts*	Data from the full survey
NEWVECTOR = *variates*, *factors* or *texts*	Data for the sample

Description

SVSAMPLE forms random samples for stratified random surveys. It can also be used to construct a new dataset containing only the sampled units. Groups of units can be sampled together, to allow cluster or multistage sampling.

SVSAMPLE is easiest to use when the vectors (variates, factors or texts) representing all the units in the population have already been created. For a single-stage sample, the number of units to be sampled is then specified by setting the NSAMPLE option to a table classified by the stratification factor; if the values of NSAMPLE are all less than 1, these are taken to be proportions to sample. NSAMPLE can be set to a scalar for unstratified samples. By default, individual units are sampled, but the CLUSTERS option can supply a factor to define clusters of units that are to be sampled together.

The sample can be saved, in a variate, by the SAMPLE parameter. If the METHOD option is set to its default setting of sample, SAMPLE will contain the numbers of the sampled units. By default, the units are numbered separately within each stratum, but you can set option

NUMBERING=population to number the units across the whole population. Alternatively, if option METHOD=population, the SAMPLE variate will have a value for every unit in the population; this stores the value one for the sampled units and zero for the unsampled units. The STRATUMFACTOR option can save a factor indicating the stratum to which each sampled unit belongs.

Printed output is controlled by the PRINT option, with settings:

list	to list the sampled units, and
summary	to give a summary of the units sampled from each stratum.

If you already have vectors containing the full data set, you can use SVSAMPLE to create a new data set containing only the sampled units. The OLDVECTOR parameter supplies the vectors from the full data set, and the NEWVECTOR parameter saves vectors with only the samples units. In a stratified survey, you may supply original vectors that have only one value for each stratum. Each corresponding NEWVECTOR then takes the appropriate values corresponding to the STRATUMFACTOR levels to which the sampled units belong.

If you do not already have the vectors for the full population, you must supply the information to create them using options of SVSAMPLE. This is primarily intended for the situation where details of the strata are read into Genstat from a spreadsheet. The SFLABELS and SFLEVELS options define the labels and levels of the STRATUMFACTOR, respectively. The NUNITS option specifies the corresponding number of primary sampling units in the full population.

SVSAMPLE can be used to construct multistage samples. In the first stage of sampling, NSAMPLE should have one value for each stratum. For the next stage of sampling a second SVSAMPLE command should be given, with NSAMPLE now having one value for each of the sampling units from the first stage. This process can be repeated, as required, for samples with more than two stages.

Options: PRINT, SAMPLE, STRATUMFACTOR, CLUSTERS, NUNITS, NSAMPLE, SFLEVELS, SFLABELS, METHOD, NUMBERING, SEED.
Parameters: OLDVECTOR, NEWVECTOR.

Action with RESTRICT

If NSAMPLE supplies a table, then any restriction on the classifying factor is taken to indicate units to be excluded from the random sampling process. This may be useful, for example, in a social survey where some people have previously indicated that they do not wish to take part in the survey, or in an ecological survey where some sites are inaccessible. Otherwise, any restrictions on the input vectors are ignored.

See also

Procedures: SVBOOT, SVCALIBRATE, SVGLM, SVHOTDECK, SVREWEIGHT, SVSTRATIFIED, SVTABULATE, SVWEIGHT, SAMPLE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Survey analysis.

SVSTRATIFIED

Analyses stratified random surveys by expansion or ratio raising (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (summary, totals, means, influence, ratios, extra); default summ, tota, infl
PLOT = *string token*	Controls which high-resolution graphs are plotted (single, separate); default * i.e. none
XMISSING = *string token*	Action if x-variable contains missing values (estimate, fault); default esti
RESTRICTED = *string token*	Action with restricted (or filtered) observations (omit, add); default omit
STRATUMFACTOR = *factor*	Stratification factor; default * i.e. unstratified
NINFLUENCE = *scalar*	Number of influential points to print; default 10
METHOD = *string token*	Method for ratio analysis (separate, combined, classicalcombined); default sepa
SAVESUMMARY = *string token*	Whether to save just the overall summaries instead of those for each stratum (yes, no); default no
COMBINEDSTRATUM = *scalar*	Stratum for which the ratio should be set to the combined ratio estimate; default *
ROWS = *scalars*	Number of rows of plot-matrix; default * i.e. set automatically depending on number of levels of STRATUMFACTOR
COLUMNS = *scalars*	Number of columns of plot-matrix; default * i.e. set automatically depending on number of levels of STRATUMFACTOR
NBOOT = *scalar*	Number of bootstrap samples to use; default 0
SEED = *scalar*	Seed for random number generator for bootstrap; default 0
CIPROBABILITY = *scalars*	The probability level for the confidence intervals; default 0.95
CIMETHOD = *string token*	Method for forming confidence intervals (automatic, tdistribution, percentile); default auto
COMPACT = *string token*	Whether to produce output in a compact (plaintext) format (yes, no); default no

Parameters

Y = *variates*	Response data
X = *variates*	Base data; if unset expansion raising is used
LABELS = *variates*, *factors* or *texts*	Structure for labelling influential points
NUNITS = *tables*, *scalars* or *variates*	Numbers of units in each stratum in the population
XTOTALS = *tables*, *scalars* or *variates*	Population totals of the base data in each stratum
TOTALS = *tables* or *scalars*	Saves total estimates
SETOTALS = *tables* or *scalars*	Saves standard errors of estimates
MEANS = *tables* or *scalars*	Saves mean estimates
SEMEANS = *tables* or *scalars*	Saves standard errors of mean estimates
RATIOS = *tables*	Saves estimates of ratios
FITTEDVALUES = *variates*	Saves fitted values for the observations
INFLUENCE = *variates*	Saves influence statistics

LTOTALS = *tables* or *scalars*	Saves lower confidence limit for total
UTOTALS = *tables* or *scalars*	Saves upper confidence limit for total
LMEANS = *tables* or *scalars*	Saves lower confidence limit for mean
UMEANS = *tables* or *scalars*	Saves upper confidence limit for mean
VARIANCES = *tables* or *scalars*	Saves residual variances in each stratum

Description

SVSTRATIFIED analyses the results from a stratified random survey, either by expansion or ratio raising, and allows detection of outliers. The sample data are supplied, in a variate, using the Y parameter. Similarly the base data are provided using the X parameter. The LABELS parameter can supply a variate, factor or text for labelling individual units in the output. If X is unset or missing, expansion raising is used (i.e. the usual stratified random sampling analysis) but within a stratum units must either all have base data or all lack it. (Note: *stratum* is used here in the survey sense, not as in the ANOVA directive: i.e. the units are assumed to be classified into groups, and each group is called a stratum.) If option XMISSING is set to fault, any missing base data will cause a fault.

The vectors Y, X and LABELS should usually have one row for each unit in the survey population, with unsampled or non-responding units having a missing value in the Y variate. However, if parameter NUNITS is set, the Y variate may contain only the response data; NUNITS then supplies the information about the number of units in each stratum in the full population. Similarly, if ratio estimation is required, XTOTALS should contain the population totals of X in each stratum.

The METHOD specifies which method of ratio estimation to use. The setting separate estimates a ratio for each stratum, whereas settings combined and classicalcombined assume a common ratio in all strata. The classicalcombined method follows the approach shown in most textbooks, where the estimate for a stratum is given by $\sum X \times ratio$ where the summation is over all units in the stratum. This approach can produce illogical estimates in some situations (e.g. the estimate may be less than the sum of the responses) and so the combined method estimates only for the unobserved units and adds this to the sum of the observed responses in the stratum, i.e. $\sum Y + \sum X \times ratio$ where the summation of Y is over sampled (or responding) units and the summation of X is over unsampled units. Option COMBINEDSTRATUM is used with the separate ratio method and allows the ratio in a particular stratum to be reset to the combined ratio value; this can be a useful technique for dealing with the extreme ratios sometimes produced when the sampling fraction in a stratum is very low.

Printing is controlled via the PRINT option. The default settings are summary, totals and influence; these print a summary of the data, estimated totals and influence statistics, respectively. The setting means produces a table showing the estimated means, whilst ratio produces a low-resolution plot of the confidence limits for the ratio estimates; this can be useful when deciding whether a combined ratio estimate is to be used. The setting extra displays extra information relating to the analysis, including sums and means of the response data and raising factors (weights).

The CIPROBABILITY option sets the probability level used in calculation of confidence limits for means and totals. The CIMETHOD option controls how confidence limits are formed after bootstrapping: percentile uses simple percentiles of the bootstrapped distribution, whilst tdistribution calculates a standard error from the bootstrapped estimates and then uses the t-distribution to form intervals; the default of automatic uses the percentile method unless less than 400 bootstrap samples have been made.

The NINFLUENCE option controls the number of points of high influence printed. The COMPACT option can be used to switch to a compact, plain-text style for the output, designed for printing concise summaries of an analysis. When COMPACT=yes, the information printed depends on the width of the first output channel, with more information being displayed when

this can be done without splitting tables.

By default all standard errors and confidence limits are calculated using the conventional approximations. Alternatively, bootstrap methods may be used by setting the NBOOT option to the required number of bootstrap samples. In the case of ratio estimation, the samples are used to form bootstrap estimates of the ratio, which are then applied to the known population totals for X. Bootstrapping is carried out independently in each stratum, using the method described by Sarndal *et al.* (1992, page 442); this involves creating a "pseudopopulation" containing *n* replicates of each observation, where *n* is nearest integer to the expansion raising factor (inverse of inclusion probability) for the stratum. Bootstrap samples of the same size as the original sample are then taken from the pseudopopulation and used to compute the estimates. The SEED option specifies the seed to use in the random number generator used to construct the bootstrap samples. The default value of zero continues an existing sequence of random numbers or, if the generator has not yet been used in this run of Genstat, it initializes the generator automatically.

Graphical output is available by setting the PLOT option. The setting single produces a single plot of the response data against X or against the stratum number if X is unset. A fitted line is shown if one of the combined ratio methods is used. The separate setting produces one graph for each stratum, with up to six graphs on each screen. All graphs are plotted on the log scale.

Output can be saved using the parameters TOTALS, SETOTALS, MEANS, SEMEANS, LTOTALS, UTOTALS, LMEANS and UMEANS. These are generally set to a table classified by the stratification factor but, if option SAVESUMMARY=yes, then they save scalars containing only the grand total summed over all strata. Ratios can be saved in a table using the RATIOS parameter, whilst the residual variances in each stratum can be saved using VARIANCES; the latter are useful for working out optimal allocation strategies for future surveys. Fitted values and influence statistics may be saved using parameters FITTEDVALUES and INFLUENCE. The fitted values are the X value multiplied by the appropriate ratio for each unit or, where expansion raising is used, the mean Y value for the stratum.

Options: PRINT, PLOT, XMISSING, RESTRICTED, STRATUMFACTOR, NINFLUENCE, METHOD, SAVESUMMARY, COMBINEDSTRATUM, ROWS, COLUMNS, NBOOT, SEED, CIPROBABILITY, CIMETHOD, COMPACT.

Parameters: Y, X, LABELS, NUNITS, XTOTALS, TOTALS, SETOTALS, MEANS, SEMEANS, RATIOS, FITTEDVALUES, INFLUENCE, LTOTALS, UTOTALS, LMEANS, UMEANS, VARIANCES.

Method

The methods used are described in most survey analysis textbooks; see for example, Sampford (1962) or Lehtonen & Pahkinen (1994). Most calculations are carried out using Genstat table structures.

Action with RESTRICT

The action with RESTRICT depends of the setting of the RESTRICTED option. By default restricted units are totally excluded from the analysis. If RESTRICTED is set to add, restricted observations are excluded from the ratio calculations but then added back into the total estimates; this is a technique for dealing with nonrepresentative outliers (see e.g. Lee, 1995), which are believed to be genuine observations but are not representative of the wider population.

References

Lee, H. (1995). Outliers in Business Surveys. Chapter 26 of *Business Survey Methods* (ed. Cox, Binder, Hinnappa, Christianson, Colledge & Kott). Wiley, New York.

Lehtonen, R. & Pahkinen, E.J. (1994). *Practical Methods for Design and Analysis of Complex Surveys*. Wiley, New York.

Sampford, M.R. (1962). *An introduction to Sampling Theory*. Oliver & Boyd, London.

See also

Procedures: SVBOOT, SVCALIBRATE, SVGLM, SVHOTDECK, SVREWEIGHT, SVSAMPLE, SVTABULATE, SVWEIGHT.

Genstat Reference Manual 1 Summary section on: Survey analysis.

SVTABULATE

Tabulates data from random surveys, including multistage surveys and surveys with unequal probabilities of selection (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (summary, stratumsummary, psusummary, totals, means, ratios, influence, wald, quantiles, monitor); default summ, tota, infl
PLOT = *string token*	Controls which high-resolution graphs are plotted (single, separate, weights, influence, diagnostic); default * i.e. none
STRATUMFACTOR = *factor*	Stratification factor; default *, i.e. unstratified
NUNITS = *table*, *scalar* or *variate*	Numbers of units in each STRATUMFACTOR level (for a multistage design these will be the number of primary sampling units)
SAMPLINGUNITS = *factor*	Factor indicating the primary sampling units; default *, i.e. single stage design
NSECONDARYUNITS = *table*, *scalar* or *variate*	Numbers of secondary sampling units for the levels of the SAMPLINGUNITS factor
CLASSIFICATION = *factors*	Domains for which separate estimates are required
NINFLUENCE = *scalar*	Number of influential points to print; default 10
MRFACTOR = *identifiers*	Identifier of factors to index the sets of multiple responses in the tables
WEIGHTS = *variate*	Survey weights
FPCOMIT = *string token*	Whether to omit the finite population correction from calculation of variances (yes, no); default no
METHOD = *string token*	Method of bootstrapping (simple, sarndal); default simp
NBOOT = *scalar*	Number of bootstrap samples to use; default 0 uses a Taylor series approximation
SEED = *scalar*	Seed for random number generator for bootstrap; default 0
CIPROBABILITY = *scalar*	The probability level for the confidence intervals; default 0.95
CIMETHOD = *string token*	Method for forming confidence intervals (automatic, tdistribution, percentile, logit); default auto
PERCENTQUANTILES = *scalar* or *variate*	Percentage points for which quantiles are required; default 50 (i.e. median)

Parameters

Y = *variates*	Response data
X = *variates*	Base data for ratio estimation
LABELS = *variates* or *texts*	Labels for influential points
OUTWEIGHTS = *tables*	Saves weights
TOTALS = *tables*	Saves total estimates
SETOTALS = *tables*	Saves standard errors of estimates
VCTOTALS = *symmetric matrices*	Saves variance-covariance matrix of total estimates
MEANS = *tables*	or scalars Saves mean estimates

SEMEANS = *table*	Saves standard errors of mean estimates
VCMEANS = *symmetric matrices*	Saves variance-covariance matrix of mean estimates
RATIOS = *tables*	Saves estimates of ratios
SERATIOS = *tables*	Saves standard errors of ratios
VCRATIOS = *symmetric matrices*	Saves variance-covariance matrix of ratio estimates
NOBSERVATIONS = *tables*	Saves numbers of (non-missing) observations
SUMWEIGHTS = *tables*	Saves sums of weights
FITTEDVALUES = *variates*	Supplies fitted values for each observation
INFLUENCE = *variates*	Saves influence statistics
WALD = *variates*	Saves Wald statistics
QUANTILES = *tables* or *pointers*	Table to contain quantiles at a single PERCENTQUANTILE or pointer of tables for several PERCENTQUANTILEs
SEQUANTILES = *tables* or *pointers*	Saves standard errors of quantiles
VCQUANTILES = *tables* or *pointers*	Saves variance-covariance matrix of quantiles
LQUANTILES = *tables* or *pointers*	Saves lower confidence limits of quantiles
UQUANTILES = *tables* or *pointers*	Saves upper confidence limits of quantiles
LTOTALS = *tables*	Saves lower confidence limits of totals
UTOTALS = *tables*	Saves upper confidence limits of totals
LMEANS = *tables*	Saves lower confidence limits of means
UMEANS = *tables*	Saves upper confidence limits of means
LRATIOS = *tables*	Saves lower confidence limits of ratios
URATIOS = *tables*	Saves upper confidence limits of ratios
CELLINFLUENCE = *variates*	Saves influence statistics for individual cells

Description

SVTABULATE procedure calculates estimates from surveys, together with the correct asymptotic standard errors, allowing for the design of the survey. In particular, information about the numbers of sampling units in the survey population is needed and this can be supplied in one of three ways.

1. The WEIGHTS option can be used to supply weights which will generally be the inverse of the probability of selection (pi expansion weights, Sarndal *et al*. 1992). This is simple, but cannot convey the full design information for multi-stage surveys.
2. The option NUNITS can be used to list the number of primary sampling units per stratum using a table or variate with one value for each stratum. Similarly, in a two-stage design, NSECONDARYUNITS indicates the number of secondary units in each primary sampling unit.
3. The dataset can contain the full survey population with unsampled (or non-responding) units indicated by missing values for the response variables. This allows Genstat to deduce the numbers of units without the need to supply any further information; it is thus simple to use, but is not feasible with large or complex surveys. The NUNITS (and NSECONDARYUNITS if appropriate) option should be set to a value of –1 to indicate that this is required.

Other information on the survey design is provided using the STRATUMFACTOR and SAMPLINGUNITS options.

The response variable is specified using the Y parameter. Estimated counts of the number of observations can be produced by leaving the parameter unset (this is equivalent to analysing a vector of 1's). The Y parameter can also be left unset if the procedure is used to calculate survey weights. The X parameter can be set in order to produce estimates of the ratio Y/X. By default

estimates of totals, means or ratios are for the whole population, but the CLASSIFICATION option can be set to one or more factors defining subsets of the data for which estimates are required. The list of CLASSIFICATION factors can also include pointers defined using the FMFACTORS procedure, representing a multiple response factor. SVTABULATE generates an ordinary factor to classify the dimension of the tables corresponding to each set of multiple responses. You can supply identifiers for these factors (thus allowing them to be accessed outside the procedure), using the MRFACTOR option.

The FITTEDVALUES parameter is used when estimating population totals via a model-assisted approach. Variance estimates are then calculated using the residual deviation about the fitted values. This can be used in conjunction with the SVCALIBRATE procedure to provide estimates following calibration weighting.

Output is controlled by the PRINT and PLOT options. The latter produces various plots that are useful in identifying outliers and influential points which may require further investigation. The setting single of the PLOT option produces a scatterplot of values of Y against X, whilst separate produces a separate graph for each combination of levels of the CLASSIFICATION factors. (excluding multiple response factors). The graphs are log-transformed, unless negative values are present. If the log-transformation is required and zeros are present a small constant is added first. When X is unset, both single and separate produce a scatterplot of Y against CLASSIFACTION. The weights and influence settings produce histograms of the weights and influence statistics respectively. The setting diagnostic produces a scatterplot of influence statistics against weights; this plot tends to be more informative than the histograms with large datasets. The influence statistic for an observation is defined as the absolute percentage change in the total estimate when the observation is replaced by a missing value and the associated weight redistributed to other units in the same stratum. When CLASSIFICATION is set, influence statistics are printed for individual cells in the table of results, as well as for the grand total. When PRINT is set to influence, details are printed of the observations with the highest influence; the number printed can be controlled by the NINFLUENCE option. By default this output is labelled by the row number of the observation, but the LABELS parameter can be used to specify more meaningful identifiers in the form of a variate, text or factor.

The FPCOMIT option is provided so that the finite population correction (see e.g. Sarndal *et al*. 1992) can be omitted. This is usually done when a simplified variance estimate is produced for multistage samples by ignoring the within-cluster component of variation (the *ultimate cluster* approach); since this is non-conservative, the omission of the FPC is sometimes advocated to counteract this and to ensure that standard errors are appropriate. Genstat will produce the ultimate cluster results if it is only provided with the survey weights (i.e. NUNITS and NSECONDARYUNITS left unset), but this approach is not recommended since the correct analysis can be produced with little extra effort.

Results of the analysis can be saved using the parameters TOTALS, MEANS, RATIOS and QUANTILES, with the corresponding standard errors using SETOTALS, SEMEANS, SERATIOS and SEQUANTILES. Confidence limits are saved using LTOTALS, LMEANS, LRATIOS and LQUANTILES for the lower limits, and UTOTALS, UMEANS, URATIOS and UQUANTILES for the upper limits. By default, 95% confidence limits are produced, but this may be changed using the CIPROBABILITY option. When the Y parameter is unset, TOTALS, SETOTALS, LTOTALS and UTOTALS contain estimated counts of observations. Numbers of (non-missing) observations and the sum of the weights can be saved using the NOBSERVATIONS and SUMWEIGHTS parameters. These are set to tables classified by the CLASSIFICATION factors; if CLASSIFICATION is unset, they are they are set to a table with a single cell labelled 'All data'. The OUTWEIGHTS and INFLUENCE parameters allow you to save variates containing the weights and influences, respectively. CELLINFLUENCE saves the influence statistics with respect to the individual cells in the table of results, as opposed to the influence statistics for the grand total, which is saved by the INFLUENCE parameter. The WALD parameter can be used to save Wald statistics

comparing means between the different levels of the CLASSIFICATION factors.

The simplest quantile, and the one produced by default, is the median (50% quantile), but the PERCENTQUANTILE option allows you to request any percentage point between 1 and 99. Moreover, by specifying a variate as the setting for PERCENTQUANTILE, you can obtain several quantiles at the same time. However, if you then want to save the results, the setting of the QUANTILES parameter must be a pointer with length equal to the required number of quantiles, instead of a single table.

By default, standard errors and confidence limits are based on Taylor-series approximations. However, bootstrap standard errors can be obtained by setting the NBOOT option to the desired number of bootstrap samples. For exploratory analyses a relatively low value (perhaps 20) may suffice, but where test statistics or confidence limits are required a value of at least 400 is recommended. The CIMETHOD option controls how the confidence limits are formed:

percentile	uses simple percentiles of the bootstrapped distribution;
tdistribution	calculates a standard error from the bootstrapped estimates and then uses the t-distribution to form intervals;
logit	is for proportions, and ensures that the calculated limits lie between 0 and 1 (see Heeringa *et al.* 2010);
automatic	uses the percentile method when at least 400 bootstrap samples have been used, otherwise it uses the t-distribution method when Y is set, and the logit method when Y is not set.

The default is CIMETHOD=automatic.

The bootstrapping method is selected using the METHOD option. In a one-stage design the default of simple forms each bootstrap sample by sampling with replacement from the original sample within each stratum. In a two-stage design (i.e. if SAMPLINGUNITS is set), primary sampling units are first sampled with replacement, and then secondary units are sampled with replacement within the selected primary units. Variance estimates from the boostrapping process will be biased where there are very few sampling units in each stratum and so the method is not recommended in this situation. The setting METHOD=sarndal constructs a "pseudo-population" by replicating each sampled unit by the rounded value of its weight, so that, for example, an observation with weight 16.1 is represented sixteen times in the pseudo-population (see Sarndal *et al*. 1992, page 442). The bootstrap sample is formed by sampling with replacement from this pseudo-population. Option SEED provides a seed for the random sampling.

Options: PRINT, PLOT, STRATUMFACTOR, NUNITS, SAMPLINGUNITS, NSECONDARYUNITS, CLASSIFICATION, NINFLUENCE, MRFACTOR, WEIGHTS, FPCOMIT, METHOD, NBOOT, SEED, CIPROBABILITY, CIMETHOD, PERCENTQUANTILES.

Parameters: Y, X, LABELS, OUTWEIGHTS, TOTALS, SETOTALS, VCTOTALS, MEANS, SEMEANS, VCMEANS, RATIOS, SERATIOS, VCRATIOS, NOBSERVATIONS, SUMWEIGHTS, FITTEDVALUES, INFLUENCE, WALD, QUANTILES, SEQUANTILES, VCQUANTILES, LQUANTILES, UQUANTILES, LTOTALS, UTOTALS, LMEANS, UMEANS, LRATIOS, URATIOS, CELLINFLUENCE.

Method

The procedure uses the methods for survey analysis described in most survey analysis textbooks; Sarndal *et al*. (1992) give the best account of these for the case where weights vary within a stratum or sampling unit. If the dataset contains the full population, as opposed to just sampled or responding units, the options NUNITS and/or NSECONDARYUNITS can be set to −1, in which case the procedure calculates the numbers using TABULATE.

When bootstrapping is used, bootstrap samples are formed using the SVBOOT procedure.

Action with RESTRICT

Restrictions of the Y variate or any of the CLASSIFICATION factors are used to define a subpopulation, and the estimates produced relate to that subpopulation. Any restrictions on SAMPLINGUNITS, STRATUMFACTOR or WEIGHTS are ignored.

References

Heeringa, S.G., West, B.T. & Berglund, P.A. (2010). *Applied Survey Data Analysis*. CRC Press, Boca Raton.

Lehtonen, R. & Pahkinen, E.J. (1994). *Practical Methods for Design and Analysis of Complex Surveys*. Wiley, Chichester.

Sarndal, C., Swenssion, B. & Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer-Verlag, New York.

See also

Procedures: SVBOOT, SVCALIBRATE, SVGLM, SVHOTDECK, SVREWEIGHT, SVSAMPLE, SVSTRATIFIED, SVWEIGHT.

Genstat Reference Manual 1 Summary section on: Survey analysis.

SVWEIGHT

Forms survey weights (S.D. Langton).

Options

PRINT = *string token*	Controls printed output (summary, stratumsummary, psusummary); default summ, stra, psus
PLOT = *string token*	Controls which high-resolution graphs are plotted (weights); default * i.e. none
STRATUMFACTOR = *factor*	Stratification factor; default *, i.e. unstratified
NUNITS = *tables*, *scalars* or *variates*	Numbers of units in each STRATUMFACTOR (for a multistage design these will be the number of primary sampling units)
SAMPLINGUNITS = *factor*	Factor indicating the primary sampling units; default *, i.e. single stage design.
NSECONDARYUNITS = *tables*, *scalars* or *variates*	Numbers of secondary sampling units for each level of the SAMPLINGUNITS factor

Parameters

Y = *variates* or *scalars*	Response data or a scalar indicating the number of sampled units
OUTWEIGHTS = *variates*	Saves weights

Description

SVWEIGHT creates weights for surveys. The information about the numbers of sampling units in the survey population can be supplied in one of two ways.

1. The option NUNITS can be used to list the number of primary sampling units per stratum using a table or variate with one value for each stratum. Similarly, in a two-stage design, NSECONDARYUNITS indicates the number of secondary units in each primary sampling unit.
2. The dataset can contain the full survey population with unsampled (or non-responding) units indicated by missing values for the response variables (Y parameter). This allows Genstat to deduce the numbers of units without the need to supply any further information; it is thus simple to use, but is not feasible with large or complex surveys. The NUNITS (and NSECONDARYUNITS if appropriate) option should be set to a value of –1 to indicate that this is required.

With the first method the Y parameter can be left unset, except in the case of a simple random sample, where it must be set in order for the procedure to know the number of sampled units; in this case Y can either be set to a variate containing the responses or to a scalar containing the number of sampled units. Other information on the survey design is provided using the STRATUMFACTOR and SAMPLINGUNITS options. The OUTWEIGHTS parameter saves the variate of weights (corresponding to each response variable in the case where more than one Y variable is set).

The PRINT option allows you to print various summaries, and you can set PLOT=weights to plot a histograms of the weights.

Options: PRINT, PLOT, STRATUMFACTOR, NUNITS, SAMPLINGUNITS, NSECONDARYUNITS.
Parameters: Y, OUTWEIGHTS.

Method
The procedure uses the methods for survey analysis described in most survey analysis textbooks, calculating weights as the inverse of the probabilities of inclusion (see for example, Sarndal *et al*. 1992).

Action with `RESTRICT`
Any restrictions on `Y`, `SAMPLINGUNITS`, `STRATUMFACTOR` or `WEIGHTS` are ignored.

Reference
Sarndal, C., Swenssion, B. & Wretman, J. (1992). *Model Assisted Survey Sampling*. Springer-Verlag, New York.

See also
Procedures: `SVBOOT`, `SVCALIBRATE`, `SVGLM`, `SVHOTDECK`, `SVREWEIGHT`, `SVSAMPLE`, `SVSTRATIFIED`, `SVTABULATE`.
Genstat Reference Manual 1 Summary section on: Survey analysis.

TABINSERT

Inserts the contents of a sub-table into a table (R.W. Payne).

Options

OLDTABLE = *tables*	Table containing the original values
SUBTABLE = *tables*	Sub-table to insert into the original table
NEWTABLE = *tables*	Tables to store the new values; if this is not set, these replace those in the original table

Parameters

OLDFACTOR = *factors*	Factors classifying the dimensions of the old table that are smaller in the sub-table
SUBFACTOR = *factors*	Specifies the factors classifying the corresponding dimensions of the sub-table
FREPRESENTATION = *string token*	How to match the values of each OLDFACTOR and SUBFACTOR (levels, labels); default leve

Description

TABINSERT allows you to replace values in a table by those in a sub-table. It can also be used to insert values into the margins of a table. The original table and the sub-table are specified by the OLDTABLE and SUBTABLE options, respectively. You can use the NEWTABLE option to a specify a table to store the modified table values. If this is not set, they replace those in the original table.

The sub-table will usually have the same number of classifying factors as the original table. Some may be in common (and these can be ignored). Pairs of factors that differ are specified by the OLDFACTOR and SUBFACTOR parameters. The FREPRESENTATION indicates whether the factors are to be matched by their levels (default) or their labels. The idea is that the levels (or labels) of the SUBFACTOR are a subset of those of the OLDFACTOR, indicating where the values of the sub-table are to be inserted. If you omit some factors of the original table from both the sub-table and the OLDFACTOR list, the values of the sub-table are inserted into their margins in the modified table.

If both tables have margins, those in the sub-table will be transferred as well as those in the body of the table. If you want to omit the marginal values, you should remove the margins from the sub-table, using the MARGIN directive with parameter METHOD=deletion. You can also use MARGIN to recalculate the margins in the new table, if they are no longer valid after the values in the sub-table have been inserted.

Options: OLDTABLE, SUBTABLE, NEWTABLE.
Parameters: OLDFACTOR, SUBFACTOR, FREPRESENTATION.

Method

TABINSERT uses COMBINE to determine where the cells of the sub-table occur in the original table.

See also

Directives: COMBINE, TABLE, TABULATE.
Procedures: TABMODE, TABSORT, T%CONTROL.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

TABMODE

Forms summary tables of modes of values (R.W. Payne).

Options

PRINT = *string token*	Controls whether or not the modes are printed (mode); default * i.e. no printing
CLASSIFICATION = *factors*	Factors classifying the tables; if unset, the overall mode is formed for all the values in each DATA vector

Parameters

DATA = *variates* or *factors*	Data values whose modes are to be formed
MODES = *tables* or *scalars*	Save the modes for each DATA vector

Description

TABMODE forms tables summarizing the values in a variate or factor by their modes i.e. by the non-missing values that occur most often. The variates or factors are specified by the DATA parameter, and the modes can be saved using the MODES parameter. The CLASSIFICATION option can supply a list of factors, so that MODES will then be a table giving the mode for the DATA values with each combinations of the factor levels. Alternatively, if CLASSIFICATION is not set, MODES is a scalar containing the mode of all the values in DATA. You can request for the modes to be printed by setting option PRINT=modes.

Options: PRINT, CLASSIFICATION.
Parameters: DATA, MODES.

Method

TABMODE converts any DATA variates into factors (with a level for each distinct value) using GROUPS. It then uses TABULATE to form tables of counts, classified by the CLASSIFICATION factors (if any) and by the DATA factor. CALCULATE is then used to identify the mode. If there are several values occurring the maximum number of times, TABMODE will take the middle one.

Action with RESTRICT

TABMODE takes account of any restrictions defined on the DATA variates or factors.

See also

Directive: TABULATE.
Procedure: DESCRIBE.
Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Calculations and manipulation.

TABSORT

Sorts tables so their margins are in ascending or descending order (R.W. Payne).

Options

PRINT = *string tokens*	Controls output (tables, histograms); default * i.e. none
DIRECTION = *string token*	Direction of sorting (ascending, descending); default asce
METHOD = *string token*	Method to use to construct a marginal table for the sorting of a factor when there is no one-way table classified by the factor in the TABLE list, and the first table in the TABLE list classified by the factor has no margins (totals, means, minima, maxima, variances, medians); default tota
FACTORS = *pointer*	Specifies or saves a list of classifying factors of the tables in the TABLE list
NEWFACTORS = *pointer*	Specifies or saves a list of classifying factors of the new tables, corresponding to those in the FACTORS pointer
EXCLUDE = *pointer*	Factors to exclude from sorting
NBEST = *string tokens*	Number of (best) levels to include from each sorted factor; default * i.e. all of them

Parameters

TABLE = *tables*	Tables to be sorted
NEWTABLE = *tables*	Allows the new sorted tables to be saved
TITLE = *texts*	Title to be used when displaying each table
FIELDWIDTH = *scalars*	Field width for printing each table
DECIMALS = *scalars*	Decimal places for each table

Description

This procedure sorts tables so that their margins are in a specified order. With a multi-way table, for example, this may help in interpreting an interaction from an analysis of variance. With a one-way table, it allows the cells to be displayed in ascending order, as in a Pareto chart.

The original tables are supplied by the TABLE parameter. The NEWTABLE parameter can be used to save the sorted tables.

If you want to specify your own ordering, the FACTORS and NEWFACTORS options can be set to pointers of pre-defined factors indicating the ways in which each dimension of the tables is to be sorted: FACTORS contains factors from the classifying sets of the original tables, and NEWFACTORS contains the corresponding factors for the new tables (with the levels in the new order).

Alternatively, you can let TABSORT define the ordering. For each factor classifying the original tables, the ordering is obtained using a one-way table for that factor. This may be available amongst the list of original tables (specified by the TABLE parameter). If not, TABSORT finds the first table in the list with the factor in its classifying set. If the table has margins, then TABSORT will extract the appropriate one-way margin. Otherwise, it first constructs the margins using the MARGIN directive; the METHOD option then defines how the margin is formed (using means, medians and so on). Having obtained a suitable one-way table, TABSORT forms a new factor whose levels are in the order that will arrange the entries of the table in either ascending or descending order according to the setting of the DIRECTION option (default ascending). The FACTORS and NEWFACTORS options can then be used to save pointers containing the factors and reordered factors for future use. Note also, that even if you do not want to use the factors in

future, you can use the pointers to specify identifiers for the new factors to be used when the tables are printed. (You must specify both of them, so that TABSORT can tell how the new identifiers correspond to the original factors.) The EXCLUDE option can be set to a pointer containing factors that are not to be re-ordered automatically, but should be left unchanged.

The NBEST option specifies the number of levels to include from each sorted factor. So, setting NBEST=5 would take only the first five levels in the sorted order. This may be useful if you have a large table, and want to show only the best part of the table (as defined by the sorting of the margins). This default is to include all of the levels.

The PRINT option controls the output produced by TABSORT. The setting tables prints the tables. The setting histograms, causes any one-way tables to be plotted by the DHISTOGRAM directive, and any two-way tables to be plotted by D3HISTOGRAM. The TITLE parameter allows you to supply a title to be used in the display of each table. The FIELDWIDTH parameter specifies field widths, and the DECIMALS parameter specified numbers of decimal places.

Options: PRINT, DIRECTION, METHOD, FACTORS, NEWFACTORS, EXCLUDE, NBEST.
Parameters: TABLE, NEWTABLE, TITLE, FIELDWIDTH, DECIMALS.

Method

TABSORT uses FACSORT to sort the factors and COMBINE to reorder the table.

See also

Directives: COMBINE, TABLE, TABULATE, MARGIN.
Procedures: MTABULATE, SVSTRATIFIED, SVTABULATE, TABINSERT, TABMODE, T%CONTROL.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Six sigma, Survey analysis.

TABTABLE

Opens a tabbed-table spreadsheet in the Genstat client, PC Windows only (D.B. Baird).

Options

IDENTIFIER = *identifier*	Identifier for the combined table when several tables are specified by TABLE
PAGEFACTOR = *factor*	Specifies the the classifying factor to go across the tabs in the spreadsheet when TABLE is set to a single table, or gives the identifier of the factor to be created to index the tables when TABLE supplies several tables

Parameter

TABLE = *tables*	Tables to be placed into a tabbed-table spreadsheet

Description

TABTABLE forms a multiple spreadsheet in the Genstat client from either a single table with three or more dimensions, or several tables with two or more dimensions.

The table or tables are specified using the TABLE parameter. If TABLE is set to a single table, the PAGEFACTOR option can be used to specify which factor is distributed across the tabs; if this is not specified, the first classifying factor is used.

Alternatively, if TABLE is set to a list of several tables, the PAGEFACTOR option specifies the identifier of the new factor that is set up to index the individual tables (default Tabs). The IDENTIFIER option then supplies an identifier for the new combined table (default Table).

Options: IDENTIFIER, PAGEFACTOR.
Parameter: TABLE.

Method

The procedure creates slices of the table and displays these as pages in a Genstat spreadsheet book by using procedure FSPREADSHEET with the internal SLICE option set.

See also

Directives: SPLOAD, TABLE.
Procedure: FSPREADSHEET.

TALLY

Forms a simple tally table of the distinct values in a vector (D.B. Baird & R.D. Stern).

Options

PRINT = *string tokens*	What to print out for each vector (frequencies, percentages, cumfrequencies, cumpercentages, cumgraph, all); default freq, perc
GRAPH = *string tokens*	What to display as graphs (cumulative, %cumulative); default * i.e. no graphs
NGROUPS = *scalar*	Number of groups to form from a DATA variate or factor (ignored for texts); default * forms a group for each distinct value allowing for rounding (see DECIMALS)
DECIMALS = *scalar*	Number of decimal places to which to round the DATA before forming the groups; default * i.e. no rounding
BOUNDARIES = *string token*	Whether to interpret the LIMITS as upper or lower boundaries (upper, lower); default lowe
DIRECTION = *string token*	Order in which to sort (ascending, descending); default asce
OMITEMPTY = *string token*	Whether empty groups are omitted (yes, no); default no
WEIGHTS = *variate*	Weights to be used in the tabulations; default * indicates that all units have weight 1
PQUANTILES = *string token*	Whether to include quantiles on the plot (yes, no); default no
WINDOW = *scalar*	Window in which to plot the graphs; default 1 if GROUPS is set, or 3 otherwise
KEYWINDOW = *scalar*	Window in which to display the key when GROUPS is set; default 2
SCREEN = *string token*	Whether to clear screen before the plot (clear, keep); default clea

Parameters

DATA = *variates*, *factors* or *texts*	Data to be tallied
GROUPS = *factors*	Defines groupings of the data, to be tallied into separate tables; default * i.e. none
LIMITS = *variates* or *texts*	Limits to define the groups within the tally tables
FREPRESENTATION = *string tokens*	Specifies the representation used to define the sort order of a DATA factor (ordinals, levels, labels); default leve
VALUES = *variates*, *texts* or *pointers*	Saves the distinct groups formed for the tally tables
FREQUENCIES = *variates* or *pointers*	Saves the frequencies of the groups in the tally tables
PERCENTAGES = *variates* or *pointers*	Saves the percentage occurrences of the groups
CUMFREQUENCIES = *variates* or *pointers*	Saves the cumulative frequencies of the groups
CUMPERCENTAGES = *variates* or *pointers*	Saves the cumulative percentages of the groups
TITLE = *texts*	Title for plot; default automatically forms a title containing the identifiers of the DATA vector and any

	GROUPS factor
XTITLE = *texts*	Title for the axis representing data values; default uses the identifier of the DATA vector

Description

TALLY forms and displays simple tally tables of a vector, giving the counts, percentages, and cumulative counts and percentages of each distinct value. The data values are specified by the DATA parameter, in either a variate, a factor or a text. You can also define groups, by specifying a factor using the GROUPS parameter. Separate tables are then formed for each group.

By default, the factor classifying the groups within the tally tables contains a level for each distinct data value. You can decrease the number of groups formed from a DATA variate or text by specifying the NGROUPS and DECIMALS options, or the LIMITS parameter. These work exactly as in the GROUPS directive. If limits are specified, the BOUNDARIES option controls whether these are interpreted as upper or lower boundaries of the groups; by default they are lower limits. The value that is used to represent each group is the median of the units in the group.

The WEIGHTS option can supply a variate of weights for the units of the vector, to be used when calculating the table. If this is not set, the units are all assumed to have weights equal to one.

The PRINT option controls which summaries are printed. The DIRECTION option controls the order of the tally table (ascending or descending). For a DATA factor, the FREPRESENTATION parameter controls which attribute is used to sort the groups (ordinals, levels or labels); by default the levels are used. The OMITEMPTY option can be used to omit empty groups.

The GRAPH option may be set to cumulative to produce a cumulative frequency graph, or %cumulative to produce a percentage graph. The PQUANTILES option controls whether or not the graphs include quantiles. The WINDOW and KEYWINDOW options specify the numbers of the windows to use for the plot and key respectively, and the SCREEN option controls whether the screen is cleared first, in the usual way. The TITLE parameter allows you to define an overall title for the graphs, and the XTITLE parameter allows you to define a title for their x-axes. If these are not set, suitable titles are defined automatically.

The VALUES, FREQUENCIES, PERCENTAGES, CUMFREQUENCIES, CUMPERCENTAGES parameters can be used to save the information. This is in variates or texts, if there are no GROUPS; otherwise it is in pointers, containing a variate or text for each group.

Options: PRINT, GRAPH, NGROUPS, DECIMALS, BOUNDARIES, DIRECTION, OMITEMPTY, WEIGHTS, PQUANTILES, WINDOW, KEYWINDOW, SCREEN.

Parameters: DATA, GROUPS, LIMITS, FREPRESENTATION, VALUES, FREQUENCIES, PERCENTAGES, CUMFREQUENCIES, CUMPERCENTAGES, TITLE, XTITLE.

Method

The GROUPS directive is called for a DATA variate or text, to form the grouping factor within the tally tables. TABULATE then forms the counts for each group.

Action with RESTRICT

Restricted units are left out of the tally results.

See also
Directive: TABULATE.
Procedures: DESCRIBE, MTABULATE, SVSTRATIFIED, SVTABULATE, TABMODE.
Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Calculations and manipulation, Survey analysis.

TCOMBINE

Combines several tables into a single table (D.B. Baird).

Options

FACTOR = *factor*	Supplies a factor to index the old tables in the new tables; if unset, an unnamed factor is used
LEVELS = *variate*	Allows levels to be supplied for the new factor
LABELS = *text*	Allows labels to be supplied for the new factor

Parameters

OLDTABLES = *pointers*	Each pointer supplies a set of tables to be combined
NEWTABLE = *tables*	Table to save each combined set of tables
FACVALUES = *variates* or *texts*	Values for the new factor, to indicate how the old tables should be ordered in the new table, or to allow some old tables to be omitted (available only when either LEVELS or LABELS are specified); default assumes that old tables are supplied for all the levels, in ascending level order
OLDDECIMALS = *scalars*, *tables* or *pointers*	Defines numbers of decimal places for the cells in the combined table contributed by each old table
NEWDECIMALS = *tables*	Saves tables to define the number of decimal places to use when printing each new table

Description

This procedure can be used to produce a new table by combining several tables together. The tables to be combined must be specified, in a pointer, by the OLDTABLES parameter. These must all have the same set of classifying factors, and must either all have margins, or all be without margins.

The new, combined table is saved by the NEWTABLE parameter. This will be classified by the same factors as the original tables, together with an additional factor to index the OLDTABLES. You can use the FACTOR option to specify an identifier for the new factor. If FACTOR is not set, an unnamed factor is used. (So, this will not have an identifier for you refer to.) However, its "extra" attribute is set to 'Tables', so that this label will appear as the title of the new dimension when the table is printed. (For more details, see the description of the EXTRA parameter of the FACTOR directive.)

You can use the LEVELS option to supply a variate of levels for the new factor, and the LABELS option to provide a text of labels. If neither LEVELS or LABELS is set, TCOMBINE uses the identifiers of the tables in the first OLDTABLES pointer as labels, and uses the standard default for the levels (i.e. integers 1, 2 etc). The OLDTABLES pointers must then contain the same number of tables, all in the same order.

Alternatively, provided one of LEVELS or LABELS has been set, you can use the FACVALUES parameter to indicate that an OLDTABLES pointer is either incomplete or in a different order. To use the levels, you set FACVALUES to a variate with a value for each table in the corresponding OLDTABLES pointer, defining the level of the new factor to which it should be allocated. Similarly, you can set FACVALUES to a text use the factor labels. Any level (or label) that is not included in a set of FACVALUES will generate a "slice" of missing values in the new table.

To print the new table effectively, you may find that different number of decimal places are needed for the cells arising from each of the old tables. For example, one of the old tables may have contained means (requiring several decimal places) while another may have contained the numbers observations uised to calculate each of the means (requiring no decimal places). When you print a table using the PRINT directive, you can set the DECIMALS parameter to a table (with

identical classifying factors) to define a different number of decimals for every table cell. You can save a suitable table of decimals using the NEWDECIMALS parameter of TCOMBINE. By default TCOMBINE decides on the number of decimals to use for each old table by looking at its decimals attribute. (This can be set by the DECIMALS parameter of the TABLE directive.) Alternatively, you can define your own numbers of decimals using the OLDDECIMALS parameter. Usually you will want to set this to a pointer containing, for each old table, either a scalar (if you want to use the same number of decimals for all its cells) or a table (if you want to specify different ones) However, if you want to use the same decimals for every old table, you can specify a single scalar or a single table instead.

Options: FACTOR, LEVELS, LABELS.

.

Parameters: OLDTABLES, NEWTABLE, FACVALUES, OLDDECIMALS, NEWDECIMALS.

See also

Directives: COMBINE, TABLE.

Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

TENSORSPLINE

Calculates design matrices to fit a tensor-spline surface as a linear mixed model (S.J. Welham & P.H.C. Eilers).

Options

`METHOD` = *string token*	Type of spline to use to construct the basis (`pspline`, `penalizedspline`); default `pspl`
`PENALTYMETHOD` = *string token*	Which tensor-spline penalty to use (`isotropic`, `semiconstrained`, `unconstrained`); default `unco`
`NX1SEGMENTS` = *scalar*	Specifies the number of segments between boundaries in the `X1` dimension; default `*` obtains a value automatically
`NX2SEGMENTS` = *scalar*	Specifies the number of segments between boundaries in the `X2` dimension; default `*` obtains a value automatically
`DEGREE` = *scalar*	Degree of polynomial used to form the underlying spline basis functions; default 1 for `METHOD=pena` and 3 for `METHOD=pspl`
`DIFFORDER` = *scalar*	Differencing order for P-spline penalty; default 2
`X1LOWER` = *scalar*	Specifies the lower boundary in the `X1` dimension; default takes the minimum value of `X1`
`X1UPPER` = *scalar*	Specifies the upper boundary in the `X1` dimension; default takes the maximum value of `X1`
`X2LOWER` = *scalar*	Specifies the lower boundary in the `X2` dimension; default takes the minimum value of `X2`
`X2UPPER` = *scalar*	Specifies the upper boundary in the `X2` dimension; default takes the maximum value of `X2`
`ORTHOGONALIZATION` = *string token*	How to orthogonalize the random basis (`fixed`, `none`); default `fixe`
`SCALING` = *scalar*	Scaling of the `XRANDOM` terms (`automatic`, `none`); default `auto`

Parameters

`X1` = *variates* or *factors*	Coordinates in the first dimension for which spline values are required
`X2` = *variates* or *factors*	Coordinates in the second dimension for which spline values are required
`XFIXED` = *matrices*	Saves the design matrix to define the fixed terms (excluding the constant) for fitting the tensor spline
`XRANDOM` = *pointers*	Saves the design matrices to define the random terms for fitting the tensor spline
`X1KNOTS` = *variates*	Saves the coordinates in the first dimension of the internal knots used to form the basis for the spline
`X2KNOTS` = *variates*	Saves the coordinates in the second dimension of the internal knots used to form the basis for the spline
`PX1` = *variates*	Specifies the coordinates in the first dimension at which to predict
`PX2` = *variates*	Specifies the coordinates in the second dimension at which to predict
`PFIXED` = *matrices*	Saves the design matrix for the fixed terms (excluding

	the constant) for the tensor spline at the prediction points
PRANDOM = *pointers*	Saves the design matrices for the random terms for the tensor spline at the prediction points

Description

TENSORSPLINE generates the fixed and random terms required to fit a tensor-spline surface as a linear mixed model, using REML estimation of the smoothing parameter.

The coordinates at which the spline is to be calculated are specified in two variates using X1 and X2 parameters. The full range of the spline in the X1 dimension can be defined using the X1LOWER and X1UPPER options; by default the lower limit is equal to the minimum values of X1 and the upper limit is equal to the maximum value. The range in the X2 dimension is defined similarly by the X2LOWER and X2UPPER options. In each dimension, the region between these bounds is divided into a number of equal segments, specified by the NX1SEGMENTS and NX2SEGMENTS options. The boundaries of these segments provide the set of knots used to form the spline terms in each dimension, and can be saved in variates by the X1KNOTS and X2KNOTS parameters. If the number of segments is unset in either dimension, the number is determined automatically as

$$\min([p/4], 35) + 1$$

(Ruppert 2002) where p is the number of unique values of the variate (X1 or X2), and $[r]$ denotes the integer part of the number r.

The METHOD option specifies whether to use P-splines (the default) or penalized splines to construct the basis. The degree of polynomial used to form the underlying spline basis functions is specified by the DEGREE option. This has a default of 3 for P-spline models ,and 1 for penalized spline models.

The DIFFORDER option specifies the differencing order to be used with P-spline models. This determines the strength of the penalty (for a given smoothness parameter). The default is to use second-order differencing. For a P-spline model, the underlying fixed polynomial in each dimension has degree d equal to DIFFORDER−1. For a penalized spline model, the underlying fixed polynomial in each dimension has degree d equal to the value specified by the DEGREE option.

The tensor-spline basis is constructed via interactions of the one-dimensional spline bases, as detailed in the methods section.

The ORTHOGONALIZATION option specifies whether the components of the spline to be fitted as random terms should be made orthogonal to the components to be fitted as fixed. The default action (ORTHOGONALIZATION=fixed) is to perform the orthogonalization, and this means that all of the polynomial trend associated with the fixed terms will be captured in the fixed part of the model. When ORTHOGONALIZATION=none, some of this trend may be contained within the random terms. Experience suggests that ORTHOGONALIZATION=none can lead to computational instability when the model is fitted using REML, especially for more complex models (d>1).

The fixed and random components of the tensor-spline terms are saved separately. The terms required to be fitted as fixed terms can be saved (in a matrix) using XFIXED parameter. This matrix does not include the constant term as this is added by default as part of a mixed model. For P-spline models with DIFFORDER=1, this is a null term and no matrix is saved.

The terms to be fitted as random can be saved (in a matrix) using the XRANDOM parameter. The terms to be fitted as random can be saved using the XRANDOM parameter. This is a saves a pointer, containing a number of matrices that depends on the setting of the PENALTYMETHOD option. The components of the random terms consist of interactions between:

- the underlying fixed polynomials in the first dimension with the random basis functions in the second dimension (d+1 terms)
- the random basis functions in the first dimension with the underlying fixed polynomials in the second dimension (d+1 terms); and

- the interaction of the two sets of random basis functions from each dimension (1 term).

The default is an unconstrained penalty, where a separate smoothing parameter is allowed for each term. In this case, the XRANDOM pointer has 2*d*+3 matrices, one for each term. For a semi-constrained penalty (PENALTYMETHOD=semiconstrained), the same smoothing parameter is imposed across the interaction of polynomials in the first dimension with random terms in the second, and for the interaction of random terms in the first dimension with polynomials in the second dimension. This is implemented by combining terms, so the XRANDOM pointer then has 3 matrices. For an isotropic penalty, which uses a single common penalty (PENALTY=isotropic), the terms are combined into a single matrix, so the pointer XRANDOM has a single element.

The random terms can be scaled so that, for each component matrix Z in XRANDOM,

```
TRACE(Z *+ T(Z)) = NROWS(Z)
```

This ensures that the average contribution of each component to the variance of an observation is equal to one. This improves interpretability of the spline variance components.

For PENALTYMETHOD=unconstrained, the fitted model is invariant to the scale of X1 and X2. This is not the case for the semi-constrained and isotropic penalties. Full (automatic) scaling is imposed by default, but this can be avoided by setting option SCALING=none. An intermediate option is available (SCALING=standardize) where the polynomial components of the random terms are standardized before being added into the random terms.

The tensor-spline terms required for prediction can be saved using the PXFIXED (for P-spline models, provided *d* is greater than 1) and PXRANDOM parameters. The PX1 and PX2 parameters provide the coordinates at which predictions are to be made.

Options: METHOD, PENALTYMETHOD, NX1SEGMENTS, NX2SEGMENTS, DEGREE, DIFFORDER, X1LOWER, X1UPPER, X2LOWER, X2UPPER, ORTHOGONALIZATION, SCALING.
Parameters: X1, X2, XFIXED, XRANDOM, X1KNOTS, X2KNOTS, PX1, PX2, PFIXED, PRANDOM.

Method

For each dimension, the appropriate one-dimensional P-spline or penalized spline basis functions are calculated (see procedures PSPLINE and PENSPLINE for details). Where the degree of the underlying fixed polynomials is equal to d (as explained under Description), there are d+1 polynomial terms for each dimension. To explain the construction of the tensor-splines, we represent these polynomials as vectors

```
X1[0...d] = X1**(0...d)
```

and

```
X2[0...d] = X2**(0...d)
```

The fixed part of the tensor-spline comprises all $(d+1)_2$ products of the form

```
X1X2[i][j] = X1[i]*X2[j]
```

for i=0...d, j=0...d. These vectors are copied into the columns of matrix XFIXED, with the exception of X1X2[0][0] which is the constant term, and added into the model automatically by the VCOMPONENTS directive. We represent the random parts of the two one-dimensional spline bases as matrices Z1 and Z2, with n1 and n2 columns, respectively. The random parts of the tensorspline are then created as interactions between the polynomial and spline terms, calculated as follows (i,j=0...d):

Term	Matrix calculation
X1[i].Z2	KRONECKER(X1[i]; ROW1(n2)) * Z2
Z1.X2[j]	Z1 * KRONECKER(ROW1(n1); X2[j])
Z1.Z2	KRONECKER(Z1; ROW1(n2)) * KRONECKER(ROW1(n1); Z2)

When SCALING=automatic or SCALING=standardize, the terms X1[] and X2[] are standardized before calculation of the random components of the model. When SCALING=none, no transformation is used.

For PENALTYMETHOD=unconstrained, the 2d+3 terms generated are kept separate, each with a separate smoothing parameter, which is estimated via the variance component associated with each random term. Because each of the terms has a separate variance component, this model is invariant to rescaling of the X1 and X2 variates. For PENALTY=semiconstrained, the matrices X1[i].Z2 (i=0...d) are concatenated into a single matrix and fitted as a single random term, with a common smoothing parameter. Similarly the matrices Z1.X2[j] (j=0...d) form a single term, and Z1.Z2 forms the third random term in the model. When SCALING=automatic or SCALING=standardized, this model is invariant to rescaling of X1 and X2, as all of the polynomial terms are standardized before formation of the matrices – this corresponds to a particular (and arbitrary) choice of scaling. When SCALING=none, the fitted model depends on the scale of the input variates. For PENALTYMETHOD=isotropic, all the random terms are concatenated into a single matrix and fitted with a common smmothing parameter. Again, with SCALING=none, the fit will depend on the scale of the input variates.

The design matrices for use in prediction are calculated by evaluating the same set of basis functions at the predict points.

Action with RESTRICT

Input structures must not be restricted.

Reference

Ruppert, D. (2002). Selecting the number of knots for penalised splines. *Computational & Graphical Statistics*, **11**, 735-757.

See also

Directives: VCOMPONENTS, REML.
Procedures: NCSPLINE, PENSPLINE, PSPLINE, RADIALSPLINE, SPLINE.
Function: SSPLINE.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

†TEQUIVALENCE

Performs equivalence, non-inferiority and non-superiority tests (R.W. Payne).

Options

`PRINT` = *string tokens*	Controls printed output (`confidence`, `description`, `test`); default `desc`, `test`
`PLOT` = *string token*	Controls plotting of the confidence intervals (`confidence`); default `*`
`CLASSIFICATION` = *pointer*	Specifies the factors classifying the table of means; must be supplied for a multi-way table
`METHOD` = *string token*	Type of test required (`equivalence`, `noninferiority`, `nonsuperiority`); default `equi`
`CIPROBABILITY` = *scalar*	The probability level for the confidence interval; default 0.95
`EQLIMITS` = *scalar* or *variate*	Limits for equivalence, non-inferiority or non-superiority
`TITLE` = *text*	Title for the graph of confidence intervals; default `'Confidence plot'`
`WINDOW` = *scalar*	Window for the graph of confidence intervals; default uses a window defined to fill the screen
`SCREEN` = *string token*	Whether to clear the screen before plotting the confidence intervals (`clear`, `keep`); default `clea`

Parameters

`MEANS` = *tables* or *variates*	Means to be compared
`CONTROL` = *scalars*, *texts* or *pointers*	Specifies the control treatment
`SED` = *symmetric matrix* or *scalar*	Standard errors of differences of the means
`DF` = *symmetric matrix* or *scalar*	Degrees of freedom for the standard errors of differences
`TSTATISTICS` = *tables* or *variates*	Saves the t-statistics for the tests
`PROBABILITIES` = *tables* or *variates*	Saves the probabilities from the tests
`DIFFERENCES` = *tables* or *variates*	Saves the differences from the control
`SEDCONTROL` = *tables* or *variates*	Saves the standard errors for the differences from the control
`DFCONTROL` = *tables* or *variates*	Saves the degrees of freedom for the differences from the control
`LOWER` = *tables* or *variates*	Saves the lower limits of the confidence intervals
`UPPER` = *tables* or *variates*	Saves the upper limits of the confidence intervals

Description

`TEQUIVALENCE` performs tests that can be used to assess whether a treatment is acceptably similar to a control (or standard) treatment.

For an equivalence test, you specify a lower and an upper limit for the difference between the mean of the treatment and the mean of the control. These define the zone within which the treatment can be regarded as equivalent to the control. The null hypothesis is that the treatment is not equivalent to the control i.e. that the difference in means lies outside that zone. The test calculates t-statistics for the distance of the difference above the lower limit, and its distance below the upper limit. Their probabilities provide the evidence to assess whether the difference lies within the equivalence zone, at the lower and upper end respectively. The procedure reports the larger (i.e. the less significant) of the two probabilities together with its t-statistic. You can

also check the tests by printing or plotting the confidence limits. Both tests need to be significant, and thus both ends of the confidence interval be within the zone, to conclude that the treatments are equivalent.

For non-inferiority, the difference between the mean of the treatment and the mean of the control must not be less than a (negative) limit. Any positive difference is acceptable, and a negative difference must be greater than the limit. The null hypothesis is that the treatment is inferior to the control i.e. that the difference is less than the limit. There is just one t-statistic, assessing whether the difference is greater than the limit, and the confidence interval is unbounded at the positive end.

Similarly, for non-superiority, the difference between the mean of the treatment and the mean of the control must not be greater than a (positive) limit. Any negative difference is acceptable, and a positive difference must be less than the limit. The null hypothesis is that the treatment is superior to the control i.e. that the difference greater than the limit. There is just one t-statistic, assessing whether the difference is less than the limit, and the confidence interval is unbounded at the negative end.

The `MEANS` parameter specifies a table or a variate containing the means that are to be assessed. For a variate, the `CONTROL` parameter specifies a scalar containing the number of the unit containing the control mean. For a table classified by a single factor, it specifies a scalar or a single-valued text to indicate the level or label of the factor for the control treatment. For a multi-way table, the `CLASSIFICATION` option must specify a pointer containing its classifying factors. The `CONTROL` parameter then specifies a pointer, containing scalars or single-valued texts, indicating the levels or labels of the classifying factors for the control (specified in the same order as in the `CLASSIFICATION` pointer). All the other means are tested against the control.

The `SED` and `DF` parameters specify standard errors for differences for the means and their numbers of degrees of freedom, respectively. These can supply scalars if they are the same for every pair of means, or otherwise symmetric matrices. The order of the rows in a symmetric matrix must be compatible with the order of the means in the table. To ensure compatibility, you should save the standard errors of differences and degrees of freedom from the same `AKEEP`, `PREDICT` or `VKEEP` statement as the means.

The `METHOD` option specifies the type of test. It can be set to either `equivalence` (default), `noninferiority` or `nonsuperiority`. The `EQLIMITS` option supplies a variate with the two limits for an equivalence test. The first value must be negative and the second must be positive. For a non-inferiority test, it supplies a scalar containing the (negative) limit. For a non-superiority test it supplies a scalar containing the (positive) limit.

Printed output is controlled by the `PRINT` option with settings:

`description`	control mean and limit(s),
`test`	t-statistic and probability level, and
`confidence`	confidence interval for the difference between means of treatments and control.

The default is `PRINT=description,test`. Usually a 95% confidence interval is calculated, but this can be changed by setting the `CIPROBABILITY` option to the corresponding probability. For the equivalence tests, the confidence interval is an amalgamation of two one-sided intervals, as you are making two one-sided tests. Each limit is therefore calculated for twice the distance from 100% (e.g. 90% instead of 95%, corresponding to a significance level of 5% for the test of equivalence).

You can plot the confidence intervals by setting option `PLOT=confidence`. The `TITLE` option specifies the title for the plot; default `'Confidence plot'`. The `WINDOW` option specifies the window to use. If this is not set, `TEQUIVALENCE` uses a window defined to fill the whole (0,1) × (0,1) square. The `SCREEN` option allows you to add the plot to an existing graphics screen; by default the screen is cleared.

The TSTATISTICS parameter can save the t-statistics for the tests, in a variate or a table according to the setting of the MEANS parameter. The PROBABILITIES parameter can similarly save the probabilities of the tests. The DIFFERENCES parameter can save the differences of the means from the control mean, again in either a variate or a table. The SEDCONTROL and DFCONTROL parameters can similarly save the standard errors of their differences and their degrees of freedom, respectively. The LOWER and UPPER parameters can save the lower and upper confidence limits, again in either a variate or a table.

Options: PRINT, PLOT, CLASSIFICATION, METHOD, CIPROBABILITY, EQLIMITS, TITLE, WINDOW, SCREEN.
Parameters: MEANS, CONTROL, SED, DF, TSTATISTICS, PROBABILITIES, DIFFERENCES, SEDCONTROL, DFCONTROL, LOWER, UPPER.

See also
Procedure: TTEST.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics, Analysis of variance.

THINPLATE

Calculates the basis functions for thin-plate splines (S.J. Welham & D.B. Baird).

No options

Parameters

`Y` = *variates* or *factors*	Y-coordinates of the data points
`X` = *variates* or *factors*	X-coordinates of the data points
`YKNOTS` = *variates* or *factors*	Y-coordinates of the knots
`XKNOTS` = *variates* or *factors*	X-coordinates of the knots
`TPSPLINE` = *variates* or *matrices*	Thin-plate spline basis, as either a pointer of variates (default if not already declared) or a matrix

Description

`THINPLATE` calculates the basis functions for thin-plate splines. The `X` and `Y` parameters each specify a variate or a factor with the x- and y-coordinates of the data points, and the `XKNOTS` and `YKNOTS` parameters similarly specify the positions of the knots. The basis functions are saved by the `TPSPLINE` parameter. By default these are stored in a pointer of variates, but you can save a matrix instead by declaring `TPSPLINE` in advance to be a matrix.

Options: none.
Parameters: `Y`, `X`, `YKNOTS`, `XKNOTS`, `TPSPLINE`.

See also

Directive: `VCOMPONENTS`.
Procedures: `SPLINE`, `LSPLINE`, `NCSPLINE`, `PENSPLINE`, `PSPLINE`, `RADIALSPLINE`, `TENSORSPLINE`.
Function: `SSPLINE`.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis, REML analysis of linear mixed models.

†TOBIT

Performs a Tobit linear mixed model analysis on data with fixed-threshold censoring (M.C. Hannah & V.M. Cave).

Options

`PRINT` = *string token*	Controls printed output (`summary`); default `summ`
`VPRINT` = *string tokens*	Controls printed output from the `REML` analysis of the data with censored observations replaced by their estimates (`model`, `components`, `effects`, `means`, `stratumvariances`, `monitoring`, `vcovariance`, `deviance`, `Waldtests`, `missingvalues`); default `mode`, `comp`, `Wald`
`PSE` = *string token*	Standard errors to be printed with tables of effects and means from the `REML` analysis (`differences`, `estimates`, `alldifferences`, `allestimates`, `none`); default `diff`
`PLOT` = *string token*	To display a scatter plot of the data with censored observations replaced by their estimates against the observed data(`scatterplot`); default `*`
`MAXCYCLE` = *scalar*	Sets a limit on the number of iterations performed by the E-M algorithm; default `30`
`TOLERANCE` = *variate*	Sets tolerance limits for convergence of the E-M algorithm on the treatment means and the variance components; default 0.1 and 0.05 for the treatment means and variance components, respectively
`RMETHOD` = *string token*	Which random terms to use when calculating the residuals during the E-step of the E-M algorithm (`final`, `all`); default `final`
`DIRECTION` = *string token*	The direction of the censoring (`left`, `right`); default `left` (i.e., the true values for the censored observations are less than or equal to the `BOUND`)

Parameters

`Y` = *variate*	Response variate to be analysed; no default, must be set
`BOUND` = *scalar*	Censoring threshold; no default, must be set
`CENSORED` = *variate*	Indicator variable for censored observations, with values of one where the response values are censored and zero otherwise
`INITIAL` = *scalar* or *variate*	Scalar or a variate providing starting values for the censored observations in the E-M algorithm
`NEWY` = *variate*	Saves a copy of the response variate with the censored observations replaced by their estimates
`YCENSORED` = *variate*	Saves a logical variate indicating which `Y` values are censored
`SAVE` = *REML save structure*	`REML` save structure from the analysis of the data with censored observations replaced by their estimates

Description

The `TOBIT` procedure performs a linear mixed model analysis on data values that are subject to fixed threshold censoring. Such censoring occurs when a measurement cannot be taken above or below a bound. For example, chemical concentrations may be censored when they fall below

a minimum level of quantification. The procedure uses an E-M algorithm to estimate values for the censored observations, and once converged, uses REML to analyse the response variate with the censored observations replaced by their estimates.

TOBIT must be preceded by a VCOMPONENTS command to define the fixed and random models. (Note, however, that TOBIT does not accommodate spline terms in VCOMPONENTS, nor linear mixed models with complex covariance structures defined by VSTRUCTURE.)

The response variate must be supplied using the Y parameter, and a scalar defining the fixed censoring threshold must be supplied using the BOUND parameter. By default, the data values are assumed to be left-censored (i.e., measurements less than or equal to the value specified by the BOUND parameter are censored). However, right-censoring (i.e., when measurements greater than or equal to the BOUND are censored) can be specified by setting the DIRECTION option to right. Censored observations in Y may be represented either by missing values or by values at or outside the BOUND (i.e., for left-censoring, y-values ≤ BOUND, or, for right-censoring, y-values ≥ BOUND). If missing values are used, an indicator variate, with values of one corresponding to censored observations and values of zero to the non-censored observations, must be supplied using the CENSORED parameter.

The MAXCYCLE, TOLERANCE and RMETHOD options, the INITIAL parameter and the VAOPTIONS procedure can be used to control various aspects of the E-M algorithm performed by TOBIT. The INITIAL parameter provides starting values for the estimates of the censored observations. If available, these may speed up convergence of the E-M algorithm. The values should be below the value specified by the BOUND parameter when DIRECTION=left, or above that value when DIRECTION=right. INITIAL can supply a scalar if a common starting value is to be used for all the censored observations. Alternatively, if different values are required, INITIAL should supply a variate of the same length as Y. Only the values corresponding to censored observations are used, the others are ignored. If INITIAL if not specified, the default is to use the value specified by the BOUND parameter.

The MAXCYCLE option specifies the maximum number of iterations performed by the E-M algorithm (default 30). By default, the E-M algorithm is deemed to have converged if the percentage change in each estimated treatment mean is less than 0.1%, and the percentage change in each estimated variance component is less than 0.05%. However, you can change these tolerance limits by setting the TOLERANCE option to a variate of length two. Its first value specifies the maximum acceptable percentage change for the treatment means, and its second value specifies the maximum acceptable percentage change for the variance components.

The RMETHOD specifies which random terms are used when estimating values for the censored observations during the E-step of the E-M algorithm. With RMETHOD=all, the censored observations are estimated from the fixed effects only, whereas when RMETHOD=final, the censored observations are estimated from the fixed and random effects; default final. Finally, the VAOPTIONS procedure can be used to specify the MAXCYCLE and WORKSPACE options of the REML commands used during the M-step of the E-M algorithm.

Printed output is controlled by the PRINT, VPRINT, and PSE options. The PRINT option has one setting, summary, which prints information on the number of E-M algorithm iterations performed, the percentage of observations censored and the censoring threshold. This is the default, but you can suppress this output by setting option PRINT=*. The VPRINT and PSE options control the printed output from the REML analysis when the censored observations have been replaced by their estimates. The VPRINT option has the same settings as the PRINT option of the REML directive, other than that covariancemodels is excluded; the default is PRINT=model,comp,Wald. Similarly, the setting of PSE are the same as those of the PSE option of the REML directive; the default is PSE=diff.

You can set option PLOT=scatterplot to display a scatter plot of the data, plotting the new y-variate, with censored observations replaced by their estimates, against the observed response variate. When censored observations in Y are entered as missing values, they are plotted at the

value specified by the BOUND parameter; otherwise, they are plotted at the values given in Y. Superimposed onto this plot are a 1-1 line and a horizontal reference line at the censoring threshold defined by the BOUND parameter. By default, no plot is produced.

The NEWY parameter allows you to save a copy of the response variate with the censored observations replaced by their estimates. An indicator variable with values of one corresponding to censored observations in Y and values of zero to non-censored observations can be saved using the YCENSORED parameter. Note, this will be equivalent to any variate supplied by CENSORED. The SAVE parameter can be used to save the save structure from the REML analysis of the data with censored observations replaced by their estimates, for later use by other REML directives and procedures, such as VDISPLAY and VGRAPH.

Options: PRINT, VPRINT, PSE, PLOT, MAXCYCLE, TOLERANCE, RMETHOD, DIRECTION
Parameters: Y, BOUND, CENSORED, INITIAL, NEWY, YCENSORED, SAVE

Method

The E-M (expectation-maximization) algorithm is an iterative two step method to optimize a model. The initial expectation step uses the initial values (either INITIAL, if given, or BOUND) for the censored observations. In the maximization step, the current estimates of the censored values are used in the y-variate in a standard REML analysis to estimate the fitted values and their variances. In subsequent expectation steps, the censored values are estimated as the expected value of the tail of Normal distribution with means and variances for these observations from previous M-step model. The expected deviate in the lower tail of a Normal distribution (x < BOUND) is

```
m - SQRT(v)*PRNORMAL(BOUND;m;v)/CLNORMAL(BOUND;m;v).
```

Action with RESTRICT

Restrictions are not allowed.

References

Amemiya, T. (1984). Tobit models: A survey. *Journal of Econometrics*, **24**, 3-61.

Dempster, A.P., Laird, N.M. & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, **39**, 1-38.

Taylor, J. (1973). The analysis of designed experiments with censored observations. *Biometrics*, **29**, 35-43.

Tobin, J. (1958). Estimation of relationships for limited dependent variables. *Econometrica*, **26**, 24-36.

See also

Directives: REML, VCOMPONENTS, VDISPLAY, VKEEP.
Procedures: CENSOR, VAOPTIONS.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

TRELLIS

Does a trellis plot (S.J. Welham & S.A. Harding).

Options

GROUPS = *factors* or *variate*	Factors or variate defining the classification for the plots
GMETHOD = *string token*	Determines the method used to partition the range when GROUPS is set to a variate (equalspacing, quantiles, distinct, limits); default equal
NGROUPS = *scalar*	Determines the number of plots to be formed when GROUPS is set to a variate and GMETHOD is set to quantiles or equalspacing
LIMITS = *variate*	Limits to use to form groups from a GROUPS variate when GMETHOD=limits
OVERLAP = *scalar*	Proportion by which a GROUPS variate should overlap between plots (scalar in range 0 - 0.5); default 0
OMITEMPTY = *string token*	Whether to omit all empty plots from the array (all), or omit levels of a GROUPS factor where all plots are empty (levels), or keep all plots in the array (none); default level
PENGROUP = *factors*	Defines factor combinations to be plotted in different colours, note that the number of colours available may differ between devices
NROWS = *scalar*	Specifies number of rows of plots to appear on one page; default determined automatically from GROUPS
NCOLUMNS = *scalar*	Specifies number of columns of plots to appear on one page; default determined automatically from GROUPS
TITLE = *text*	Supplies a title for the plot
FIRSTPICTURE = *string token*	Whether to put the first picture at bottom or top left of the grid (bottomleft, topleft); default topl
TMETHOD = *string token*	Whether to give plot titles as factor names with labels or just labels (names, labels); default name
YTITLE = *text*	Supplies an overall y-axis title
XTITLE = *text*	Supplies an overall x-axis title
YMARGIN = *scalar*	Relative size of margins for the y-axis labels on individual plots; default 0.04
XMARGIN = *scalar*	Relative size of margins for the x-axis labels on individual plots; default 0.04
TMARGIN = *scalar*	Relative size of margin for titles of individual plots; default 0.04
PENSIZE = *scalar*	Proportionate adjustment to the pen size for individual plot titles and axis labels; default 1
USEPENS = *string token*	Whether to use current pen definitions in the procedure (no, yes); default no
USEAXES = *string token*	Which aspects of the current axis definitions of window 1 to use (none, limits, style, marks, mpositions, nsubticks, transform); default none
NRMAX = *scalar*	Maximum number of rows on page; default 8 for a square frame, 7 for a landscape frame and 10 for a portrait frame
NCMAX = *scalar*	Maximum number of columns on page; default 8 for a square frame, 10 for a landscape frame and 7 for a

	portrait frame
KEYHEIGHT = *scalar*	Space in y-direction to use for key (0 to suppress key); default * i.e. determined automatically
YPENMETHOD = *string token*	Whether to use the same or different pens for each y-variate (different, same); default diff
FRAMESHAPE = *string token*	Shape of the plotting frame (landscape, portrait, square); default squa

Parameters

Y = *variates*	Y-values of the data to be plotted
X = *variates* or *factors*	X-values of the data to be plotted
METHOD = *string tokens*	Type of plot (point, line, mean, median, histogram, boxplot, spline, schematicboxplot); default poin
DESCRIPTION = *texts*	Annotation for key

Description

TRELLIS plots one or more y-variates for each level generated by the GROUPS option, and arranges these plots in a grid (or trellis) arrangement on the page.

The data to be plotted are specified using the Y parameter. If more than one variate is specified, these will all be displayed on the same plots. This means that e.g. data points can be plotted with means. The type and method of plotting (points, lines, mean values, medians, histograms, boxplots or splines) is specified using the METHOD parameter. The default is METHOD=point. For methods point, line, mean, median and spline, a graph is produced of y-variates against x-variates, which are specified using the X parameter. When METHOD is set to mean or median, a line is drawn to join the mean or median data values at each value of the x-variate for each level of PENGROUP. In any of these cases, if PENGROUPS is set to one or more factors, a different pen will be used for each of the levels of the combined factors. By default, the pen numbers are incremented so that a different set of pens is used for each y-variate. Alternatively, you can set option YPENMETHOD=same, to use the same set for each one.

When METHOD=histogram, a histogram of the data values is drawn in each plot. In this case, options NGROUPS and LIMITS can be used to specify the number of groups in the histogram or the group limits, respectively. If more than one y-variate is specified, parallel histograms will be drawn for the variates. The PENGROUPS option is ignored when METHOD=histogram.

When METHOD=boxplot, a boxplot of the data values is drawn in each plot. Alternatively, you can set METHOD=schematicboxplot to obtain a schematic boxplot, which displays individual outlying points as well as the box; see the BOXPLOT procedure for more details. If you set the PENGROUP option, parallel box plots (one for each level of PENGROUPS) are drawn within every plot. You can also obtain parallel box plots by supplying several y-variates, which are then plotted in parallel in every plot. However, you cannot simultaneously specify several y-variates and set the PENGROUPS option.

The division of the data into separate plots is determined by the setting of the GROUPS option. This can be set to one or more factors, indicating that a separate plot should be drawn for each combinations of the factor levels. The OMITEMPTY option controls what happens if there are no data for some combinations. The default setting levels omits complete levels of any factor for which there are no data points, while the setting all omits all empty plots, i.e. plots where there are no data points. OMITEMPTY=none displays all plots regardless of whether or not they contain any data points.

If the GROUPS option is set to a variate, the plots will show the values of the data for different intervals in the range of the GROUPS variate. The GMETHOD, NGROUPS, LIMITS and OVERLAP options determine how many plots are displayed, and which data points they contain. The default

option of GMETHOD is equalspacing. The distinct setting of GMETHOD converts the variate into a factor with a level (and thus a plot) for each distinct value of the variate. With equalspacing, the groups are defined by dividing the range of the GROUPS variate into the required number of intervals of equal length; while with quantiles, the intervals are defined so that each has an equal number of points, according to the ordering of the GROUPS variate. When GMETHOD is set to equalspacing or to quantiles, the number of groups to form can be specified by the NGROUPS option; if NGROUPS is not set, TRELLIS sets the number to the square root of the number of data values, or to the number of distinct values if this is smaller. Finally, when GMETHOD=limits, the LIMITS option specifies boundaries between the intervals; the first group then contains all data points with values of the GROUPS variate less than the first limit, the second group has all values greater than or equal to the first limit but less than the second limit, and so on.

The OVERLAP option allows the intervals of the GROUPS to overlap. The default overlap is 0, so there is no overlap between plots. If OVERLAP is set to 0.1, then 10% of the points (for PARTITION=quantiles) or 10% of the range (for PARTITION=equalspacing) will be in common between neighbouring plots. OVERLAP can be set anywhere in the range 0 (for no overlap) to 0.5.

The FRAMESHAPE option specifies the shape of the graphics frame, with settings:

landscape	for a frame of size 1.4 × 1.0 i.e. wider in the x- than the y-direction,
portrait	for a frame of size 1.0 × 1.4 i.e. wider in the y- than the x-direction,
square	for a frame of size 1.0 × 1.0.

Some graphics devices do not support the use of device coordinates greater than 1.0, so the default is FRAMESHAPE=square. (See FRAME and DEVICE for more information.)

The default layout on the page can be changed by using NROWS and NCOLUMNS to specify the number of rows of plots on the page, and the number of columns of plots across the page, respectively. By default the layout is arranged so that the area of the page used for plotting is maximized, with a maximum of 8 rows and 8 columns of plots for a square frame, 7 rows and 10 columns for a landscape frame, and 10 rows and 7 columns for a portrait frame. Options NRMAX and NCMAX can be used to override these default maximum numbers of rows and columns of plots, so that more can be produced on a page.

An overall title can be put at the head of each page using the TITLE option, and overall titles for the y- and x- axes can be specified using the YTITLE and XTITLE options respectively. By default the plots start at the top left of the page, but you can set option FIRSTPICTURE=bottomleft to start at the bottom left. When GROUPS is set to one or more factors, the plot titles are constructed by default with the factor name and label/level, but this can be restricted to just the label/level by setting option TMETHOD=label.

The margins and pen size are set to give a reasonable picture on the Windows PC implementation, but can be adjusted using options YMARGIN (space for y-axis labels), XMARGIN (space for x-axis labels), TMARGIN (space for plot titles) and PENSIZE (pen size for axis markings and plot titles).

By default the pen and axes attributes are determined automatically within the procedure. Some predefined attributes can be used, as indicated by the USEPENS and USEAXES options. Setting USEPENS to yes, requests all current pen definitions (for pens 1-29) to be used.

You can specify various aspects of the axes, by defining them for window 1, and indicating that they are to be used by setting the USEAXES option. The following settings are available:

limits	y- and x-axis limits (LOWER and UPPER parameters of XAXIS and YAXIS);
style	axis styles (ACTION parameter of XAXIS and YAXIS, together with the GRID option and BOX parameter of

FRAME);

marks	location and labelling of the tick marks (MARKS, LABELS, LDIRECTION, LROTATION, DECIMALS, DREPRESENTATION, and VREPRESENTATION parameters of XAXIS and YAXIS);
mpositions	positions of the tick marks (MPOSITION parameter of XAXIS and YAXIS); and
nsubticks	number of subticks per interval (NSUBTICKS parameter of XAXIS and YAXIS); and
transform	axis transformations (TRANSFORM parameter of XAXIS and YAXIS).

TRELLIS includes a key on each graphics page for plots other than boxplots if each window of the trellis contains more than plot (i.e. if there is more than one Y variate, or there is a PENGROUPS factor with more than one level). You can use the KEYHEIGHT option to control the size of the key in the y-direction, and setting this to zero will suppress the key. The DESCRIPTION parameter can be used to supply annotation for the key, in the same way as in the DGRAPH directive.

Options: GROUPS, GMETHOD, NGROUPS, LIMITS, OVERLAP, OMITEMPTY, PENGROUP, NROWS, NCOLUMNS, TITLE, FIRSTPICTURE, TMETHOD, XMARGIN, YMARGIN, TMARGIN, PENSIZE, USEPENS, USEAXES, NRMAX, NCMAX, KEYHEIGHT, YPENMETHOD, FRAMESHAPE.
Parameters: Y, X, METHOD, DESCRIPTION.

Action with RESTRICT
TRELLIS takes account of any restriction on Y, X or GROUPS.

Method
TRELLIS uses the standard Genstat facilities for data manipulation and plotting.

See also
Procedures: DTABLE, DMSCATTER, DSCATTER..
Genstat Reference Manual 1 Summary section on: Graphics.

TTEST

Performs a one- or two-sample t-test (S.J. Welham).

Options

PRINT = *string tokens*	Controls printed output (confidence, summary, test, variance, permutationtest); default conf, summ, test, vari
†METHOD = *string token*	Type of test required (twosided, greaterthan, lessthan, equivalence, noninferiority, nonsuperiority); default twos
GROUPS = *factor*	Defines the groups for a two-sample test if only the Y1 parameter is specified
CIPROBABILITY = *scalar*	The probability level for the confidence interval; for a one-sided test this will be for the mean and for a two-sided test for the difference in means; default *, i.e. no confidence interval is produced
NULL = *scalar*	The value of the mean under the null hypothesis; default 0
VMETHOD = *string token*	Selects between the standard two-sample t-test, with a pooled estimate of the variances of the samples, and the use of separate estimates for the sample variances (automatic, pooled, separate); default auto uses a pooled estimate unless there is evidence of unequal variances
PLOT = *string token*	How to plot the statistics from a permutation test (histogram); default * i.e. no plots
NTIMES = *scalar*	Number of random allocations to make when PRINT=perm; default 999
PERMMETHOD = *string token*	Which statistic to use in a permutation test (difference, t); default t
SEED = *scalar*	Seed for the random number generator used to make the allocations; default 0 continues from the previous generation or (if none) initializes the seed automatically
†EQLIMITS = *scalar* or *variate*	Limits for equivalence, non-inferiority or non-superiority

Parameters

Y1 = *variates*	Identifier of the variate holding the first sample
Y2 = *variates*	Identifier of the variate holding the second sample
TESTRESULTS = *variates*	Identifier of variate (length 3) to save test statistic, d.f. and probability value
LOWER = *scalars*	Identifier of scalar to save the lower limit of each confidence interval
UPPER = *scalars*	Identifier of scalar to save the upper limit of each confidence interval
W1 = *variates*	Weights (replications) of the values in Y1; default * i.e. all 1
W2 = *variates*	Weights (replications) of the values in Y2; default * i.e. all 1
SAVEPERMUTATIONS = *variates*	Saves the permutation statistics

Description

The data for TTEST are specified by the parameters Y1 and Y2 and the option GROUPS. For a one-sample test, the Y1 parameter should be set to a variate containing the data. TTEST then performs a one-sample t-test for the mean of a Normal distribution. The value of the mean under the null hypothesis can be specified by the option NULL; by default NULL=0.

The data for a two-sample test can either be specified in two separate variates using the parameters Y1 and Y2. Alternatively, they can be given in a single variate, with the GROUPS option set to a factor to identify the two samples; the GROUPS option is ignored when the Y2 parameter is set. The standard two-sample t-test assumes that the two samples arise from Normal distributions with equal variances and forms a pooled estimate for the variance of both samples. If, however, the variances are unequal, a separate estimate can be used for the variance of each sample. This is known as Welch's t-test or Welch's analysis of variance (Welch 1947). The degrees of freedom of the test are then only approximate (see, for example, Snedecor & Cochran 1989, page 97) but these seem to work well in practice. The VMETHOD option specifies how to estimate the variances for the test. The default setting, automatic, uses a pooled estimate unless there is evidence of unequal variances, pooled always uses a pooled estimate and separate always uses separate estimates. If either pooled or automatic are selected, TTEST will print a warning if there is evidence of inequality of variances.

The W1 and W2 parameters can supply variates of weights to accompany Y1 or Y2, respectively. You can use these to specify replicate observations. For example, instead of specifying variate for Y1 with values (11, 12, 12, 13, 14, 14, 14, 15) you could give Y1 the values (11, 12, 13, 14, 15) together with weight variate W1 containing values (1, 2, 1, 3, 1) indicating the number of replications of each of the values in Y1. The calculation of the t-test assumes that the weights are positive integers defining the replications of the values inside Y1 or Y2 (or zero or missing values to exclude the corresponding values in Y1 or Y2). A warning is given if any positive weight is given that is not an integer.

The METHOD option indicates the type of test to be done, with the following settings:

twosided	does a two-sided test (default);
greaterthan	does a one-sided test of the null hypothesis that mean(Y1) is not greater than mean(Y2) or NULL (for a two-sample or one-sample test, respectively);
lessthan	does a test of the null hypothesis that mean(Y1) is not less than mean(Y2) or NULL;
equivalence	does an equivalence test;
noninferiority	does a non-inferiority test;
nonsuperiority	does a non-superiority test.

A small "p-value" indicates that the data is inconsistent with the null hypothesis. If any sample has fewer than six values, a warning is given that the sample size is too small and the test may not be valid.

For a two-sample equivalence test, the null hypothesis is that the difference between the mean of the first sample and the mean of the second sample lies outside two limits specified, in a variate, by the EQLIMITS option. For a one-sample test, the difference is the mean of the sample minus NULL. TTEST does two tests: first to test whether the difference is outside the lower limit (specified by the first element of the variate), then to test whether it is outside the upper limit. The p-value is the larger of the values from the two tests.

For a two-sample non-inferiority test, the null hypothesis is that the mean of the first sample minus the mean of the second sample is less than the negative value specified, in a scalar, by the EQLIMITS option. For a one-sample test, the null hypothesis is that the mean of the sample minus NULL is less than that value.

For a two-sample non-superiority test, the null hypothesis is that the mean of the first sample minus the mean of the second sample is greater than the positive value specified, in a scalar, by

the EQLIMITS option. For a one-sample test, it is that the mean of the sample minus NULL is greater than that value.

Printed output is controlled by the PRINT option with settings:

summary	number of observations, mean, variance, standard deviation and standard error of mean;
test	t-statistic and probability level;
confidence	confidence interval for the difference between mean and NULL for a one-sample test, or the two means for a two-sample test;
variance	F test for equality of the sample variances in a two-sample test; and
permutationtest	probabilities calculated by a random permutation test (relevant only for two-sample tests).

The default is PRINT=summary,test,confidence,variance. Usually a 95% confidence interval is calculated, but this can be changed by setting the CIPROBABILITY option to the required value (between 0 and 1) or leaving it unset to suppress the interval. For equivalence tests, the confidence interval is an amalgamation of two one-sided intervals, as you are making two one-sided tests. Each limit is therefore calculated for twice the distance from 100% (e.g. 90% instead of 95%, corresponding to a significance level of 5% for the test of equivalence).

By default, for the permutation test, TTEST makes 999 random allocations of the data to the two samples (using a default seed), and determines the probability from the distribution of the t-statistic over these randomly generated data sets. Alternatively, you can set option PERMMETHOD=difference to use the difference between the means instead of the t-statistic. The NTIMES option allows you to request another number of allocations, and the SEED option allows you to specify another seed. TTEST checks whether NTIMES is greater than the number of possible ways in which the data values can be allocated. If so, it does an exact test instead, which takes each possible allocation once. For a visual indication, you can set option PLOT=histogram to display a histogram of the statistics from the permuted data sets, with a vertical line to show the position of the statistic from the original data set.

Results can be saved using the TESTRESULTS, LOWER and UPPER parameters. TESTRESULTS saves the t-statistic, its degrees of freedom and probability level in a variate of length 3. LOWER and UPPER save the lower and upper limits of the confidence interval. The SAVEPERMUTATIONS parameter can save the values of the statistics from the permutation tests in a variate; the final value in the variate is the statistic from the original data set.

Options: PRINT, METHOD, GROUPS, CIPROBABILITY, NULL, VMETHOD, PLOT, NTIMES, PERMMETHOD, SEED.
Parameters: Y1, Y2, TESTRESULTS, LOWER, UPPER, W1, W2, SAVEPERMUTATIONS.

Method

A standard t-statistic is calculated in both cases, together with an F-statistic in the two-sample case (to test equality of variances) as described in any standard textbook. The squared t-statistics and the F-ratio are compared with the appropriate F-distribution using the function FPROBABILITY, and confidence intervals are constructed using the function FED. For the exact test, the allocations are formed using the SETALLOCATIONS directive.

Action with RESTRICT

Y1 and Y2 may be subject to different restrictions; these restrictions will be obeyed. Restrictions are also obeyed on Y1 and GROUPS, allowing RESTRICT to be used for example to limit the data to only one or two groups when the GROUPS factor has more than two levels. Any restrictions on TESTRESULTS will be removed.

References

Snedecor, G.W. & Cochran, W.G. (1989). *Statistical Methods (eighth edition)*. Iowa State University Press, Ames.

Welch, B.L. (1947).The generalization of 'Student's' problem when several different population variances are involved. *Biometrika*, **34**, 28-35.

See also

Procedure: STTEST, AONEWAY, A2WAY, MANNWHITNEY, SIGNTEST, TEQUIVALENCE, WILCOXON.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

TUKEYBIWEIGHT

Estimates means using the Tukey biweight algorithm (D.B. Baird).

Options

CUTPOINT = *scalar*	Cut point after which weight is set to zero; default 5
TOLERANCE = *scalar*	Tolerance to avoid division by zero; default 0.00001

Parameters

DATA = *variates* or *pointers*	Data values
GROUPS = *factors*	Groupings of the data values
MEANS = *variates*	Saves the means
SE = *variates*	Saves standard errors

Description

TUKEYBIWEIGHT estimates means using the Tukey biweight algorithm. This weights the data values depending on how far they are from the median, and discards any that are more than a specified number of median absolute distances away. The number of differences is specified by the CUTPOINT, with a default of 5.

The data values are specified by the DATA parameter. They can be in a single variate, with any groupings specified by the GROUPS parameter. Alternatively, they can be in separate variates, one for each group. The MEANS parameter saves the estimated means, and the SE parameter saves standard errors for the means.

Options: CUTPOINT, TOLERANCE.
Parameters: DATA, GROUPS, MEANS, SE.

Action with RESTRICT

TUKEYBIWEIGHT respects any restrictions on DATA or GROUPS.

See also

Procedures: MPOLISH, ROBSSPM.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

TVARMA

Fits a vector autoregressive moving average (VARMA) model (A.I. Glaser).

Options

`PRINT` = *string tokens*	What to print (`model`, `summary`, `estimates`, `correlations`); default `mode`, `summ`, `esti`
`LIKELIHOOD` = *string token*	Method of likelihood calculation (`exact`, `conditional`); default `exac`
`CONSTANT` = *string token*	How to treat the constant (`estimate`, `fixtozero`); default `esti`
`ARMA` = *variate*	Variate of length two, containing the number of AR and MA parameters respectively
`ARFIXED` = *pointer*	Specifies fixed values of the AR parameters
`MAFIXED` = *pointer*	Specifies fixed values of the MA parameters
`MUFIXED` = *variate*	Specifies fixed values of the constant parameters
`NDIFFERENCING` = *variate* or *scalar*	Specifies the order of differencing for each series; default 0
`NCROSSRESIDUAL` = *scalar*	Number of residual cross-correlation matrices to be computed for calculating the modified portmanteau statistic; default 20
`MAXCYCLE` = *scalar*	Maximum number of iterations; if this is not set, an appropriate default is determined automatically according to the number of parameters
`TOLERANCE` = *scalar*	Convergence criterion; default 0.0001

Parameters

`SERIES` = *pointers*	Time series to be modelled (output series)
`RESIDUALS` = *pointers*	Saves the residual series
`ESTIMATES` = *pointers*	Saves estimates of parameters for each `SERIES` variate
`SEESTIMATES` = *pointers*	Saves standard errors of the estimates
`VCRESIDUALS` = *symmetric matrices*	Variance-covariance matrix of the residuals
`DEVIANCE` = *scalars*	Saves the residual sum of squares or deviance
`CORRELATIONS` = *symmetric matrices*	Saves the correlation matrix of the estimates
`GRADIENTS` = *variates*	Saves the first derivative of the loglikelihood function
`SAVE` = *pointers*	Saves information for use with `TVGRAPH` or `TVFORECAST`

Description

`TVARMA` fits parameters to vector autoregressive moving average (VARMA) time series models. These are natural extensions of the ordinary ARMA models, with the generalization that there may be correlations between as well as within the series.

If $X_t = (x_{1t}, x_{2t} \ldots x_{kt})'$ denotes a vector of k time series for times $t = 1 \ldots n$, then a VARMA(p; q) model can be written as

$$X_t - \mu = \varphi_1 (X_{t-1} - \mu) + \varphi_2 (X_{t-2} - \mu) + \ldots + \varphi_p (X_{t-p} - \mu) + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ldots + \theta_q \varepsilon_{t-q}$$

where $\varepsilon_t = (\varepsilon_{1t}, \varepsilon_{2t}, \ldots \varepsilon_{kt})'$, for $t = 1 \ldots n$ is a vector of k residual series, assumed to be Normally distributed with zero mean and variance-covariance matrix Σ. Each AR parameter (φ) and MA parameter (θ) is a $k \times k$ matrix.

The SERIES parameter lists the variates to which the model is to be fitted. The VARMA model is selected by setting the ARMA option to a variate of length two which contains the number of AR and MA parameters in the model (p and q as noted above). By default, the vector of constants of the k time series is estimated, but they can be fixed at zero by setting option CONSTANT=fixtozero.

The LIKELIHOOD option specifies the criterion that is used to calculate the estimates of the parameters. The default setting, exact, calculates the maximum of the log-likelihood subject to stationarity and invertibility constraints. The alternative setting, conditional, calculates conditional maximum likelihood estimates. This may be useful when the parameter estimates are close to the boundary of the invertibility region.

Printed output is controlled by the PRINT option with settings:

model	brief description of the model, including identifiers and model order;
summary	brief summary of the model, including the log-likelihood;
estimates	parameter estimates and their standard errors; and
correlations	correlations between the parameter estimates of the model.

The AR, MA and constant parameters can be fixed to values supplied by the ARFIXED, MAFIXED and MUFIXED options respectively. The options ARFIXED and MAFIXED should supply a pointer to matrices of size $k \times k$, where k is the number of time series being analysed. The matrices contain the fixed values, with the row and column index indicating which element(s) of the parameter are to be kept fixed, and the pointer suffix indicates in which parameter. For an analysis of four time series ($k = 4$), with $p = 3$ and $q = 1$, the example below sets the value in the first row and second column of the third AR parameter to zero, and the value in the fourth row and second column of the first MA parameter to 0.5.

```
MATRIX [ROWS=4; COLUMNS=4] arf
MATRIX [ROWS=4; COLUMNS=4] maf
CALC arf$[1;2] = 0
CALC maf$[4;2] = 0.5
POINTER [NVALUES=4] arset, maset
CALC arset[3] = arf
CALC maset[1] = maf
```

To set the constant parameters to a fixed value, you should set MUFIXED to a variate of length k, where element i sets the value of the parameter v_i in the mean vector.

The NDIFFERENCING option can be used if the time series being investigated are not stationary, and need to be differenced. It should be set to a variate with an element for each series, indicating the amount of differencing required. This is a variate since it may not be optimal to difference each component of X_t in the same way; see Chatfield (2004) Section 12.4.

Results from the analysis can be saved by using the parameters RESIDUALS, ESTIMATES, SEESTIMATES, VCRESIDUALS, DEVIANCE, CORRELATIONS, GRADIENTS and SAVE. The ESTIMATES and their associated SEESTIMATES are stored in pointers to tables: the first table in the pointer contains the values of v, the next p tables contain the AR parameters, and the final q tables contain the MA parameters.

Options: PRINT, LIKELIHOOD, CONSTANT, ARMA, ARFIXED, MAFIXED, MUFIXED, NDIFFERENCING, NCROSSRESIDUAL, MAXCYCLE, TOLERANCE.

Parameters: SERIES, RESIDUALS, ESTIMATES, SEESTIMATES, VCRESIDUALS, DEVIANCE, CORRELATIONS, GRADIENTS, SAVE.

Action with RESTRICT

The SERIES variates must not be restricted.

Method
TVARMA uses the NAG directive to call routine G13DDF for model fitting, G13DLF for differencing and G13DSF for diagnostic checking.

Reference
Chatfield, C. (2004). The Analysis of Time series, an Introduction (6th edition). Chapman and Hall, London.

See also
Procedures: TVFORECAST, TVGRAPH.
Commands for: Time series.

TVFORECAST

Forecasts future values from a vector autoregressive moving average (VARMA) model (A.I. Glaser).

Options

PRINT = *string tokens*	What to print (forecasts, se); default fore, se
MAXLEAD = *scalar*	Maximum lead time i.e. number of forecasts to be made; default 1

Parameters

FORECASTS = *matrices*	Saves the forecasts
SE = *matrices*	Saves standard errors of the forecasts
SAVE = *pointers*	Save structure from a previous TVARMA

Description

TVFORECAST can be used after the TVARMA procedure to predict future values of a vector time series. The SAVE parameter supplies details of the analysis. If this is not set, TVFORECAST produces forecasts from the most recent TVARMA analysis.

Printed output is controlled by the PRINT option with settings:

forecasts	forecasts from the model, and
se	standard errors of the forecasts.

The number of forecasts to make is specified by the MAXLEAD option. The forecasts and their standard errors can be saved using the FORECASTS and SE parameters.

Options: PRINT, MAXLEAD.
Parameters: FORECASTS, SE, SAVE.

Method

TVFORECAST uses the NAG directive to call routine G13DJF.

See also

Procedures: TVARMA, TVGRAPH.
Commands for: Time series.

TVGRAPH

Plots a vector autoregressive moving average (VARMA) model (A.I. Glaser).

Options

TIMEPOINTS = *variate*	X-coordinates for the graphs; default uses the integers 1, 2...
TITLE = *texts*	Overall title for the graphs
YTITLE = *texts*	Titles for the y-axes; default * forms titles automatically from the identifiers or labels of the y-variables
XTITLE = *texts*	Title for the x-axis in each set of graphs; default * uses the identifier of TIMEPOINTS (if set)
NROWS = *scalar*	Specifies the number of rows of graphs to appear on the graphics screen; default * takes the number of y-variables
NCOLUMNS = *scalar*	Specifies the number of columns of graphs to appear on the graphics screen; default 1

Parameter

SAVE = *pointers*	Save structure from TVARMA with information about the analysis; default plots information from the most recent TVARMA analysis

Description

TVGRAPH plots results from an analysis by the TVARMA procedure. By default this will be from the most recent analysis, but you can use the SAVE parameter to supply results from an earlier analysis (saved using the SAVE parameter of TVARMA).

The TIMEPOINTS option supplies the time points. If this is not set (or if there are at most only two unique values), TVGRAPH uses the integers 1 ... *n*, where *n* is the number of time points in the analysis.

You can use the TITLE option to supply a title for the graphs. If TITLE is not set, no title is displayed.

The YTITLE option supplies a title for the y-axes; this must be set either to a text with a value for each y-variable, or one with a single value (which will then be used for all of them). You can set YTITLE='' to stop a title appearing on the y-axes. If YTITLE is not set, TVGRAPH forms the titles automatically from the identifiers of the series in the TVARMA analysis.

The XTITLE option supplies a title for the x-axes; this must be set to a text with a single value. If XTITLE is not set, TVGRAPH uses the identifier of the TIMEPOINTS option (if specified).

By default, the graphs are plotted in a single column, but this can be altered by using NROWS and NCOLUMNS options to specify the required number of rows and columns respectively. The graphs will be spread over several screens if the values supplied for NROWS and NCOLUMNS, are too small to include all the graphs on a single screen.

Options: TIMEPOINTS, TITLE, YTITLE, XTITLE, NROWS, NCOLUMNS.
Parameter: SAVE.

Action with RESTRICT

DATA variates must not be restricted.

See also
Procedures: TVARMA, TVFORECAST.
Commands for: Time series, Graphics.

TXPAD

Pads strings of a text structure with extra characters so that their lengths are equal (J.T.N.M. Thissen).

Options

PADDINGCHARACTERS = *string token*	Character(s) used for padding; default uses the dot character `'.'`
METHOD = *string token*	Whether the character(s) of PADDINGCHARACTERS should be placed before or after the strings of OLDTEXT (`before`, `after`); default `afte`
REMOVESPACES = *string tokens*	Whether to remove initial and/or trailing spaces in the strings of OLDTEXT (`leading`, `trailing`); default `*` i.e. none

Parameters

OLDTEXT = *texts*	Texts to be padded; must be set
NEWTEXT = *texts*	Saves the padded texts
WIDTH = *scalars*	Sets a limit on the length of the strings in the padded texts; default is the width of the largest string in OLDTEXT

Description

Procedure TXPAD can be used to make the strings of a text of equal length by padding them with extra characters. This may be used to make printed output more readable. The text is specified by the OLDTEXT parameter, and the new text can be saved by the NEWTEXT parameter. If NEWTEXT is not specified, the new text values replace those in OLDTEXT. The length can be specified by the WIDTH parameter, which defaults to the number of characters of the largest string.

The character(s) to be added to the strings with smaller length can be specified by the PADDINGCHARACTERS option, with a default of the dot character(.). The METHOD option specifies whether the characters of PADDINGCHARACTERS are placed before or after the strings of OLDTEXT (default after). The REMOVESPACES option specifies whether leading and/or trailing spaces in the strings of OLDTEXT should be removed; the default is to remove no spaces.

Options: PADDINGCHARACTERS, METHOD, REMOVESPACES.
Parameters: OLDTEXT, NEWTEXT, WIDTH.

Action with RESTRICT

If the OLDTEXT structure is restricted, only the restricted units are modified to have equal widths. If the OLDTEXT structure is restricted and a NEWTEXT structure is specified, the units of NEWTEXT not in the restriction set are set to repeated PADDINGCHARACTERS character(s).

See also

Directives: CONCATENATE, TXCONSTRUCT, TXREPLACE.
Procedure: TXPROGRESSION, TXSPLIT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

TXPROGRESSION

Forms a text containing a progression of strings (R.W. Payne).

Options

INCLUDECHARACTERS = *string tokens*	Defines the set of characters to include in the progression (lower, upper, digits, _, %, space); default lowe
DIRECTION = *string token*	Direction of the progression (ascending, descending); default asce
FIRSTLETTERS = *string token*	Controls which letters come first (alllower, allupper, lower, upper); default uppe
OWNCHARACTERSET = *text*	Can supply an alternative set of characters

Parameters

FIRST = *texts*	Single-valued text specifying the first string in each progression
SECOND = *texts*	Single-valued text specifying the second string in each progression
LAST = *texts*	Single-valued text defining the end of each progression
PROGRESSION = *texts*	Saves the progression

Description

TXPROGRESSION forms a text from a progression of strings. This is saved by the PROGRESSION parameter. It could be used, for example, for labels of factors, or for defining rows and columns of matrices.

The INCLUDECHARACTERS option specifies the characters to include in the progression, with settings:

lower	for lower-case letters (a-z);
upper	for upper-case letters (A-Z);
digits	for the numerical characters 0-9;
_	for the underscore character;
%	for the percent character;
space	for the space character.

If they are all specified, the characters will appear in the order: space, percent, digits 0-9, underscore, and then letters. The default is to include only lower-case letters. The alternative, if you do not like any of these possibilities, is to specify your own set of characters, using the OWNCHARACTERS option.

The FIRSTLETTERS option controls the ordering of lower- and upper-case letters, if both are included, with settings:

alllower	all lower-case letters first;
allupper	all upper-case letters first;
lower	letters interspersed, in pairs, with the lower-case letter first (i.e. a, A, b, B etc.);
upper	letters interspersed, in pairs, with the upper-case letter first (i.e. A, a, B, b etc.).

The default is upper.

The DIRECTION option specifies whether the progression is in ascending order (e.g. a-z) or descending order (e.g. z-a). Ascending order is the default.

The first string in the progression is specified by the FIRST parameter. The SECOND parameter can supply the second string in the progression, thus defining the increment between the strings.

If this is not specified, the default is to increment the right-hand character in the string by one for an ascending progression, and minus one for a descending progression. The LAST parameter defines the end of the progression. (The progression stops when the next string would go beyond LAST.) FIRST, SECOND and LAST must all contain the same number of characters.

Options: INCLUDECHARACTERS, DIRECTION, FIRSTLETTERS, OWNCHARACTERSET.
Parameters: FIRST, SECOND, LAST, PROGRESSION.

Method
The sequence of characters to form the progression is formed using the COUNTER directive.

Action with RESTRICT
Any restrictions are ignored.

See also
Directives: TXBREAK, TXCONSTRUCT, TXFIND, TXPOSITION, TXREPLACE.
Procedure: TXPAD, TXSPLIT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

TXSPLIT

Splits a text into individual texts, at positions on each line marked by separator character(s) (R.W. Payne).

Options

SEPARATOR = *text*	Defines the character(s) that indicate where to split each line; default `','`
INCLUDE = *string tokens*	Whether to retain the separator at the end of a split text, or any spaces at its start and end (`separators`, `spaces`) ; default `*` i.e. include neither

Parameters

TEXT = *texts*	Text to split
SPLITTEXTS = *texts*	Saves the texts into which TEXT is split

Description

TXSPLIT splits a text into individual texts. The positions at which to split each line are marked by the character, or characters, specified by the SEPARATOR option; by default, the separator character is a comma.

By default, TXSPLIT removes the separators between the split texts, as well as any spaces at the start and end of each spit text (i.e. any spaces around the separators, or at the start or end of the original text). The INCLUDE option allows you to request that the separator be left at the end of a split text, and that these spaces should be retained.

The TEXT parameter supplies the text that is to be split. The texts into which it is split are saved, in a pointer, by the SPLITTEXTS parameter.

Options: SEPARATOR, INCLUDE.
Parameters: TEXT, SPLITTEXTS.

Action with RESTRICT

Any restrictions on the original text are ignored.

See also

Directives: TXBREAK, TXCONSTRUCT, TXFIND, TXPOSITION, TXREPLACE.
Procedure: TXPAD, TXPROGRESSION.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

T%CONTROL

Expresses tables as percentages of control cells (R.W. Payne).

Option

PRINT = *string token*	Controls printed output (percentages); default perc

Parameters

OLDTABLE = *tables*	Tables containing the original values
NEWTABLE = *tables*	Tables to store the percentage values
FACTOR = *factors* or *pointers*	Factor, or pointer of factors, with control levels
CONTROL = *scalars*, *vaiates*, *texts* or *pointers*	Identifies the control level or levels of each FACTOR (if more than one is specified for a factor, their mean is used); default uses the reference level

Description

T%CONTROL allows you to express the body of a table as percentages of the values of "control" levels of one or more of its classifying factors. These controls might be standard or check varieties in a variety trial, or placebo treatments in a medical trial, or zero levels of fertilizers in an agricultural field experiment, etc.

You supply the table using the OLDTABLE parameter. You can save a new table containing the percentages using the NEWTABLE parameter. The factors containing the control levels are specified by the FACTOR parameter; if there are several you must put them into a pointer. FACTOR need not be set if the tables are one-way. The CONTROL parameter identifies the control levels of each factor. Usually the factor will have a single control, specified either by giving its level (in a scalar) or its label (in a string or single-valued text). Alternatively, you can define several controls, by specifying a variate (of levels) or a multi-valued text (of labels); T%CONTROL then takes means over the control levels. Again, if there are several factors, you must put the corresponding CONTROL settings into a pointer. If CONTROL is unset or missing for any factor, T%CONTROL uses its reference level.

Not all the factors in the table need to have control levels. Suppose, for example, we have a 2-way table with factors A and B where the first level of A (a_1) is the control. Then the cell (a_i, b_j) will be given as a percentage of the cell (a_1, b_j).

By default T%CONTROL prints the table of percentages, but you can set option PRINT=* to suppress this.

Option: PRINT.
Parameters: OLDTABLE, NEWTABLE, FACTOR, CONTROL.

Method

T%CONTROL uses COMBINE to access the control cells of the table.

See also

Directives: COMBINE, TABLE, TABULATE, MARGIN.
Procedures: MTABULATE, PERCENT, SVSTRATIFIED, SVTABULATE, TABINSERT, TABMODE, TABSORT.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

UNSTACK

Splits vectors into individual vectors according to levels of a factor (R.W. Payne).

Options

DATASET = *factor*	Factor identifying the unstacked data sets
IDSTACKED = *factors*	Factors identifying how the units of the unstacked data sets should be matched
IDUNSTACKED = *factors*	Factors defined to identify these units in the unstacked vectors
MVINCLUDE = *strings*	Which missing values to include (datasets, idstacked); default * i.e. none

Parameters

STACKEDVECTOR = *variates*, *factors* or *texts*
: Vectors to be unstacked

DATASETINDEX = *scalars* or *texts*
: Level or label of the DATASET factor indicating the group whose units are to be stored in the UNSTACKEDVECTOR; default takes the levels of DATASET one at a time (and then recycling this list to match the other parameters)

UNSTACKEDVECTOR = *variates*, *factors* or *texts*
: Unstacked vectors

Description

UNSTACK allows you to split up (or unstack) vectors into individual vectors. The contents of the individual vectors are determined by a factor, specified by the DATASET option. In the simplest case, each original (stacked) vector is split into several new (unstacked) vectors, one for each level of DATASET. The process assumes that the sets are "replicate" sets of data. For example DATASET might correspond to days on which identical sampling schemes were followed. In the most straightforward case, each set contains the same number of observations all stored in an identical order. However, if the observations are in different orders or if some are absent in some of the sets, you can use the IDSTACKED option to specify one or more factors to identify the matching observations within the sets. The IDUNSTACKED option then allows you to save new factors to indicate where the observations are stored in the new (unstacked) vectors. The unstacked vectors are all of the same length, and missing values are inserted for absent observations.

The MVINCLUDE option controls the inclusion of missing values in the unstacked vectors, with the following settings:

idstacked	includes units with missing values for levels of the IDSTACKED factors that do not occur in the data set (otherwise these are omitted), and
datasets	stacked vectors that correspond to data set indexes that do not occur in the data are defined and filled with missing values (otherwise these are left undeclared, and a warning is given).

By default none of these are included.

There are three parameters. STACKEDVECTOR lists the vectors (variates, factors or texts) that are to be split up. DATASETINDEX specifies a level of the DATASET factor for each member of the STACKEDVECTOR list, and UNSTACKEDVECTOR specifies a new vector to store the units of the STACKEDVECTOR corresponding to that DATASETINDEX. So, for example

```
UNSTACK [DATASET=Days] 5(Weight,Height);\
  DATASETINDEX=1,2,3,4,5;\
  UNSTACKEDVECTOR=W1,W2,W3,W4,W5,H1,H2,H3,H4,H5
```

would put the weight measurements made on days 1-5 into `W1`, `W2`, `W3`, `W4` and `W5`, respectively, and the height measurements into `H1`, `H2`, `H3`, `H4` and `H5`. (The construct `5(Weight,Height)` is equivalent to typing `Weight` five times and then `Day` five times, and the `DATASETINDEX` list `1,2,3,4,5` is repeated twice so that it matches the lengths of the other parameter lists.) This method of specification means that you are free to list the vectors and levels in whatever order is most convenient. For example

```
UNSTACK [DATASET=Days] (Weight,Height)5;\
  DATASETINDEX=2(1,2,3,4,5);\
  UNSTACKEDVECTOR=W1,H1,W2,H2,W3,H3,W4,H4,W5,H5
```

lists them in group order rather one stacked vector at a time. If `DATASETINDEX` is not specified, the levels of `DATASET` are taken in order one at a time (and recycled if necessary).

Option: `DATASET`, `IDSTACKED`, `IDUNSTACKED`, `MVINCLUDE`.
Parameter: `STACKEDVECTOR`, `DATASETINDEX`, `UNSTACKEDVECTOR`.

Method
The vectors are unstacked using the standard Genstat manipulation commands, including `SUBSET` and `EQUATE`.

Action with `RESTRICT`
Any restrictions on the vectors are ignored.

See also
Directive: `EQUATE`.
Procedures: `JOIN`, `RESHAPE`, `STACK`, `SUBSET`.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

UTMCONVERSION

Converts between geographical latitude and longitude coordinates and UTM eastings and northings (D.B. Baird).

Options

CONVERTTO = *string token*	Whether to convert to UTM eastings and northings from geographical latitude and longitude coordinates, or to geographical coordinates from UTM (geographical, utm); default utm
DATUM = *string token*	The datum to use when constructing the grid for eastings and northings (WGS84, NAD83, GRS80, OSGB36, WGS72, AUSTRALIAN1965, KRASOVSKY1940, NORTHAMERICAN1927, INTERNATIONAL1924, HAYFORD1909, CLARKE1880, CLARKE1866, AIRY1830, BESSEL1841, EVEREST1830); default WGS8
CENTRALMERIDIAN = *scalar*	Central meridian in degrees for the UTM coordinates
SINGLEZONE = *string token*	Whether to convert to easting and /northings in a single zone (yes, no); default no
EORIGIN = *scalar*	False origin for easting; default 500000
NORIGIN = *scalar*	False origin for northing; default 0

Parameters

LATITUDE = *scalar*s or *variates*	Latitudes
LONGITUDE = *scalars* or *variates*	Longitudes
DIRECTION = *string tokens*	Directions of the angles of latitude and longitude coordinates (NE, NW, SE, SW); default NE
EASTING = *scalars* or *variates*	UTM easting grid references
NORTHING = *scalars* or *variates*	UTM northing grid references
ZONE = *scalars* or *variates*	UTM zones

Description

UTMCONVERSION converts geographical latitude and longitude coordinates to the Universal Transverse Mercator (UTM) coordinate system which uses eastings and northings to represent a point on the earth's surface. These easting and northing coordinates are given on a 1 metre scale, so the distance between two points can be calculated by Pythagoras' theorem in the usual way.

The UTM system is made up of 60 zones which each cover 6 degrees of longitude. It divides the surface of Earth between latitudes 80°S and 84°N into 60 zones (numbered from 1 to 60), each 6° of longitude in width and centred over a meridian of longitude. Zone 1 is bounded by longitude 180° to 174° W and is centred on the 177th West meridian. Zone numbering increases in an easterly direction. Each of the 60 longitude zones in the UTM system is based on a transverse Mercator projection, which is capable of mapping a region of large north-south extent with a low amount of distortion. By using narrow zones of 6° (up to 800 km) in width, and reducing the scale factor along the central meridian by only 0.0004 (to 0.9996, a reduction of 1:2500) the amount of distortion is held below 1 part in 1,000 inside each zone. Distortion of scale increases to 1.0010 at the outer zone boundaries along the equator. In each zone 500,000 is used as the origin of the easting coordinate, at the central meridan.

By default UTMCONVERSION converts from geographical latitude and longitude coordinates to UTM eastings and northings. However, you can set option CONVERTTO=geographical to convert from UTM to geographical coordinates instead.

The LATITUDE parameter specifies or saves the latitudes, and the LONGITUDE parameter

specifies or saves the longitudes. These can be scalars to convert a single coordinate, or variates for several. By default the latitudes and longitudes are assumed to have a north, east orientation, with the latitudes giving the angle from the equator between the North and South poles in the range -90 and 90, and the longitudes giving the angle east or west around the earth from the Greenwich UK Meridian in the range -180 to 180. However, you can specify other orientations using the DIRECTION parameter. This may be a single text value (NE, NW, SE or SW) if all angles have the same orientation, or a text compatible with LATITUDE if the orientation varies over the units. The angles must be given as decimal numbers. You can convert from degrees, minutes and seconds, with the calculation

```
Angle = degrees + minutes/60 + seconds/3600
```

The UTM eastings and northings are saved or specified by the EASTINGS and NORTHINGS parameters, respectively, and the UTM zone by the ZONE parameter. If you are unsure of the zone when converting to latitudes and longitudes, you should convert a latitude and longitude within your set of points not its zone.

Distortion of scale increases as you approach the boundaries between the UTM zones. However, it is often convenient or necessary to measure a series of locations on a single grid, even when some are located in two adjacent zones. Ideally, the coordinates of each position should be measured on the grid for the zone in which they are located. However, as the scale factor is still relatively small near zone boundaries, it is possible to overlap measurements into an adjoining zone when necessary. By setting SINGLEZONE=yes you can force all the points to be mapped into a common zone (the one in which the mean of the longitudes lies).

Historically, several different constants have been used in the UTM projection calculations. By default, UTMCONVERSION uses the standard WGS84 system, which is the same as the NAD83 system. However, you can use the DATUM option to request alternative sets of constants.

The CENTRALMERIDIAN option specifies the central meridian for the UTM coordinates (i.e. the longitude of the UTM origin). If this is unset, the central meridian is taken from the standard UTM zone that is closest to the mean longitude of the data. The EORIGIN option specifies a false origin for easting i.e. the value for the UTM easting along the central meridian; default 500000. Similarly the NORIGIN option specifies a false origin for northing i.e. the value for the UTM northing along the equator (latitude zero); default 0.

Options: CONVERTO, DATUM, SINGLEZONE, CENTRALMERIDIAN, EORIGIN, NORIGIN.
Parameters: LATITUDE, LONGITUDE, DIRECTION, EASTING, NORTHING, ZONE.

Method

UTMCONVERSION uses the 1973 US Army calculations, which are accurate to within less than a metre within each given zone.

Action with RESTRICT

UTMCONVERSION takes account of restrictions on any of the input variates.

Reference

Dutch, S. Converting UTM to Latitude and Longitude (or vice versa)
http://www.uwgb.edu/dutchs/UsefulData/UTMFormulas.HTM.

See also

Procedure: UTMCONVERSION.

VABLOCKDESIGN

Analyses an incomplete-block design by REML, allowing automatic selection of random and spatial correlation models (R.W. Payne).

Options

PRINT = *string tokens*	Controls what summary output is produced about the models (deviance, aic, bic, sic, dffixed, dfrandom, change, exit, best, description); default best, desc
PBEST = *string tokens*	Controls the output from the REML analysis with the best model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
PTRY = *string tokens*	Controls the output to present from the REML analysis used to try each model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
FIXED = *formula*	Fixed model terms; default * i.e. none
RANDOM = *formula*	Additional random model terms; default * i.e. none
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
REPLICATES = *factor*	Replicate factor
BLOCKS = *factor*	Block factor; no default (must be specified)
ROWS = *factor*	Row factor for spatial analysis
COLUMNS = *factor*	Column factor for spatial analysis
ROWCOORDINATES = *variate* or *factor*	Row coordinates for fitting trends and spatial models if the design is irregular; if unset, these are defined from the levels of the ROWS factor
COLCOORDINATES = *variate* or *factor*	Column coordinates for fitting trends and spatial models if the design is irregular; if unset, these are defined from the levels of the COLUMNS factor
PLOTFACTOR = *factor*	Factor numbering the plots in the design; if unset, a local factor is defined automatically
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or variates or y-variates

VCONSTRAINTS = *string token*	Whether to constrain variance components to be positive (none, positive); default none
RSTRATEGY = *string token*	Strategy for selecting the random model (all, allfeasible, optimal, automatic, full); default allf
METHOD = *string token*	Criterion to choose the best random model (aic, sic, bic); default sic
TRYSPATIAL = *string token*	Whether to try spatial models (always, ifregular); default * i.e. no spatial models
TRYTRENDS = *string token*	Whether to see whether row and column trends are needed in the fixed model (yes, no); default no
SPATIALFACTOR = *factor*	Factor to use to define the term for a two-dimensional power-distance model; if unset, a local factor is defined automatically

Parameters

Y = *variates*	Response variates
BESTMODEL = *pointers*	Saves a model-definition structure for the best model for each y-variate
EXIT = *scalars*	Exit status of the best model for each y-variate
SAVE = *REML save structures*	Save structure from the analysis of the best model for each y-variate

Description

VABLOCKDESIGN analyses data from an incomplete-block design by REML. An *incomplete-block design* is one where the blocks each have too few units to contain one of each of the treatments. In the context of a REML analysis, the treatment factors are usually the fixed factors. So this is to say that the blocks are unable each to contain a unit with every combination of levels of the fixed factors.

Some designs are *resolvable*. The blocks can then be grouped together into subsets in which each treatment is replicated once. These groupings of blocks thus form replicates, which may be useful while the experiment is taking place. For example, if several operators are needed to make observations in a field trial, it is usual to get each one to observe the plots of a complete replicate. Then any operator differences will be included in the between-replicate variation, and will not add to the variability of the treatment estimates. Of course it can be useful to include a replicate factor even if the "replicates" are not exact, e.g. if some of the treatments do not occur at every level of the replicate factor.

The RSTRATEGY option selects the strategy to use to determine the random model, with the following settings.

all	fits the full random model (replicates and blocks within replicates if a replicate factor has been specified, or just blocks if there are no replicates).
allfeasible	tries to fit the full random model. If this is not possible, it tries models removing one, and then two random terms, until successful.
optimal	tries all feasible random models.
full	synonym of all.
automatic	synonym of optimal.

The full random model should reflect the way in which the treatments were randomized onto the experiment, so it is generally best to use this. The default of RSTRATEGY=allfeasible, will do this if possible, or use a simpler random model if REML is unable to fit the full model. Note: VABLOCKDESIGN regards a model as successful, if the REML directive returns an exit status of

zero (i.e. successful fitting) and there are no bound or aliased variance parameters.

The BLOCKS option must specify the block factor, and the replicate factor (if any) is specified by the REPLICATES option. If you want to fit spatial covariance models, you must specify row and column factors, using by the ROWS and COLUMNS options respectively. If the replicates are adjacent to each other in the field and you want to fit spatial covariance models across the whole field, rather than within each replicate, you should define the levels of the row and column factors to run across the experiment. Otherwise they should be defined within replicates (i.e. using the same numbers within each replicate). The spatial models will then be fitted within replicates.

You can use the ROWCOORDINATES and COLCOORDINATES options to specify variates or factors giving the actual positions of the plots in the field. These are needed if you want to fit row or column trends (i.e. covariates) in the fixed model, or to fit a power-distance covariance model when the plots are on an irregular grid. If the levels of the ROWS and COLUMNS factors are defined across the whole experiment rather than within replicates, their values are used as defaults if ROWCOORDINATES and COLCOORDINATES are not set. Their values are also used as defaults if ROWCOORDINATES or COLCOORDINATES are set to variates or factors with no values; the variates or factors are then defined to contain those values.

The PLOTFACTOR option allows you to specify a factor to index the plots, which is used to define the null random model (i.e. the one with no block or replicate effects), or to include a random term for measurement error when fitting covariance models. If this is not set, a local factor called plots is set up automatically.

The FIXED option specifies the fixed terms to be fitted in the analysis. The default fixed model consists of just the constant term, which then becomes the grand mean. The constant term can be omitted by setting option CONSTANT=omit, provided a fixed model has been specified. The FACTORIAL option sets a limit on the number of factors and variates allowed in each fixed term (default 3); any term containing more than that number is deleted from the model. The RANDOM option allows you to specify any extra random terms to include (in addition to replicates and blocks-within-replicates). The VCONSTRAINTS option allows you to constrain the variance components to be positive; by default they are not constrained.

The TRYSPATIAL option indicates whether to try fitting spatial models, with settings:

always	always tries to fit them,
ifregular	fit them only if the plots are on a regular grid.

With the default, TRYSPATIAL=*, no spatial models are fitted. For a regular grid, VAROWCOLUMNDESIGN tries models with order 1 auto-regressive structures on the rows and/or the columns of the design, provided there are more than four rows or columns, respectively. For an irregular grid, if there are more than four rows and more four columns, it tries an anisotropic power-distance model using city-block distance. Otherwise, if there is only one dimension with more than four coordinates, it tries an isotropic power-distance model.

The SPATIALFACTOR option allows you to specify a factor to use to define the term required for a two-dimensional power-distance model. If this is not set, a local factor called RowColumn2d is used.

You can set option TRYTRENDS=yes to see whether row and column trends (i.e. covariates) are needed in the fixed model. By default this is not done.

The MVINCLUDE option controls whether units with missing values in the explanatory factors and variates and/or the y-variate are included in the analysis, as in the REML directive.

The METHOD option specifies how to assess the random (and spatial) models

aic	uses their Akaike information coefficients,
sic or bic	uses their Schwarz (Bayesian) information coefficients (default).

The PRINT option specifies the summary output to be produced about the models. The settings are mainly the same as those of the VRACCUMULATE procedure (which is used to store and then

print details of the analyses). There is an extra setting, `description`, to provide a description of the model and strategy. There is also a setting, `best`, to print the description of the best model. By default, `PRINT=best,description`.

The `PBEST` option specifies the output to be produced from the `REML` analysis with the best model. Similarly, the `PTRY` option indicates what output should be produced for each candidate random model when it is tried. Their settings are mainly the same as those of the `PRINT` option of the `REML` directive. There are also extra settings `aic` and `sic` (with a synonym `bic`) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively. The default for both these options is to produce no output.

The `PTERMS` option operates as in `REML`, to specify the terms whose means and effects are printed by `PBEST` and `PTRY`; the default is all the fixed terms. Likewise, the `PSE` option controls the type of standard error that is displayed with the means and effects; the default is to give a summary of the standard errors of differences.

The `Y` parameter specifies the response variate. A model-definition structure for the best model can be saved, in a pointer, by the `BESTMODEL` parameter; the `VMODEL` procedure can use this to define the model (using the `VCOMPONENTS` and `VSTRUCTURE` directives) so that you can reanalyse it yourself using the `REML` directive. Alternatively, you can save the `REML` save structure from the analysis with the best model using the `SAVE` parameter. The `EXIT` parameter allows you to save a code from `REML`, giving the "exit status" of the fit (zero if successful).

Options: `PRINT`, `PBEST`, `PTRY`, `FIXED`, `RANDOM`, `CONSTANT`, `FACTORIAL`, `REPLICATES`, `BLOCKS`, `ROWS`, `COLUMNS`, `ROWCOORDINATES`, `COLCOORDINATES`, `PLOTFACTOR`, `PTERMS`, `PSE`, `MVINCLUDE`, `VCONSTRAINTS`, `RSTRATEGY`, `METHOD`, `TRYSPATIAL`, `TRYTRENDS`, `SPATIALFACTOR`.
Parameters: `Y`, `BESTMODEL`, `EXIT`, `SAVE`.

Method
Model definition structures are defined for the various candidate models. (Run the example to see those that are considered for a resolvable block design.) The `VARANDOM` procedure is used to fit them, with the `VRACCUMULATE` procedure storing the necessary details for the best one to be selected.

See also
Directives: `REML`, `VCOMPONENTS`, `VSTRUCTURE`.
Procedures: `VAOPTIONS`, `VARANDOM`, `VARECOVER`, `VAROWCOLUMNDESIGN`, `VASERIES`, `VALINEBYTESTER`, `VFMODEL`, `VFSTRUCTURE`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VAIC

Calculates the Akaike and Schwarz (Bayesian) information coefficients for REML (R.W. Payne & V.M. Cave).

Options

PRINT = *string tokens*	Controls printed output (deviance, aic, bic, sic, dffixed, dfrandom, changes); default aic
INCLUDE = *string tokens*	When LMETHOD=residual, which constants to include that depend only on the fixed model (determinant, pi); default pi
DMETHOD = *string token*	Method to use to calculate log(determinant(X′X)) (choleski, lrv); default chol
LMETHOD = *string token*	Whether the residual or full log-likelihood is used to calculate the information coefficients (residual, full); default resi
REPEAT = *string token*	Whether to repeat output from the previous VAIC (yes, no); default no

Parameters

DEVIANCE = *scalars*	Saves the deviance
AIC = *scalars*	Saves the Akaike information coefficient
SIC = *scalars*	Saves the Schwarz (Bayesian) information coefficient
DFFIXED = *scalars*	Saves the number of parameters fitted in the fixed model
DFRANDOM = *scalars*	Saves the number of parameters fitted in the random model (and any covariance models)
CHANGES = *variates*	Saves changes since the previous VAIC; the units of the variates are labelled by the names of the coefficients (deviance, aic, sic, dffixed and dfrandom)
SAVE = *REML save structures*	Save structure for which to calculate the coefficients; default uses the save structure from the most recent REML

Description

The Akaike and Schwarz (Bayesian) information coefficients are often used to compare the fit of models. Traditionally the residual log-likelihood is used to form the information coefficients, and these can be used to assess the appropriateness of random (and covariance) models in REML. However, for REML models with different fixed effects, the residual log-likelihoods are not comparable and hence information coefficients derived from the residual log-likelihood cannot be used. To compare models that differ in their fixed effects, information coefficients formed using the full log-likelihood evaluated at the REML estimates must be used. The LMETHOD option is thus provided to control whether the information coefficients are formed using the residual log-likelihood (the default) or the full log-likelihood.

When LMETHOD=residual, the information coefficients are calculated from the REML deviance:

$$\text{aic} = \text{deviance} + 2 \times r$$

$$\text{sic} = \text{deviance} + \log(n - p) \times r$$

where n is the total number of usable units in the analysis, r is the number of parameters fitted in the random model (and any covariance models), and p is the number of parameters fitted in the fixed model. An additional consideration is that the REML deviance omits some constants that depend on the fixed model. In fact the full deviance is given by

$$\text{full deviance} = \text{deviance} + (n-p)*\log(2\pi) - \log(\det(X'X))$$

where X is the design matrix of the fixed model. Other software systems tend to include the first

term, involving π, but omit the log-determinant term which is more time-consuming to calculate. The inclusion of these terms in the calculation is controlled by the `INCLUDE` option, with settings

`determinant`	$-\log(\det(X'X))$
`pi`	$+(n-p)*\log(2\pi)$

The `DMETHOD` option controls how $-\log(\det(X'X))$ is calculated when this is included. However, the default is `INCLUDE=pi`.

When `LMETHOD=full`, the information coefficients are calculated by:

aic = deviance + 2 × (*r* + *p*) + *logdet*

sic = deviance + log(*n*) × (*r* + *p*) + *logdet*

where *logdet* is the log-determinant of the variance-covariance matrix for the full set of fixed and random effects. See Verbyla (2019) for more details. The options `INCLUDE` and `DMETHOD` are not relevant, and are ignored.

Printed output is controlled by the `PRINT` option, with settings:

`deviance`	prints the deviance (adding the extra terms specified by `INCLUDE` when `LMETHOD=residual`);
`aic`	prints the Akaike information coefficient;
`bic` or `sic` (synonyms)	print the Schwarz (Bayesian) information coefficient;
`dffixed`	prints the number of parameters fitted in the fixed model;
`dfrandom`	prints the number of parameters fitted in the random model (and any covariance models);
`changes`	prints changes in the values of the coefficients since the previous use of `VAIC` (provided the fixed model of the `REML` analysis has not also changed when `LMETHOD=residual`).

These can all be saved using the `DEVIANCE`, `AIC`, `SIC`, `DFFIXED`, `DFRANDOM` and `CHANGES` parameters. By default `VAIC` prints just the Akaike information coefficient.

By default, each time that you use `VAIC`, its record of the current and previous `REML` analyses is updated. However, you can set option `REPEAT=yes` to repeat output from the previous `VAIC`. The analysis record is then not updated, so the information required to calculate changes remains available.

The coefficients are usually calculated for the most recent `REML` analysis. However, you can use the `SAVE` parameter to specify the save structure from an earlier analysis.

Options: `PRINT`, `INCLUDE`, `DMETHOD`, `LMETHOD`, `REPEAT`.
Parameters: `DEVIANCE`, `AIC`, `SIC`, `DFFIXED`, `DFRANDOM`, `CHANGES`, `SAVE`.

References

Verbyla, A.P. (2019). A note on model selection using information criteria for general linear models estimated using REML. *Australia & New Zealand Journal of Statistics*, **61**, 39-50.

See also

Directives: `REML`, `VKEEP`.
Procedure: `VRACCUMULATE`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VALINEBYTESTER

Provides combinabilities and deviances for a line-by-tester trial analysed by VABLOCKDESIGN or VAROWCOLUMNDESIGN (R.W. Payne).

Options

PRINT = *string tokens*	Controls what summary output is produced about the models (combinability, tests); default comb, test
LINES = *factor*	Specifies the line (usually female parent); no default (must be specified)
TESTERS = *factor*	Specifies the tester (usually male parent); no default (must be specified)
CONTROLS = *factor*	Distinguishes between control and test (line × tester) genotypes; default is that there are no controls
PCOMBINABILITYTERMS = *formula*	Terms whose combinability effects are to be printed (LINES and/or LINES.TESTERS; default is to print both of them
MVINCLUDE = *string tokens*	When the SAVE parameter is not set, this specifies whether to include units with missing values in the explanatory factors and variates and/or the y-variates in the analyses (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or variates or y-variates

Parameters

Y = *variates*	Response variates
MODELSTRUCTURE = *pointers*	Model-definition structure used for the analysis of each y-variate
COMBINABILITY = *pointers*	Pointer to tables of combinability effects for each y-variate
SECOMBINABILITY = *pointers*	Pointer to tables of standard errors of combinability effects for each y-variate
DEVIANCES = *variates*	Saves deviances for LINES and LINES.TESTERS
SAVE = *REML save structures*	Save structure from the analysis of each y-variate

Description

VALINEBYTESTER provides further output for a line-by-tester trial, already analysed using VABLOCKDESIGN or VAROWCOLUMNDESIGN. These are trials in which several "lines" (usually female parents) are all mated with a smaller number of testers (usually male parents). Generally, all combinations of parent will be present in the trial, but incomplete arrangements, such as diallels, are also possible. Control, or check, genotypes may also be present.

The factor used to define the lines must be specified by the LINES option. Similarly, the testers factor must be specified by the TESTERS option. If there are any control genotypes, their factor must be specified by the CONTROLS option; this should have a different level for each control genotype, and a single level for all the line-by-tester genotypes.

The PRINT option specifies the output to be produced, with settings:

combinability	to print the BLUPs for LINES (i.e. SCA) and/or LINES.TESTERS (i.e. GCA), within CONTROLS if specified, and
tests	to print deviances for LINES, LINES.TESTERS.

You can set the PCOMBINABILITYTERMS option to a model formula specifying exactly which

of the combinability terms you want; by default, both are printed.

You can define the analysis with either the MODELSTRUCTURE or the SAVE parameter or, preferably, both. The SAVE parameter specifies the REML save structure from the analysis. This is used to obtain the combinability effects. The MODELSTRUCTURE parameter specifies the model-definition structure for the analysis. This is needed to calculate the deviances, and also allows a save structure to be formed if SAVE is not set. If both MODELSTRUCTURE and SAVE are specified, VALINEBYTESTER aims to check that the save structure genuinely corresponds to the supplied model structure, in case of mistakes. However, the checks are not foolproof, so you do need to be careful.

When the SAVE parameter is unset, the MVINCLUDE specifies whether to include units with missing values in the explanatory factors and variates and/or the y-variates in the analyses. (When SAVE is set, the units to include can be determined from the save structure.)

The Y parameter specifies the response variate. The COMBINABILITY parameter can save a pointer to tables containing the combinability BLUPs, requested by the PCOMBINABILITYTERMS option. Similarly, the SECOMBINABILITY parameter can save a pointer to tables containing the standard errors of the BLUPs. The DEVIANCES parameter can save the deviances printed by the test setting of PRINT, in a variate.

Options: PRINT, LINES, TESTERS, CONTROLS, PCOMBINABILITYTERMS, MVINCLUDE.
Parameters: Y, MODELSTRUCTURE, COMBINABILITY, SECOMBINABILITY, DEVIANCES, SAVE.

See also
Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VABLOCKDESIGN, VAROWCOLUMNDESIGN, VLINEBYTESTER.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VALLSUBSETS

Fits all subsets of the fixed terms in a REML analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (results); default resu
FORCED = *formula*	Terms to include in every model
FACTORIAL = *scalar*	Limit for expansion of FORCED terms; default 3
SELECTION = *string tokens*	One or two criteria to be printed with the models (r2, adjusted, cp, ep, aic, sic, bic, rss, rms); default aic, sic
NBESTMODELS = *scalar*	Number of models to print; default * i.e. all
BESTMODEL = *pointer*	Saves the best model according to the selected criteria
RESULTS = *pointer*	Pointer to save variates containing the criteria for the sets, and F and Wald statistics for the terms that they contain
MARGINALTERMS = *string token*	How to treat terms that are marginal to other terms (forced, free); default forc
SAVE = *REML save structure*	Specifies the analysis whose fixed terms are to be tested; by default this will be the most recent REML

No parameters

Description

VALLSUBSETS fits all subsets of the fixed terms in a REML analysis. It does this by a generalized regression analysis, with a weight matrix based on the variances estimated from the REML analysis (i.e. with the full fixed model). The subsets are thus assessed using identical estimates of the variance components, allowing statistics such as the Akaike information criterion to be used to assess which subset may be best.

By default, VALLSUBSETS uses the most recent REML analysis. However, you can take an earlier analysis, by using the SAVE option of VALLSUBSETS to specify its save structure (saved using the SAVE parameter of the earlier REML command).

The subsets are formed from all the fixed terms, but you can use the FORCED option to specify terms that should always be included. Terms that are marginal to another fixed term are usually also treated as forced. However, you can set option MARGINALTERMS to free to retain them in the "free" terms that are used to form the subsets. Note that VALLSUBSETS considers only models that obey the principle of marginality. This states that a model that includes an interaction term must also include all its marginal terms. For example, a model that includes the interaction A.B must also include the main effects A and B.

The SELECTION option selects one or two criteria to be printed with the sets, with the settings:

r2	% sum of squares accounted for (taking the total sum of squares as the residual from the forced model),
adjusted	% variance accounted for (compared to the residual mean square from the forced model),
cp	Mallows Cp,
ep	mean squared error of prediction,
aic	Akaike information criterion,
sic or bic	Schwarz (Bayesian) information criterion,
rss	residual sum of squares, and
rms	residual mean square.

For more details, see the RSEARCH procedure (which is used to do the analyses). VALLSUBSETS reports which subset is best, according to each of the selected criteria. The default selects the

Akaike and Schwarz (Bayesian) information criteria.

In addition to the selected criteria, the output shows the number of degrees of freedom fitted in the subset, and probabilities assessing the effect of dropping each of its terms from the subset. The probabilities are obtained from F statistics if the denominator degrees of freedom are available from the original REML analysis. Otherwise they are based on Wald statistics. Terms that are marginal to another term in the subset cannot be dropped. This is indicated by printing marg instead of a probability. Also, terms that are aliased are indicated by printing aaa. By default, all the subsets are printed, but you can set the NBESTMODELS to a scalar, *n* say, to print only the *n* best subsets according to the first criterion specified by the SELECTION option.

The results are printed by default. However, you can set option PRINT=* if you want only to save them, using the RESULTS option. This saves a pointer containing variates storing all the available criteria and the numbers of degrees of freedom, then the Wald statistics for the terms, followed by their probabilities, and then the F statistics and their probabilities.

You can also use the BESTMODEL option to save the best model according to each of the selected criteria. It saves them in a pointer containing either one or two model formulae (according to the number of selected criteria). The formulae are stored in the order in which the criteria were specified by the SELECTION option, and are labelled in the pointer by the names of the criteria.

Options: PRINT, FORCED, FACTORIAL, SELECTION, NBESTMODELS, BESTMODELS, RESULTS, MARGINALTERMS, SAVE.
Parameters: none.

Method

VALLSUBSETS defines a weighted regression, with weight matrix given by the inverse of the unit-by-unit variance-covariance matrix (obtained using the UVCOVARIANCE option of VKEEP). It then calls the RSEARCH procedure to fit the subsets.

Action with RESTRICT

Any restriction applied to vectors used in the REML analysis will apply also to the results from VALLSUBSETS.

See also

Directive: REML.
Procedures: RSEARCH, VRFIT, VSCREEN.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VAMETA

Performs a REML meta analysis of a series of trials, previously analysed by VASERIES (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default mode, comp, Wald
PTRY = *string tokens*	Controls the output to present from the REML analysis used to try each model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
PRECOVERY = *string tokens*	Controls what summary output is produced about the models that are tried during recovery (deviance, aic, bic, sic, dffixed, dfrandom, change, exit, best); default devi, aic, sic, dfra, best
FIXED = *formula*	Fixed model terms; if unset, these are taken from the MODELSTRUCTURES
RANDOM = *formula*	Additional random model terms; default * i.e. none
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
RECOVER = *string token*	Whether to try to recover with a simpler random model if REML cannot fit the model (yes, no); default no
METHOD = *string token*	How to choose the best model during recovery (aic, sic, bic); default sic

Parameters

Y = *variates*	Response variates
MODELDEFINITIONS = *pointers*	Descriptions of the models for each y-variate, saved from VASERIES
EXIT = *scalars*	Exit status for the fit (zero if successful)
SAVE = *vsaves*	REML save structure from the analysis of each y-variate

Description

VAMETA can perform a REML meta analysis of a series of trials with either incomplete-block or row-and-column designs. The trials must previously have been analysed by the VASERIES procedure, to determine the best random model to use with each trial. Details of the models must be saved using the MODELDEFINITIONS parameter of VASERIES, and then supplied to VAMETA using its own MODELDEFINITIONS parameter. However, you can redefine the fixed model to fit in the meta analysis, and the action to take with the constant term (estimate or omit), by

setting the FIXED and CONSTANT options, respectively. The FACTORIAL option sets a limit on the number of factors and variates allowed in each term defined by FIXED (default 3). You can also use the RANDOM option to specify some additional random terms to include in the analysis. Note: these terms are removed, if necessary, from the random terms selected by VASERIES to be fitted independently for any trial.

The PRINT option specifies the output to be produced from the analysis. The settings are mainly the same as those of the PRINT option of the REML directive but with extra settings aic and sic (with a synonym bic) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively. The default is to print model descriptions, estimated variance components and Wald or F tests for fixed effects.

The Y parameter specifies the response variate. The SAVE parameter can save pointer containing a REML save structure from the analysis that can be used e.g. to display further output using the VDISPLAY directive. The EXIT parameter allows you to save a code from REML, giving the "exit status" of the fit (zero if successful).

The random models in meta analysis can become complicated, and REML may be unable to achieve a successful fit if there are more random terms than are actually needed to explain the random variation.(The REML likelihood may be too flat for any clear optimum to be found.) You can guard against this situation by setting option RECOVER=yes. VAMETA then tries models removing first one random term (and any associated spatial model), then two and so on, until successful. Note: it regards a model as successful, if the REML directive returns an exit status of zero (i.e. successful fitting) and there are no bound or aliased variance parameters.

The METHOD option specifies how to choose the random (and spatial) model if there is more than one possible model with the same number of random terms removed:

aic	uses their Akaike information coefficients,
sic or bic	uses their Schwarz (Bayesian) information coefficients (default).

The PRECOVERY option specifies the summary output to be produced about the models that are fitted during recovery. The settings are mainly the same as those of the VRACCUMULATE procedure (which is used to store and then print details of the analyses). There is an extra setting, best, to print the description of the best model. The default is to print the best description, together with the deviance, the Akaike and Schwarz (Bayesian) information coefficients and the number of degrees, for all the models. The PTRY option, with the same settings as PRINT, controls output from each individual analysis.

The PTERMS option operates as in REML, to specify the terms whose means and effects are printed by PRINT and PTRY; the default is all the fixed terms. Likewise, the PSE option controls the type of standard error that is displayed with the means and effects; the default is to give a summary of the standard errors of differences.

Options: PRINT, PTRY, PRECOVERY, FIXED, RANDOM, CONSTANT, FACTORIAL, PTERMS, PSE, RECOVER, METHOD.
Parameters: Y, MODELDEFINITIONS, EXIT, SAVE.

Method

The VRMETAMODEL procedure is used to define the random model for the meta analysis, if there are random terms that need to be fitted for only some of the trials. The VRESIDUAL directive is used to define spatial covariance models if required in any of the trials.

See also
Directives: REML, VDISPLAY, VKEEP, VRESIDUAL.
Procedures: VASERIES, VMETA, VRMETAMODEL.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VAOPTIONS

Defines options for the fitting of models by VARANDOM and associated procedures (R.W. Payne).

Options

MAXCYCLE = *scalar*	Limit on the number of iterations in REML analyses; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm
MINSPATIALCOORDINATES = *scalar*	Minimum number of different coordinates in a direction for a spatial model to be fitted by VAROWCOLUMNDESIGN; default 4
LIMPRTREND = *scalar*	Critical value for the probability of a row or column trend in the initial basic REML analysis (with replicates but no other random terms) for this to be included in the later analyses) by VAROWCOLUMNDESIGN; default 0.01
REPORTFAILURES = *string token*	Whether the accumulated summary should include models that fail to fit or that have bound variance parameters (yes, no) ; default no

No parameters

Description

There are several procedures, with the prefix VA, that investigate potential models for a REML analysis. For example, VABLOCKDESIGN does this for data from an incomplete-block design, and VAROWCOLUMNDESIGN does it for data from a field trial arranged in rows and columns. These two procedures both use a general procedure VARANDOM, which allows you to try several alternative random models for a REML analysis, and then select the best one according to either their Akaike or Schwarz (Bayesian) information coefficients.

VABLOCKDESIGN and VAROWCOLUMNDESIGN are designed to be used by non-expert users, and so various decisions are made there (and in VARANDOM) about how the analysis is to be done and how the models are to be selected. This procedure, VAOPTIONS, is provided to allow more experienced users to modify some of the options that control the process.

The MAXCYCLE and WORKSPACE options are relevant to all the procedures, and define the settings for the MAXCYCLE and WORKSPACE options of the REML directive, when this is used in any of the procedures.

Other options are relevant to specific procedures. The MINSPATIALCOORDINATES and LIMPRTREND options control aspects of the analyses in VAROWCOLUMNDESIGN. MINSPATIALCOORDINATES sets a limit on the number of different row or column coordinates for a spatial model to be fitted in that direction (default 4). LIMPRTREND defines the critical value for the probability of a row or column trend in an initial basic REML analysis, that is performed with replicates but no other random terms, to decide whether this is be included in the later analyses (default 0.01). The REPORTFAILURES option controls whether the accumulated summary, printed from VARANDOM, should include models that fail to fit or that have bound variance parameters.

Options: MAXCYCLE, WORKSPACE, MINSPATIALCOORDINATES, LIMPRTREND, REPORTFAILURES.
Parameters: none.

See also
Directive: REML.
Procedures: VABLOCKDESIGN, VAROWCOLUMNDESIGN, VARANDOM.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VARANDOM

Finds the best REML random model from a set of models defined by VFMODEL (R.W. Payne).

Options

PRINT = *string tokens*	Controls what summary output is produced about the models (deviance, aic, bic, sic, dffixed, dfrandom, change, exit, best); default devi, aic, sic, dfra, best
PBEST = *string tokens*	Controls the output from the REML analysis with the best model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
PTRY = *string tokens*	Controls the output to present to present from the REML analysis used to try each model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
MODELSTRUCTURES = *pointer*	Model-definition structures specifying the models to try
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or variates or y-variates
METHOD = *string token*	How to choose the best model (aic, sic, bic); default sic
PROHIBIT = *string token*	Whether to exclude models where any estimated variance parameters are held at a bound (bound); default *

Parameters

Y = *variates*	Response variates
NBESTMODEL = *scalars*	Saves the number of the best model for each y-variate, returning a missing value if no models could be fitted successfully
SAVE = *REML save structures*	Save structure from the analysis of the best model for each y-variate

Description

VARANDOM allows you to try several alternative random models for a REML analysis, and then select the best one according to either their Akaike or Schwarz (Bayesian) information coefficients.

Model-definition structures for the models to be assessed must be specified using the MODELSTRUCTURES option. These are formed using the VFMODEL and VFSTRUCTURE procedures, which define the aspects controlled by the VCOMPONENTS and VSTRUCTURE

directives, respectively.

The response variate for the analysis must be specified by the Y parameter. The number of the best model can be saved, in a scalar, by the NBESTMODEL parameter; it returns a missing value if no models could be fitted successfully The REML save structure from the analysis with the best model can be saved using the SAVE parameter.

The MVINCLUDE option controls whether units with missing values in the explanatory factors and variates and/or the y-variate are included in the analysis, as in the REML directive.

The METHOD option specifies how to assess the models

aic	uses their Akaike information coefficients,
sic or bic	uses their Schwarz (Bayesian) information coefficients (default).

You can set option PROHIBIT = bound, to excludes models with any estimated variance parameters held at a bound.

The PRINT option specifies the summary output to be produced about the models. The settings are mainly the same as those of the VRACCUMULATE procedure (which is used to store and then print details of the analyses). There is also an extra setting best, which prints the description of the best model. The default is to print the best description, together with the deviance, the Akaike and Schwarz (Bayesian) information coefficients and the number of degrees, for all the random models.

The PBEST option specifies the output to be produced from the REML analysis with the best model. Similarly, the PTRY option indicates what output should be produced for each candidate random model when it is tried. Their settings are mainly the same as those of the PRINT option of the REML directive. There are also extra settings aic and sic (with a synonym bic) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively. The default for both these options is to produce no output.

The PTERMS option operates as in REML, to specify the terms whose means and effects are printed by PBEST and PTRY; the default is all the fixed terms. Likewise, the PSE option controls the type of standard error that is displayed with the means and effects; the default is to give a summary of the standard errors of differences.

Options: PRINT, PBEST, PTRY, MODELSTRUCTURES, PTERMS, PSE, MVINCLUDE, METHOD, PROHIBIT.
Parameters: Y, NBESTMODEL, SAVE.

See also

Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VAOPTIONS, VARECOVER, VFMODEL, VFSTRUCTURE, VMODEL, VRACCUMULATE.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VARECOVER

Recovers when REML, is unable to fit a model, by simplifying the random model (R.W. Payne).

Options

PRINT = *string tokens*	Controls what summary output is produced about the simpler random models that are tried (deviance, aic, bic, sic, dffixed, dfrandom, change, exit, best); default devi, aic, sic, dfra, best
PBEST = *string tokens*	Controls the output from the REML analysis with the best simpler model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
PTRY = *string tokens*	Controls the output to present to present from the REML analysis used to try each model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
PLOTFACTOR = *factor*	Factor numbering the plots in the design, required if VARECOVER needs to try a null random model; if unset, a local factor is defined automatically
FORCED = *formula*	Specifies terms that must not be removed from the random model; by default any of the random terms can be removed
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or variates or y-variates
METHOD = *string token*	Criterion to choose the best model (aic, sic, bic); default sic
PROHIBIT = *string token*	Whether to exclude models where any estimated variance parameters are held at a bound (bound); default *

Parameters

Y = *variates*	Response variates
MODELSTRUCTURE = *pointers*	Model-definition structure for the unsuccessful analysis of each y-variate
BESTMODEL = *pointers*	Saves a model-definition structure for the best model for each y-variate
EXIT = *scalars*	Exit status of the best model for each y-variate
SAVE = *REML save structures*	Save structure from the analysis of the best model for

each y-variate

Description

VARECOVER can be used to recover after an unsuccessful attempt to fit a REML model. Usually this can be resolved by omitting non-significant random terms – there may be too little information about these term in the REML likelihood for the algorithm to find the optimum. VARECOVER automates the process of finding a simpler model that can be fitted successfully. First it tries random models that omit one term from the random model (and if there is a correlation model defined on the omitted random term, that will be omitted too). Then, if none of these models can be fitted, it tries models that omit two random terms, and so on, until eventually a null random model may have to be fitted. If there is more than one candidate model available from those that omit the same number of random terms, VARECOVER chooses the one with the smallest Akaike or Schwarz (Bayesian) information coefficient, according to the setting of the METHOD option. If you set option PROHIBIT = bound, VARECOVER excludes models with any estimated variance parameters held at a bound: i.e. the fitting of these models is also regarded as unsuccessful. You can also use the FORCED option to specify any terms that must not be dropped from the random model (e.g. because you want to estimate their BLUPs).

The PRINT option specifies the summary output to be produced about the models that are tried. The settings are mainly the same as those of the VRACCUMULATE procedure (which is used to store and then print the information). However, there is also a setting, best, to print the description of the best model (i.e. the simplified model that has been chosen). By default, PRINT=best.

The PBEST option specifies the output to be produced from the REML analysis with the best model. Similarly, the PTRY option indicates what output should be produced for each candidate random model when it is tried. Their settings are mainly the same as those of the PRINT option of the REML directive. There are also extra settings aic and sic (with a synonym bic) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively. The default for both these options is to produce no output.

The PTERMS option operates as in REML, to specify the terms whose means and effects are printed by PBEST and PTRY; the default is all the fixed terms. Likewise, the PSE option controls the type of standard error that is displayed with the means and effects; the default is to give a summary of the standard errors of differences.

The PLOTFACTOR option allows you to specify a factor to index the plots, which will used if it is necessary to try the null random model. If this is not set, a local factor called plots is set up automatically.

The MVINCLUDE option controls whether units with missing values in the explanatory factors and variates and/or the y-variate are included in the analysis, as in the REML directive.

The Y parameter specifies the response variate, and the MODELSTRUCTURE parameter specifies a model-definition structure defining the model used in the unsuccessful REML analysis. A model-definition structure for the best of the simplified models can be saved, in a pointer, by the BESTMODEL parameter; the VMODEL procedure can use this to define the model (using the VCOMPONENTS and VSTRUCTURE directives) so that you can reanalyse it yourself using the REML directive. Alternatively, you can save the REML save structure from the analysis with the best model by using the SAVE parameter. The EXIT parameter can save a scalar containing the REML exit status of the best model for each y-variate; see VKEEP for details.

Options: PRINT, PBEST, PTRY, PLOTFACTOR, FORCED, PTERMS, PSE, MVINCLUDE, METHOD, PROHIBIT.

Parameters: Y, MODELSTRUCTURE, BESTMODEL, EXIT, SAVE.

Method
Model definition structures are defined for the various candidate models. The VARANDOM procedure is used to fit them, with the VRACCUMULATE procedure storing the necessary details for the best one to be selected.

See also
Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VAOPTIONS, VARANDOM, VFMODEL, VFSTRUCTURE, VMODEL.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VAROWCOLUMNDESIGN

Analyses a row-and-column design by REML, with automatic selection of the best random and spatial covariance model (R.W. Payne).

Options

PRINT = *string tokens*	Controls what summary output is produced about the models (deviance, aic, bic, sic, dffixed, dfrandom, change, exit, best, description); default best, desc
PBEST = *string tokens*	Controls the output from the REML analysis with the best model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
PTRY = *string tokens*	Controls the output to present from the REML analysis used to try each model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
FIXED = *formula*	Fixed model terms; default * i.e. none
RANDOM = *formula*	Additional random model terms; default * i.e. none
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
REPLICATES = *factor*	Replicate factor, if relevant
ROWS = *factor*	Row factor; default * i.e. must be specified
COLUMNS = *factor*	Column factor; default * i.e. must be specified
ROWCOORDINATES = *variate* or *factor*	Row coordinates for fitting trends and spatial models if the design is irregular; if unset, these are defined from the levels of the ROWS factor
COLCOORDINATES = *variate* or *factor*	Column coordinates for fitting trends and spatial models if the design is irregular; if unset, these are defined from the levels of the COLUMNS factor
PLOTFACTOR = *factor*	Factor numbering the plots in the design; if unset, a local factor is defined automatically
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or variates or y-variates
VCONSTRAINTS = *string token*	Whether to constrain variance components to be positive

	(`none`, `positive`); default `none`
`RSTRATEGY` = *string token*	Strategy for selecting the random model (`all`, `allfeasible`, `set`, `setfeasible`, `fastoptimal`, `optimal`, `automatic`, `comprehensive`, `full`, `given`); default `allf`
`METHOD` = *string token*	Criterion to choose the best random model (`aic`, `sic`, `bic`); default `sic`
`TRYSPATIAL` = *string token*	Whether to try spatial models (`always`, `ifregular`); default `*` i.e. no spatial models
`TRYTRENDS` = *string token*	Whether to see whether row and column trends are needed in the fixed model (`yes`, `no`); default `no`
`SPATIALFACTOR` = *factor*	Factor to use to define the term for a 2-dimensional power-distance model; if unset, a local factor is defined automatically

Parameters

`Y` = *variates*	Response variates
`BESTMODEL` = *pointers*	Saves a model-definition structure for the best model for each y-variate
`EXIT` = *scalars*	Exit status of the best model for each y-variate
`SAVE` = *REML save structures*	Save structure from the analysis of the best model for each y-variate

Description

`VAROWCOLUMNDESIGN` allows you to try various random and covariance models for a `REML` analysis of data from a row-and-column design, and select the best one according to either their Akaike or Schwarz (Bayesian) information coefficients.

A *row-and-column design* is a design where the plots are set out in a rectangular grid. Often this is a regular grid, where the rows and columns are equally spaced and there are no gaps, but irregular arrangements can be handled too. Some designs are *resolvable*. The field can then be divided into sections in which each treatment is replicated once. These replicates can be useful while the experiment is taking place. For example, if several operators are needed to make observations of the plots, it is usual to get each one to observe the plots of a complete replicate. Then any operator differences will be included in the between-replicate variation, and will not add to the variability of the treatment estimates. Of course it can be useful to include a replicate factor even if the "replicates" are not exact, e.g. if some of the treatments do not occur at every level of the replicate factor. The replicate factor, if available, is specified by the `REPLICATES` option.

The row and column factors are specified by the `ROWS` and `COLUMNS` options respectively. If the replicates are adjacent to each other in the field and you want to fit spatial covariance models across the whole field, rather than within each replicate, you should define the levels of the row and column factors to run across the experiment. Otherwise they should be defined within replicates (i.e. using the same numbers within each replicate). The spatial models will then be fitted within replicates.

You can use the `ROWCOORDINATES` and `COLCOORDINATES` options to specify variates or factors giving the actual positions of the plots in the field. These are needed if you want to fit row or column trends (i.e. covariates) in the fixed model, or to fit a power-distance covariance model when the plots are on an irregular grid. If the levels of the `ROWS` and `COLUMNS` factors are defined across the whole experiment rather than within replicates, their values are used as defaults if `ROWCOORDINATES` and `COLCOORDINATES` are not set. Their values are also used as defaults if `ROWCOORDINATES` or `COLCOORDINATES` are set to variates or factors with no values;

the variates or factors are then defined to contain those values.

The PLOTFACTOR option allows you to specify a factor to index the plots (which is needed to include a random term for measurement error). If this is not set, a local factor called plots is set up automatically.

The FIXED option specifies the fixed terms to be fitted in the analysis. The default fixed model consists of just the constant term, which then becomes the grand mean. The constant term can be omitted by setting option CONSTANT=omit, provided a fixed model has been specified. The FACTORIAL option sets a limit on the number of factors and variates allowed in each fixed term (default 3); any term containing more than that number is deleted from the model.

The RANDOM option allows you to specify any extra random terms to include (in addition to replicates, rows or rows-within-replicates and columns or columns-within-replicates). The VCONSTRAINTS option allows you to constrain the variance components to be positive; by default they are not constrained.

The RSTRATEGY option selects the strategy to use to determine the random model, with the following settings.

all	fits the full random model, i.e. replicates, rows within replicates and columns within replicates if REPLICATES is set, or rows and columns otherwise. This is appropriate if the row and column factors played a key role in the design and its randomization. For example, some factors may have been applied to complete rows or complete columns, as in a strip-block design.
allfeasible	tries to fit the full random model. If this is not possible, it tries models removing first one random term, then two and so on, until successful.
set	simply uses the random model (if any) defined by the RANDOM option. This is useful when you know the random model and want to investigate the effect of adding spatial covariance models.
setfeasible	tries to fit the random model defined by the RANDOM option. If this is not possible, it tries models removing first one random term, then two and so on, until successful.
fastoptimal	follows an automatic strategy that aims to find the best random model without having to fit all of them. So, for example, it does not try models that include a column main effect as well as a spatial covariance model along rows.
optimal	tries all feasible random models. This may take a while, and so may be best left for the occasions when you are unsure what to do, or want to check the result from an automatic search.
full	synonym of all.
given	synonym of set.
automatic	synonym of fastoptimal.
comprehensive	synonym of optimal.

VAROWCOLUMNDESIGN regards a model as successful, if the REML directive returns an exit status of zero (i.e. successful fitting) and there are no bound or aliased variance parameters. The default is RSTRATEGY=allfeasible.

The TRYSPATIAL option indicates whether to try fitting spatial models, with settings:

always	always tries to fit them,
ifregular	fit them only if the plots are on a regular grid.

With the default, TRYSPATIAL=*, no spatial models are fitted. For a regular grid,

VAROWCOLUMNDESIGN tries models with order 1 auto-regressive structures on the rows and/or the columns of the design, provided there are more than four rows or columns, respectively. For an irregular grid, if there are more than four rows and more four columns, it tries an anisotropic power-distance model using city-block distance. Otherwise, if there is only one dimension with more than four coordinates, it tries an isotropic power-distance model.

The SPATIALFACTOR option allows you to specify a factor to use to define the term required for a two-dimensional power-distance model. If this is not set, a local factor called RowColumn2d is used.

You can set option TRYTRENDS=yes to see whether row and column trends (i.e. covariates) are needed in the fixed model. By default this is not done.

The MVINCLUDE option controls whether units with missing values in the explanatory factors and variates and/or the y-variate are included in the analysis, as in the REML directive.

The METHOD option specifies how to choose the best random model

aic	uses their Akaike information coefficients,
sic or bic	uses their Schwarz (Bayesian) information coefficients (default).

The PRINT option specifies the summary output to be produced about the models. The settings are mainly the same as those of the VRACCUMULATE procedure (which is used to store and then print details of the analyses). There is an extra setting, description, to provide a description of the model and strategy. There is also a setting, best, to print the description of the best random model. By default, PRINT=best,description.

The PBEST option specifies the output to be produced from the REML analysis with the best model. Similarly, the PTRY option indicates what output should be produced for each candidate random model when it is tried. Their settings are mainly the same as those of the PRINT option of the REML directive. There are also extra settings aic and sic (with a synonym bic) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively. The default for both these options is to produce no output.

The PTERMS option operates as in REML, to specify the terms whose means and effects are printed by PBEST and PTRY; the default is all the fixed terms. Likewise, the PSE option controls the type of standard error that is displayed with the means and effects; the default is to give a summary of the standard errors of differences.

The Y parameter specifies the response variate. A model-definition structure for the best model can be saved, in a pointer, by the BESTMODEL parameter; the VMODEL procedure can use this to define the model (using the VCOMPONENTS and VSTRUCTURE directives) so that you can reanalyse it yourself using the REML directive. Alternatively, you can save the REML save structure from the analysis with the best model using the SAVE parameter. The EXIT parameter allows you to save a code from REML, giving the "exit status" of the fit (zero if successful).

Options: PRINT, PBEST, PTRY, FIXED, RANDOM, CONSTANT, FACTORIAL, REPLICATES, ROWS, COLUMNS, ROWCOORDINATES, COLCOORDINATES, PLOTFACTOR, PTERMS, PSE, MVINCLUDE, VCONSTRAINTS, RSTRATEGY, METHOD, TRYSPATIAL, TRYTRENDS SPATIALFACTOR.
Parameters: Y, BESTMODEL, EXIT, SAVE.

Method

Model definition structures are defined for various candidate models, involving rows, columns, measurement error, replicates (if specified) and spatial models (if requested) are defined using the VFMODEL procedure. (Run the example to see those that are considered for a resolvable, regular row-and-column design.) The VARANDOM procedure is used to fit them, with the VRACCUMULATE procedure storing the necessary details for the best one to be selected.

See also
Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VABLOCKDESIGN, VAOPTIONS, VARANDOM, VARECOVER, VASERIES, VALINEBYTESTER, VFMODEL.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VASDISPLAY

Displays further output from an analysis by VASERIES (R.W. Payne).

Options

PRINT = *string tokens*	What output to present (model, components, effects, means, stratumvariances, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default mode, comp, Wald, cova
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
CFORMAT = *string token*	Whether printed output for covariance models gives the variance matrices or the parameters (variancematrices, parameters); default vari
FMETHOD = *string token*	Controls whether and how to calculate F-statistics for fixed terms (automatic, none, algebraic, numerical); default auto
MODELDEFINITIONS = *pointer*	Definitions of the models used by VASERIES
SAVE = *pointer*	REML save structures from the VASERIES analysis

Parameter

EXPERIMENT = *scalars* or *texts*	Specifies the experiment, from the series, whose output is to be displayed; no default, must be set

Description

The VASDISPLAY procedure allows you to display further output from the analysis of a series of experiments, analysed by VASERIES. The model definitions and REML save structures from the analysis can be specified by the MODELDEFINITIONS and SAVE options. If either of these is not specified, the output is taken from the most recent VASERIES analysis.

The EXPERIMENT parameter specifies the experiment from the series, whose output is to be displayed. This can be set to a scalar, or to a single-valued text if the experiments factor has labels.

The options PRINT, PTERMS, PSE, CFORMAT and FMETHOD operate as in VDISPLAY, except that PRINT has an additional settings aic and sic (with a synonym bic) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively.

Options: PRINT, PTERMS, PSE, CFORMAT, FMETHOD, MODELDEFINITIONS, SAVE.
Parameter: unnamed.

See also

Directives: REML, VDISPLAY.
Procedures: VASERIES, VASKEEP.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VASERIES

Analyses a series of trials with incomplete-block or row-and-column designs by REML, automatically selecting the best random models (R.W. Payne).

Options

PRINT = *string tokens*	Controls what summary output is produced about the models (deviance, aic, bic, sic, dffixed, dfrandom, change, exit, best, summary); default devi, aic, sic, dfra, best
PBEST = *string tokens*	Controls the output from the REML analysis with the best model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
PTRY = *string tokens*	Controls the output to present to present from the REML analysis used to try each model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic); default * i.e. none
FIXED = *formula*	Fixed model terms; default * i.e. none
RANDOM = *formula*	Additional random model terms; default * i.e. none
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
EXPERIMENTS = *factor*	Experiment factor
REPLICATES = *factor*	Replicate factor, if required
BLOCKS = *factor*	Block factor, if required
ROWS = *factor*	Row factor, if required
COLUMNS = *factor*	Column factor, if required
ROWCOORDINATES = *variate* or *factor*	Row coordinates for fitting trends and spatial models if the design is irregular; if unset, these are defined from the levels of the ROWS factor
COLCOORDINATES = *variate* or *factor*	Column coordinates for fitting trends and spatial models if the design is irregular; if unset, these are defined from the levels of the COLUMNS factor
PLOTFACTOR = *factor*	Factor numbering the plots in the design; if unset, a local factor is defined automatically
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or

	variates or y-variates
VCONSTRAINTS = *string token*	Whether to constrain variance components to be positive (none, positive); default none
RSTRATEGY = *string token*	Strategy for selecting the random model (all, allfeasible, fastoptimal, optimal); default allf
METHOD = *string token*	How to choose the best random model (aic, sic, bic); default sic
TRYSPATIAL = *string token*	Whether to try spatial models (always, ifregular); default * i.e. no spatial models
TRYTRENDS = *string token*	Whether to see whether row and column trends are needed in the fixed model (yes, no); default no
SPATIALFACTOR = *factor*	Factor to use to define the term for a 2-dimensional power-distance model; if unset, a local factor is defined automatically

Parameters

Y = *variates*	Response variates
MODELDEFINITIONS = *pointers*	Saves definitions of the best models for use by VAMETA
EXIT = *variates*	Exit status of the best models (zero if successful)
SAVE = *pointers*	REML save structures for the best analysis of each experiment

Description

VASERIES performs mixed-model analyses of a series of trials with either incomplete-block or row-and-column designs

An *incomplete-block design* is one where the blocks each have too few units to contain one of the treatments. In the context of a REML analysis, the treatment factors are the fixed factors. So this is to say that the blocks are unable each to contain a unit with every combination of levels of the fixed factors.

A *row-and-column design* is a design used for field trials, where the plots are set out in a rectangular grid. Often this is a regular grid, where the rows and columns are equally spaced, and there are no gaps, but irregular arrangements can be handled too.

VASERIES analyses each trial (or experiment), using REML, trying a range of appropriate random models. It then selects the best one according to either their Akaike or Schwarz (Bayesian) information coefficients.

The EXPERIMENTS option specifies a factor to identify the individual trials (or experiments). The block factor is specified by the BLOCKS option, and the row and column factors are specified by the ROWS and COLUMNS options respectively. If all the experiments have incomplete-block designs, the ROWS and COLUMNS options need not be specified, and BLOCKS need not be specified if they are all row-and-column designs. If there is a mixture, the row and column factors should either have only one level or missing values in each of the block designs, and the block factor should have only one level or missing values in each row-and-column design.

You can use the ROWCOORDINATES and COLCOORDINATES options to specify variates or factors giving the actual positions of the plots in a row-and-column design. These are needed if you want to fit row or column trends (i.e. covariates) in the fixed model, or to fit a spatial covariance model when the plots are on an irregular grid. The values of the ROWS and COLUMNS factors are used as defaults, if ROWCOORDINATES and COLCOORDINATES are not set. Their values are also used as defaults if ROWCOORDINATES or COLCOORDINATES are set to variates or factors with no values; the variates or factors are then defined to contain those values.

The PLOTFACTOR option allows you to specify a factor to index the plots, which is needed to specify a power-distance model, or to include a measurement-error term when fitting spatial

models to plots on a regular grid. If this is not set, a local factor called `plots` is set up automatically.

Some designs are *resolvable*. The field can then be divided into sections in which each treatment is replicated once. These replicates can be useful while the experiment is taking place. For example, if several operators are needed to make observations of the plots, it is usual to get each one to observe the plots of a complete replicate. Then any operator differences will be included in the between-replicate variation, and will not add to the variability of the treatment estimates. Of course it can be useful to include a replicate factor even if the "replicates" are not exact, e.g. if some of the treatments do not occur at every level of the replicate factor.

The replicate factor is specified by the `REPLICATES` option. Note, it is assumed that the blocks, rows and columns are still numbered across the experiment, rather than within replicates.

The `FIXED` option specifies the fixed terms to be fitted in the analysis. The default fixed model consists of just the constant term, which then becomes the grand mean. The constant term can be omitted by setting option `CONSTANT=omit`, provided a fixed model has been specified. The `FACTORIAL` option sets a limit on the number of factors and variates allowed in each fixed term (default 3); any term containing more than that number is deleted from the model. The `RANDOM` option allows you to specify any extra random terms to include (in addition to replicates and blocks-within-replicates). The `VCONSTRAINTS` option allows you to constrain the variance components to be positive; by default they are not constrained.

The `TRYSPATIAL` option indicates whether to try fitting spatial models for row-and-column designs, with settings:

`always`	always tries to fit them,
`ifregular`	fit them only if the plots are on a regular grid.

With the default, `TRYSPATIAL=*`, no spatial models are fitted. For a regular grid, `VRCBEST` tries models with order 1 auto-regressive structures on the rows and/or the columns of the design, provided there are more than four rows or columns, respectively. For an irregular grid, if there are more than four rows and more four columns, it tries an anisotropic power-distance model using Euclidean distance. Otherwise, if there is only one dimension with more than four coordinates, it tries an isotropic power-distance model (again using Euclidean distance).

The `SPATIALFACTOR` option allows you to specify a factor to use to define the term required for a two-dimensional power-distance model. If this is not set, a local factor called `RowColumn2d` is used.

You can set option `TRYTRENDS=yes` to see whether row and column trends (i.e. covariates) are needed in the fixed model for a row-and-column design. By default this is not done.

The response variate for the analysis must be specified by the `Y` parameter. A model-definition structure for the best model can be saved, in a pointer, by the `BESTMODEL` parameter; the `VMODEL` procedure can use this to define the model (using the `VCOMPONENTS` and `VSTRUCTURE` directives) so that you can reanalyse it yourself using the `REML` directive. Alternatively, you can save the `REML` save structure from the analysis with the best model using the `SAVE` parameter.

The `MVINCLUDE` option controls whether units with missing values in the explanatory factors and variates and/or the y-variate are included in the analysis, as in the `REML` directive.

The `RSTRATEGY` option selects the strategy to use to determine the random model for each trial, with the following settings.

`all`	fits the full random model. This is appropriate if the random factors played a key role in the design and its randomization. For example, some factors may have been applied to complete rows or complete columns of a row-and-column-design (as in a strip-block design).
`allfeasible`	tries to fit the full random model. If this is not possible, it tries models removing first one random term, then two and so on, until successful.

`fastoptimal`	follows an automatic strategy that aims to find the best random model for a row-and-column design without having to fit all of them. So, for example, it does not try models that include a row main effect as well as a spatial covariance model along rows. This setting is the same as the `optimal` setting for a block design
`optimal`	tries all feasible random models. With row-and-column designs this may take a while, and so may be best left for the occasions when you are unsure what to do, or want to check the result from an automatic search.

`VASERIES` regards a model as successful, if the `REML` directive returns an exit status of zero (i.e. successful fitting) and there are no bound or aliased variance parameters. The default is `RSTRATEGY=allfeasible`.

The `METHOD` option specifies how to assess the random (and spatial) models

`aic`	uses their Akaike information coefficients,
`sic` or `bic`	uses their Schwarz (Bayesian) information coefficients (default).

The `PRINT` option specifies the summary output to be produced about the models. The settings are mainly the same as those of the `VRACCUMULATE` procedure (which is used to store and then print details of the analyses). There is also an extra setting `best` which prints the description of the best model, and a setting `summary` which summarizes all the best models at the end of the output. The default is to print the best description, together with the deviance, the Akaike and Schwarz (Bayesian) information coefficients and the number of degrees of all the random models.

The `PBEST` option specifies the output to be produced from the `REML` analysis with the best model. Similarly, the `PTRY` option indicates what output should be produced for each candidate random model when it is tried. Their settings are mainly the same as those of the `PRINT` option of the `REML` directive. There are also extra settings `aic` and `sic` (with a synonym `bic`) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively. The default for both these options is to produce no output.

The `PTERMS` option operates as in `REML`, to specify the terms whose means and effects are printed by `PBEST` and `PTRY`; the default is all the fixed terms. Likewise, the `PSE` option controls the type of standard error that is displayed with the means and effects; the default is to give a summary of the standard errors of differences.

The `Y` parameter specifies the response variate. The `SAVE` parameter can save pointer containing a `REML` save structure from the analysis of the best model for each experiment, so that you can generate further output. The `MODELDEFINITIONS` parameter can save a pointer to define the models, that can be used by the `VAMETA` procedure to produce a meta analysis combining information from all the experiments. `MODELDEFINITIONS[0]` stores the various factors and variates involved in the models, and `MODELDEFINITIONS[i]` is a model-definition structure for the best model for the *i*th experiment (see the `VFMODEL` and `VFSTRUCTURE` procedures for details). The `EXIT` parameter allows you to save variate containing a code from `REML` for each experiment, giving the "exit status" of the fit (zero if successful).

You can use procedure `VASDISPLAY` to display further output from the analyses of any of the experiments, and procedure `VASKEEP` to save information into Genstat data structures.

Options: `PRINT`, `PBEST`, `PTRY`, `FIXED`, `RANDOM`, `CONSTANT`, `FACTORIAL`, `EXPERIMENTS`, `REPLICATES`, `BLOCKS`, `ROWS`, `COLUMNS`, `ROWCOORDINATES`, `COLCOORDINATES`, `PLOTFACTOR`, `PTERMS`, `PSE`, `MVINCLUDE`, `VCONSTRAINTS`, `RSTRATEGY`, `METHOD`, `TRYSPATIAL`, `TRYTRENDS`, `SPATIALFACTOR`.

Parameters: `Y`, `MODELDEFINITIONS`, `EXIT`, `SAVE`.

Method
The VABLOCKDESIGN procedure is used to decide on the best model for experiments with incomplete-block designs, and the VAROWCOLUMNDESIGN procedure is used for those with row-and-column designs.

See also
Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VABLOCKDESIGN, VAMETA, VAOPTIONS, VAROWCOLUMNDESIGN, VASDISPLAY, VASKEEP, VASMEANS, VMODEL.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VASKEEP

Copies information from an analysis by VASERIES into Genstat data structures (R.W. Payne).

Options

EXPERIMENT = *scalar* or *text*	Specifies the experiment, from the series, whose output is to be saved; no default, must be set
FACTORIAL = *scalar*	Limit on the number of factors or covariates in the terms generated from the TERMS parameter; default 3
RESIDUALS = *variate*	Residuals from the analysis
FITTEDVALUES = *variate*	Fitted values from the analysis
DEVIANCE = *scalar*	Residual deviance from fitting the full fixed model
DF = *scalar*	Residual degrees of freedom after fitting the full fixed model
AIC = *scalar*	Saves the Akaike information coefficient
SIC = *scalar*	Saves the Schwarz (Bayesian) information coefficient
RMETHOD = *string token*	Which random terms to use when calculating RESIDUALS (final, all); default fina
FMETHOD = *string token*	Controls how to calculate F-statistics for fixed terms (automatic, none, algebraic, numerical); default auto
WMETHOD = *string token*	Controls which Wald statistics are saved (add, drop); default drop
MODELDEFINITIONS = *pointer*	Definitions of the models used by VASERIES
SAVE = *pointer*	REML save structures from the VASERIES analysis

Parameters

TERMS = *formula*	Terms for which information is to be saved
COMPONENTS = *scalars*	Estimated variance components
MEANS = *tables*	Table of predicted means for each term
SEDMEANS = *symmetric matrices*	Standard errors of differences between the predicted means
VARMEANS = *symmetric matrices*	Variance-covariance matrix of the means
EFFECTS = *tables*	Table of estimated regression coefficients for each term
SEDEFFECTS = *symmetric matrices*	Standard errors of differences between the estimated parameters of each term
VAREFFECTS = *symmetric matrices*	Variance-covariance matrix of the effects of a term
WALD = *scalars*	Wald statistic (fixed terms only)
FSTATISTIC = *scalars*	F statistics (fixed terms only)
NDF = *scalars*	Numerator d.f. (fixed terms only)
DDF = *scalars*	Denominator d.f. (fixed terms only)

Description

VASKEEP copies results from the analysis of a series of experiments, analysed by VASERIES, into Genstat data structures. The model definitions and REML save structures from the analysis can be specified by the MODELDEFINITIONS and SAVE options. If either of these is not specified, the output is taken from the most recent VASERIES analysis.

The EXPERIMENT option specifies the experiment from the series, whose output is to be saved. This can be set to a scalar, or to a single-valued text if the experiments factor has labels.

VASKEEP caters for only the most commonly required types of output, as most of the output can be saved by VKEEP in the usual way: the SAVE parameter of VASERIES saves a pointer of REML save structures, one for each experiment in their order in the levels of the experiments

factor. However, VASKEEP has the advantage that it takes account of the additional complication that, to analyse each experiment, VASERIES uses subset factors and variates containing only the data from that experiment. So, VASKEEP automatically makes the conversion from these subset vectors to those in the full data set. The variates of residuals and fitted values will thus have the same number of units as the original y-variate, with missing values in the units belonging to the other experiments. Similarly the tables of means and effects will have values for all levels of relevant factors, with missing values for those that were absent in the current experiment. An exception, though, is that the symmetric matrices of standard errors of differences or variances will have units only for comparisons between means or effects for levels of factors that actually occurred in the experiment.

Options RESIDUALS, FITTEDVALUES, DEVIANCE, DF, RMETHOD, FMETHOD, WMETHOD, and parameters TERMS, COMPONENTS, MEANS, SEDMEANS, VARMEANS, EFFECTS, SEDEFFECTS, VAREFFECTS, WALD, FSTATISTIC, NDF, DDF operate as in VKEEP In addition. there is a FACTORIAL option to set a limit on the number of factors or covariates in the terms generated from the TERMS parameter (default 3). There are also options AIC, SIC to save the Akaike and Schwarz (Bayesian) information coefficients, respectively.

Options: EXPERIMENT, FACTORIAL, RESIDUALS, FITTEDVALUES, DEVIANCE, DF, AIC, SIC, RMETHOD, FMETHOD, WMETHOD, MODELDEFINITIONS, SAVE.
Parameters: TERMS, COMPONENTS, MEANS, SEDMEANS, VARMEANS, EFFECTS, SEDEFFECTS, VAREFFECTS, WALD, FSTATISTIC, NDF, DDF.

See also
Directives: REML, VKEEP.
Procedures: VASERIES, VASDISPLAY, VASMEANS.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VASMEANS

Saves experiment × treatment means from analysis of a series of trials by VASERIES (R.W. Payne).

Options

FACTORIAL = *scalar*	Limit on the number of factors in the terms generated from the TERMS parameter; default 3
RESIDUALVARIANCES = *table*	Saves residual variances from the experiments
MODELDEFINITIONS = *pointer*	Definitions of the models used by VASERIES
SAVE = *pointer*	REML save structures from the VASERIES analysis

Parameters

TERMS = *formula*	Terms for which means are to be saved
MEANS = *tables* or *pointers*	Experiment × term tables of means
SEMEANS = *tables* or *pointers*	Experiment × term tables of standard errors of means
AVESEDMEANS = *tables* or *pointers*	Average standard errors of differences for the experiments

Description

VASMEANS saves experiment-by-term tables of means, and associated information, from a series of experiments analysed by the VASERIES procedure. The model definitions and REML save structures from the analysis can be specified by the MODELDEFINITIONS and SAVE options. If either of these is not specified, the output is taken from the most recent VASERIES analysis.

The parameters of VASMEANS save the information about the fixed terms. The TERMS parameter specifies a model formula, which Genstat expands to form the series of terms about which you wish to save information. The FACTORIAL option sets a limit on the number of factors in each term. Any term containing more than that limit is deleted.

The subsequent parameters allow you to specify identifiers of data structures to store the information about each of the terms that you have specified. The MEANS parameter saves tables of predicted means, classified by the experiment factor and the factors of the term. Similarly the SEMEANS parameter saves tables of standard errors. The AVESEDMEANS parameter saves tables of average standard errors for the means, classified by the experiment factor (as required for the VMETA procedure). If you have a single term, you can supply a table for each of these parameters, as appropriate. However, if you have several terms, you must supply a pointer which will then be set up to contain as many tables as there are terms.

The RESIDUALVARIANCES option saves residual variances from the experiments, in a table classified by the experiment factor (also as required for the VMETA procedure).

Options: FACTORIAL, RESIDUALVARIANCES, MODELDEFINITIONS, SAVE.
Parameters: TERMS, MEANS, SEMEANS, AVESEDMEANS.

See also

Directives: REML, VKEEP.
Procedures: VASERIES, VASKEEP, VMETA.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VAYPARALLEL

Does the same REML analysis for several y-variates, and collates the output (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (summary, monitoring); default * i.e. none
MODELDEFINITION = *pointer*	Defines the model for the analysis
FSAVETERMS = *formula*	Fixed terms for which to save information; if this is not set, information is saved for all the fixed terms
RSAVETERMS = *formula*	Random terms for which to save information; if this is not set, no information is saved for the random terms
RECOVER = *string token*	Whether to try to recover with a simpler random model if REML cannot fit the model for a particular y-variate (yes, no); default no
METHOD = *string token*	How to choose the best model during recovery (aic, sic, bic); default sic
SPREADSHEET = *string tokens*	What results to save in spreadsheets (components, fixedtests, means, vcmeans, effects, vceffects, residuals, fittedvalues); default * i.e. none
SHEETLAYOUT = *string token*	How to store the results in spreadsheets (yrows, ycolumns, onesheet); default ycol

Parameters

Y = *pointers*	Y-variates for the analyses
RESIDUALS = *matrices*	Saves the residuals
FITTEDVALUES = *matrices*	Saves the fitted values
COMPONENTS = *matrices*	Saves the variance components
MEANS = *pointers*	Pointer to a matrix for each of the terms in FSAVETERMS, saving the predicted means
VCMEANS = *pointers*	Pointer to matrices saving variances and covariances for the means
EFFECTS = *pointers*	Pointer to matrices saving effects for the terms in FSAVETERMS and RSAVETERMS
VCEFFECTS = *pointers*	Pointer to matrices saving variances and covariances for the effects
WALD = *matrices*	Saves the Wald statistics for the terms in FSAVETERMS
FSTATISTIC = *matrices*	Saves the F statistics for the terms in FSAVETERMS
NDF = *matrices*	Saves the numerator degrees of freedom for the terms in FSAVETERMS
DDF = *matrices*	Saves the denominator degrees of freedom for the terms in FSAVETERMS
PRFIXED = *matrices*	Saves the probabilities for the F statistics if available, or otherwise the Wald statistics, for the terms in FSAVETERMS
EXIT = *pointers*	Pointer to scalars saving the exit codes from the initial REML analyses
OUTFILENAME = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create

Description

The VAYPARALLEL procedure does a "parallel" REML analysis of variance for several y-variates, combining and summarizing the information from all the analyses. The MODELDEFINITION option defines the model to be fitted in the analyses. This must be constructed beforehand, using the VFMODEL and VFSTRUCTURE procedures.

Printed output is controlled by the PRINT option, with settings:

monitoring	to print a running total of the number of analyses that have been analysed, and
summary	to print a summary of the significance levels found for the analyses for each of the SAVETERMS.

The SPREADSHEET option allows you to save various output components in spreadsheets. By default, these are opened within Genstat. However, you can set the OUTFILENAME parameter to save them in a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx). If the name supplied by OUTFILENAME is specified without a suffix, '.gwb' is added (so that a Genstat workbook is saved).

The FSAVETERMS and RSAVETERMS options specify the fixed terms and random terms, respectively, whose information is to be saved. By default, information is saved for all the fixed terms and none of the random terms.

The SHEETLAYOUT option controls how the output is stored in the spreadsheet. By default, the various components are stored in separate pages, with a different column for each y-variate. Setting SHEETLAYOUT=yrows still has the components in separate pages, but the output is transposed so that there is a row for each y-variate. Setting SHEETLAYOUT=onesheet also transposes the output, but the components are now put into a single page.

The Y parameter supplies a pointer, containing the y-variates for the analyses. The RESIDUALS, FITTEDVALUES, MEANS, VCMEANS, EFFECTS, VCEFFECTS, WALD, FSTATISTIC, NDF, DDF and PRFIXED parameters allow you to save output components in Genstat data structures. The identifiers of the data structures are also used to identify the corresponding components, if saved in spreadsheets.

The RESIDUALS and FITTEDVALUES parameters save the residuals and fitted values, respectively. By default, these will each be in a matrix, with a column for each y-variate. The matrix is transposed, so that there is now a row for each y-variate, when SHEETLAYOUT is set to yrows or onesheet.

The MEANS parameter saves tables of predicted means for fixed effects. The information is stored in a pointer with a matrix for each of the terms in the formula supplied by the FSAVETERMS option. By default, the matrices have a column for each y-variate, and the rows are labelled to show how they correspond to the cells of the table. The matrices are transposed, when SHEETLAYOUT is set to yrows or onesheet. Similarly VCMEANS parameter saves the variances and covariances of the means.

The EFFECTS parameter saves effects for fixed and random terms. The information is stored in a pointer with a matrix for each of the terms in the FSAVETERMS and RSAVETERMS formulae. The layout of the matrices is controlled by the SHEETLAYOUT option, in the same way as the matrices of means. The VCEFFECTS parameter saves variances and covariances for the effects.

Parameters WALD, FSTATISTIC, NDF, DDF and PRFIXED store information from the corresponding columns of the tables of tests for fixed effects. By default, these are in matrices with a column for each y-variate, and a row for each fixed term. The matrices are transposed, when SHEETLAYOUT is set to yrows or onesheet.

If there are more random terms than are actually needed to explain the random variation, REML may be unable to achieve a successful fit. (The REML likelihood may be too flat for any clear optimum to be found.) You can guard against this situation by setting option RECOVER=yes. Then, if this happens for a particular y-variate, VAYPARALLEL tries models removing first one random term (and any associated spatial model), then two, and so on until successful. The EXIT

parameter allows you to save a pointer, with scalars containing the exit codes from the original REML analyses, so that you can see which y-variates required recovery.

The METHOD option specifies how to choose the random (and spatial) model if there is more than one possible model with the same number of random terms removed:

aic	uses their Akaike information coefficients,
sic or bic	uses their Schwarz (Bayesian) information coefficients (default).

Options: PRINT, MODELDEFINITION, FSAVETERMS, RSAVETERMS, RECOVER, METHOD, SPREADSHEET, SHEETLAYOUT.

Parameters: Y, RESIDUALS, FITTEDVALUES, MEANS, VCMEANS, EFFECTS, VCEFFECTS, WALD, FSTATISTIC, NDF, DDF, PRFIXED, EXIT, OUTFILENAME.

Method

The analyses are performed by the REML directive. The VARECOVER procedure is used to recover from unsuccessful fits.

Action with RESTRICT

Any restrictions on the y-variates will be removed.

See also

Directive: REML.

Procedures: AYPARALLEL, RYPARALLEL.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VBOOTSTRAP

Performs a parametric bootstrap of the fixed effects in a REML analysis (C.J. Brien & R.W. Payne).

Options

†PRINT = *string tokens*	Controls printed output (observedteststatistics, pvalues, vdiagnostics, nnotconverged, monitoring, all, ownstatistics); default obse, pval
VPRINT = *string tokens*	Controls the output from the REML analysis of each sample (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default * i.e. none
PLOT = *string*	What to plot (histogram); default *
NBOOT = *scalar*	Number of bootstrap samples to take; default 99
NRETRIES = *scalar*	Maximum number of extra samples to take when some REML analyses fail to converge; default NBOOT
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
METHOD = *string token*	Indicates whether to use the standard Fisher-scoring algorithm or the new AI algorithm with sparse matrix methods (Fisher, AI); default AI
MAXCYCLE = *scalar*	Sets a limit on the number of iterations in the REML analyses; default 30
FMETHOD = *string token*	Controls whether and how to calculate F statistics for fixed terms (automatic, none, algebraic, numerical); default none
WMETHOD = *string token*	Controls which Wald statistics are saved (add, drop); default add
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm
†OWNMETHOD = *string token*	Type of test required for own statistics (twosided, greaterthan, lessthan); default twos
†CIPROBABILITY = *scalar*	Probability level for the confidence interval for own statistics; default 0.95

Parameters

SAVE = *REML save structures*	Specifies the (REML) save structure of the original analysis; default * uses the SAVE structure from the most recent REML analysis
UMEANS = *variates*	Specifies the expected values for the units under the null hypothesis of no effects from the FIXEDTERMS
UVCOVARIANCE = *symmetric matrices*	Specifies the variances and covariances of the units under the null hypothesis of no effects from the FIXEDTERMS
FIXEDTERMS = *formula*	Specifies the fixed terms to test; default * tests all the fixed terms in the original analysis
FSTATISTICS = *pointers*	Saves a pointer with a variate for each of the FIXEDTERMS, containing the F statistics from the

	bootstrap samples
PVALUES = *pointers*	Saves a pointer with a scalar for each of the FIXEDTERMS, containing the test probability obtained from the position of its F statistic within those from the bootstrap samples
NNOTCONVERGED = *scalars*	Saves the number of bootstrap samples whose REML analysis failed to converge
†OWNDATA = *pointers*	Data required to calculate own statistics
†OWNOBSERVEDVALUES = *variates*	Saves observed values of the own statistics
†OWNPROBABILITIES = *variates*	Saves bootstrap probabilities for the own statistics
†OWNESTIMATES = *variates*	Saves boostrap estimates for the own statistics
†OWNSES = *variates*	Saves boostrap standard errors for the own statistics
†OWNLOWERCIS = *variates*	Saves boostrap lower values of the confidence intervals for the own statistics
†OWNUPPERCIS = *variates*	Saves boostrap upper values of the confidence intervals for the own statistics
†OWNSTATISTICS = *pointers*	Saves the own statistics obtained from the bootstrap samples, in a pointer with a variate for each statistic

Description

VBOOTSTRAP performs a parametric bootstrap for fixed effects in a REML analysis. The model to be fitted must be defined using the VCOMPONENTS and VSTRUCTURE directives, in the usual way. The SAVE parameter supplies the save structure from the original analysis; if this is not set, the most recent REML analysis is used.

The bootstrap samples are generated from a multivariate Normal distribution with dimension equal to the number of units in the analysis. The UMEANS parameter supplies the expected values for the distribution, Usually, this contains the fitted values under the null model for the terms being tested. If UMEANS is not set, a variate containing the grand mean of the response is used. The UVCOVARIANCE parameter supplies the variances and covariances of the units. If this is not set, the unit-by-unit variance-covariance matrix from the original analysis is used (see the UVCOVARIANCE option of VKEEP). Note: you can use the VUVCOVARIANCE procedure to form the variance-covariance matrix, if you know the variance components for a REML model that contains no covariance models.

By default all the fixed terms in the original analysis are tested simultaneously. However, you can set the FIXEDTERMS parameter to test a smaller model, and you should then also set UMEANS to specify the expected values under the null model.

The NBOOT option specifies the number of bootstrap samples to take (default 99). The NRETRIES option specifies the maximum number of extra samples to take when some REML analyses fail to converge; the default is to use the same number as specified by NBOOT. The SEED option supplies the seed for the random number generator used to make the permutations; default 0 continues from the previous generation or (if none) initializes the seed automatically. The NNOTCONVERGED parameter can save the number of samples whose analyses did not converge, in a scalar.

The bootstrap p-values are calculated by taking the proportion of F statistics in the bootstrap samples that are larger than the observed F statistic of each fixed term. The WMETHOD option controls whether these statistics are obtained from the table where terms are added sequentially (the default), or from the table where suitable terms are dropped from the full fixed model. Note that, if you use the table where terms are dropped, the only terms that can be tested are those that are not marginal to any other term in the fixed model: for example, the main effect A cannot be tested if the model contains an interaction, such as A.B.

The bootstrap F statistics can be saved, in a pointer with a variate for each of the

FIXEDTERMS, using the FSTATISTICS parameter. The p-values can be saved, in a pointer with a scalar for each of the FIXEDTERMS, using the PVALUES parameter. You can obtain a plot of a histogram showing the position of the observed F statistic, compared to those from the bootstrap samples, by setting option PLOT=histogram.

You can define your own statistics to be assessed by the bootstrap. They are calculated by a procedure _VBOOTownstatistics, which is called by VBOOTSTRAP following the REML analysis of each bootstrap sample. Its use is shown in the VBOOTSTRAP example, which can be modified to calculate your own statistics instead. The information required by _VBOOTownstatistics to do the calculations is supplied, in a pointer, by the OWNDATA parameter. The OWNMETHOD option specifies the type of test to be made. The default, twosided tests whether the statistics differ from zero. The greaterthan setting tests whether they are greater than zero, and the lessthan setting tests whether they are less than zero. Bootstrap estimates, standard errors and confidence intervals are also calculated, The CIPROBABILITY option specifies the probability for the confidence intervals (default 0.95). The OWNOBSERVEDVALUES parameter can save a variate containing the values of the own statistics from the original data set. The OWNPROBABILITIES can save a variate containing the probabilities from the tests. The OWNESTIMATES can save a variate containing the bootstrap estimates of the statistics (calculated as the mean of the values obtained from the bootstrap samples) The OWNSES can save a variate containing standard errors of bootstrap estimates. The OWNLOWERCIS and OWNUPPERCIS parameters can save variates containing the lower and upper values, respectively, of the confidence intervals. Finally, the OWNSTATISTICS can save the values of the own statistics obtained from the bootstrap samples, in a pointer with a variate for each statistic.

Printed output is controlled by the PRINT option, with settings:

observedteststatistics	to print the values of the observed Wald or F statistics for the fixed terms in the original REML analysis,
pvalues	to print the bootstrap p-values of the observed Wald or F statistics for the fixed terms,
vdiagnostics	to print the diagnostics from the REML analyses performed on the bootstrap samples,
nnotconverged	to print the number of samples whose analyses did not converge,
monitoring	to print the progress of the bootstrapping,
ownstatistics	to print the estimates, standard errors and confidence intervals for the own statistics, and
all	to print all the information other than the own statistics.

By default, the observed statistics and the p-values are printed.

The VPRINT option controls the output from the REML analyses of the bootstrap samples, with the same settings as the PRINT option of REML. By default, nothing is printed.

The MAXCYCLE option sets a limit on the number of iterations in the REML analyses (default 30). The METHOD option controls whether REML uses the standard Fisher-scoring algorithm, or the new AI algorithm with sparse matrix methods (the default). The FMETHOD option controls whether and how to calculate F statistics for fixed terms; the default is not to calculate the statistics. (This is relevant if tests for fixed effects are being printed in the REML analyses of the bootstrap samples.) The WORKSPACE option specifies the number of blocks of internal memory to be set up for use by the REML algorithm; the default is to use the same value as in the original REML analysis.

Options: PRINT, VPRINT, PLOT, NBOOT, NRETRIES, SEED, METHOD, MAXCYCLE, FMETHOD, WMETHOD, WORKSPACE, OWNMETHOD, CIPROBABILITY.

Parameters: SAVE, UMEANS, UVCOVARIANCE, FIXEDTERMS, FSTATISTICS, PVALUES, NNOTCONVERGED, OWNDATA, OWNOBSERVEDVALUES, OWNPROBABILITIES, OWNESTIMATES, OWNSES, OWNLOWERCIS, OWNUPPERCIS, OWNSTATISTICS.

See also
Directive: REML.
Directives: REML, VCOMPONENTS.
Procedures: BOOTSTRAP, VCRITICAL, VFLC, VPERMTEST, VRPERMTEST, VUVCOVARIANCE.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VCHECK

Checks standardized residuals from a REML analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (largeresiduals, similarunits, stability); default larg
RMETHOD = *string token*	Which random terms to use when calculating the standardized residuals (final, all); default fina
RLIMIT = *scalar*	Limit for detection of large standardized residuals; if this is not set, the limit is set automatically according to the number of residual degrees of freedom
COMMONFACTORS = *factors*	Factors to define similar units; if this is not set, the factors in the fixed model are used
REPORTFACTORS = *factors*	Additional factors to include in the table of similar units
PROBABILITY = *scalar*	Critical value for the test probabilities to decide whether to generate warning messages from the Levine test for variance stability; default=0.025
NLARGERESIDUALS = *scalar*	Saves the number of large standardized residuals that have been detected
LARGERESIDUALUNITS = *variate*	Saves the unit numbers of the large standardized residuals
SIMILARINFORMATION = *pointer*	Saves details of large standardized residuals and residuals in similar units
STABILITYTEST = *pointer*	Saves the results of the Levene test for stability of the variance of the standardized residuals
SAVE = *REML save structure*	Specifies the analysis to be checked; by default this will be the most recent REML

No parameters

Description

Procedure VCHECK checks standardized residuals from a REML analysis. By default, these are taken from the recent REML analysis. However, you can check an earlier analysis, by using the SAVE option of VCHECK to specify its save structure (saved using the SAVE parameter of the earlier REML command).

The RMETHOD option controls which random terms are used to calculate the standardized residuals, with settings:

all	uses all of the random effects, and
final	uses only the final random term (default).

Output is controlled by the PRINT option, with the following settings.

largeresiduals	reports any large standardized residuals, with their unit numbers.
similarunits	reports large standardized residuals, together with the residuals from similar units.
stability	performs two Levene tests to check whether the residual variance differs according to the size of the response. The data are divided into three groups (small, intermediate and large) according to the sizes of their fitted values. The tests compare the variance of the standardized residuals in the first (small) group with those in the third (large) group, and the variance of the second (intermediate) group with the

variance of other two groups combined..

By default PRINT=largeresiduals.

The RLIMIT option specifies the limit that must be exceeded by the absolute value of a standardized residual for it to be identified as large. If this is not set, the default is taken as 2.0 if the number of degrees of freedom *d* of the random terms in the REML analysis is less than 20, and 4.0 if *d* is greater than 15773. For other values of *d*, the default is the critical value of the Normal distribution for a two-sided test with significance probability 1/*d*. These calculations are the same as those used in regression and analysis of variance, and are intended to ensure that a report should appear for any extreme outlier, but that reports should not appear too often just as a result of random variation.

The NLARGERESIDUALS option saves the number of large standardized residuals that have been found, and the LARGERESIDUALUNITS option can save a variate containing their unit numbers.

The COMMONFACTORS option lists the factors whose levels should be shared by the units that are listed in the report as similar to those with the large residuals. If this is not set, the default is to take the factors in the fixed model. The REPORTFACTORS option lists any other factors that are to be included in the report. The SIMILARINFORMATION option can save a pointer containing details of the table that has been printed. The first element of the pointer, labelled 'Column labels', contains labels to use as column headings for the other elements, The second element, labelled 'Unit number', contains unit numbers. The third element, labelled 'Unit type', is a factor indicating whether each unit contains a large standardized residual, or the standardized residual from a similar unit. The remaining columns contain the values of the factors displayed in the report.

The results of the Levene test for stability of the variance of the standardized residuals can be saved, in a pointer, by the STABILITYTEST option.

If nothing is to be saved and no printed output is requested, VCHECK provides a safety check. It prints a warning message if any large standardized residuals are detected, or if either of the Levene tests generates a test probability less than or equal to the value specified by the PROBABILITY option. The default value is 0.025 (i.e. 2.5%), which is the same as the value used for the similar messages that may occur with the summary of analysis in regression of from procedure ACHECK following an analysis of variance. It is important to realise that the estimated residuals will be correlated. The Levene tests assume that the residuals are independent Normally-distributed observations. Their test probabilities may therefore be too low – and generate too many significant results. So the use of a smaller critical probability value provides some protection against spurious messages.

Options: PRINT, RMETHOD, RLIMIT, COMMONFACTORS, REPORTFACTORS, PROBABILITY, NLARGERESIDUALS, LARGERESIDUALUNITS, SIMILARINFORMATION, STABILITYTEST, SAVE.

Parameters: none.

Method

The standardized residuals are obtained by using VFRESIDUALS to save the residuals with their standard errors. Details about Levene tests can be found in Snedecor & Cochran (1989); also see O'Neill & Mathews (2002) for information about the issues that arise in their use in balanced analysis of variance.

Action with RESTRICT

If the y-variate in the REML was restricted, only the units not excluded by the restriction will be included in the checks.

References

O'Neill, M.E. & Mathews, K.L. (2002) Levene tests of homogeneity of variance for general block and treatment designs. *Biometrics*, **58**, 216-224.

Snedecor, G.W. & Cochran, W.G. (1989). *Statistical Methods (eighth edition)*. Iowa State University Press, Ames.

See also

Directive: `REML`.

Procedures: `ACHECK`, `VFRESIDUALS`, `VRCHECK`, `VPLOT`, `VSOM`.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VCRITICAL

Uses a parametric bootstrap to estimate critical values for a fixed term in a REML analysis (R.W. Payne & C.J. Brien).

Options

PRINT = *string tokens*	Prints the critical values (critical, fcritical, tcritical, wcritical, monitoring); default crit, fcri, tcri, wcri
VPRINT = *string tokens*	Controls the output from the REML analyses (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default * i.e. none
TERM = *formula*	Fixed term to be tested
UMEANS = *variate*	Specifies the expected values for the units under the null hypothesis of no effects from the TERM; default is to use the constant from the SAVE structure
UVCOVARIANCE = *symmetric matrix*	Specifies the variances and covariances of the units under the null hypothesis of no effects from the TERM; default is to take this from the SAVE structure
WCRITICAL = *variate*	Saves the critical values of the Wald statistic
FCRITICAL = *variate*	Saves the critical values of the F statistic
NBOOT = *scalar*	Number of bootstrap samples to take; default 99
NRETRIES = *scalar*	Maximum number of extra samples to take when some REML analyses fail to converge; default NBOOT
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
PROBABILITIES = *scalar* or *variate*	Significance levels for which critical values are required; default 0.05
METHOD = *string token*	Indicates whether to use the Fisher-scoring algorithm or the AI algorithm with sparse matrix methods (Fisher, AI); default AI
MAXCYCLE = *scalar*	Sets a limit on the number of iterations in the REML analyses; default 30
FMETHOD = *string token*	Controls how to calculate estimated denominator degrees of freedom when these are to be saved (automatic, none, algebraic, numerical); default auto
WMETHOD = *string token*	Controls which Wald statistics are saved (add, drop); default add
TMETHOD = *string token*	Type of test to be made for the contrasts (twosided, greaterthan, lessthan, equivalence, noninferiority); default twos
WALD = *variate*	Saves the Wald statistics from the samples
FSTATISTIC = *variate*	Saves the F statistics from the samples
NDF = *scalar*	Saves the numerator degrees of freedom for the Wald and F statistics
DDF = *variate*	Saves the estimated denominator degrees of freedom for

	the F statistics
NNOTCONVERGED = *scalar*	Saves the number of bootstrap samples whose REML analysis failed to converge
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm
SAVE = *vsave*	REML save structure to provide the information about the analysis

Parameters

XCONTRASTS = *variates* or *tables*	X-variate defining a contrast to be detected
CONTRASTTYPE = *string tokens*	Type of contrast (regression, comparison) default rege
ESTIMATE = *variates*	Saves the estimated values of the contrasts from the samples
SE = *variates*	Saves the standard errors for the estimates of the contrasts from the samples
CRITICAL = *variates*	Saves the critical values for the contrasts
TCRITICAL = *variates*	Saves the critical values for the t-statistics of the contrasts

Description

The conventional way to assess fixed terms in a REML analysis is to use either the Wald or the F tests, in the table of tests for fixed effects that is produced by setting option PRINT=wald in either REML or VDISPLAY. The Wald tests have the disadvantage of being biased, i.e. they tend to generate significant results too frequently. The F tests are more reliable. However, their denominator degrees of freedom need to be estimated, using the method of Kenward & Roger (1997), and this may not be feasible for some data sets. These denominator degrees of freedom can also be used in t-tests to assess contrasts amongst the effects of a term; see procedure VTCOMPARISONS. However, those tests must be used with caution, as the degrees of freedom are relevant for assessing the fixed term as a whole, and may differ over the various contrasts.

VCRITICAL provides an alternative method of assessment, that may be useful if the decision from the conventional tests is not clear-cut, or if contrasts are to be assessed. It uses a parametric bootstrap, in the same way as the VBOOTSTRAP procedure. However, it differs from VBOOTSTRAP, in that it generates critical values, rather than assessing the significance of terms in a specific data set. These critical values can be used test hypotheses with a specific data set, and the critical values for the F, Wald and t-statistics may be useful with similar data sets. The critical values for the t-statistics also allow you to determine the size of the contrast that may be detectable in these investigations.

The model to be fitted must be defined using the VCOMPONENTS and VSTRUCTURE directives, in the usual way. The bootstrap samples are generated from a multivariate Normal distribution with dimension equal to the number of units in the analysis. The UMEANS option supplies the expected values for the distribution. This should contain the fitted values under the null model for the term being tested. The UVCOVARIANCE option supplies the variances and covariances of the units. If either UMEANS or UVCOVARIANCE is not specified, defaults are taken from the REML analysis supplied by the SAVE option, or from the most recent REML if SAVE is not set. For UMEANS the default is a variate containing the constant estimated in that analysis. For UVCOVARIANCE it is the unit-by-unit variance-covariance matrix from the analysis (see the UVCOVARIANCE option of VKEEP). Note: you can use the VUVCOVARIANCE procedure to form the variance-covariance matrix, if you know the variance components for a REML model that contains no covariance models.

The NBOOT option specifies the number of bootstrap samples to take (default 99). The NRETRIES option specifies the maximum number of extra samples to take when some REML

analyses fail to converge; the default is to use the same number as specified by NBOOT. The SEED option supplies the seed for the random number generator used to form the samples; default 0 continues from the previous generation or (if none) initializes the seed automatically. The NNOTCONVERGED option can save the number of samples whose analyses did not converge, in a scalar.

The fixed term to be assessed is specified by the TERM option. If the term is a main effect (i.e. if TERM contains just one factor) you can use the XCONTRASTS parameter to specify variates or tables containing the coefficients defining the contrasts amongst the effects of the term. The CONTRASTTYPE option indicates whether each of these is a regression contrast (as specified in analysis of variance by the REG function) or a comparison (as specified by the COMPARISON function).

The TMETHOD option specifies the type of test that is to be used to assess the contrasts, with the following settings.

twosided	assumes a two-sided test to assess whether the contrast differs from zero (default).
lessthan	assumes a one-sided test to assess whether the contrast is less than zero.
greaterthan	assumes a one-sided test to assess whether the contrast is greater than zero.
noninferiority	assumes a test to check that the contrast is not significantly less then zero. (See Method for more details.)
equivalence	assumes a one-sided test to check that the contrast does not differ significantly from zero; see Method for more details.

The PROBABILITIES option specifies the significance levels for which you want to obtain critical values; the default is 0.05, i.e. 5%.

Printed output is controlled buy the PRINT option, with the following settings.

critical	prints critical values for the contrasts,
fcritical	prints critical values for the F statistics,
tcritical	prints critical values for the t-statistics of the contrasts,
wcritical	prints critical values for the Wald statistics,
nnotconverged	prints the number of bootstrap samples whose analysis failed to converge, and
monitoring	prints monitoring information, showing the progress of the bootstrap sampling.

By default, all the critical values printed.

The critical values for the contrasts and their t-statistics can be saved, in variates, by the CRITICAL and TCRITICAL parameters, respectively. The critical values for the F and Wald statistics can be saved, again in variates by the FCRITICAL and WCRITICAL options.

You can also save the values estimated for the various statistics, in the analyses of the bootstrap samples, in variates (with a unit for each sample). Those for the contrasts and their standard errors can be saved the ESTIMATES and SE parameters, respectively. The F and Wald statistics can be saved by the FSTATISTIC and WALD options. The degrees of freedom for the Wald statistics and numerator degrees for the F statistics can be saved, in a scalar, using the NDF option. The estimated denominator degrees of freedom for the F tests can be saved, in a variate, using the DDF option.

The VPRINT option controls the output from the REML analyses of the bootstrap samples, with the same settings as the PRINT option of REML. By default, nothing is printed.

The MAXCYCLE option sets a limit on the number of iterations in the REML analyses (default 30). The METHOD option controls whether REML uses the Fisher-scoring algorithm, or the AI algorithm with sparse matrix methods (the default). The WMETHOD option controls whether the Wald and F statistics are obtained from the table where terms are added sequentially (the

default), or from the table where suitable terms are dropped from the full fixed model. Note that, if you use the table where terms are dropped, the TERM must not be not marginal to any other term in the fixed model: for example, the main effect A cannot be tested if the model contains an interaction, such as A.B. The FMETHOD option controls how to estimate the denominator degrees of freedom for the F tests. (This is relevant if tests for fixed effects are being printed in the REML analyses of the bootstrap samples, or if the DDF option is set.) The WORKSPACE option specifies the number of blocks of internal memory to be set up for use by the REML algorithm. The default is to use the same value as in the SAVE structure, if SAVE has been set. Otherwise, it uses the value from the most recent REML analysis, or the standard REML default if there has been no analysis.

Options: PRINT, VPRINT, TERM, UMEANS, UVCOVARIANCE, WCRITICAL, FCRITICAL, NBOOT, NRETRIES, SEED, PROBABILITIES, METHOD, MAXCYCLE, FMETHOD, WMETHOD, TMETHOD, WALD, FSTATISTIC, NDF, DDF, NNOTCONVERGED, WORKSPACE, SAVE.
Parameters: XCONTRASTS, CONTRASTTYPE, ESTIMATE, SE, CRITICAL, TCRITICAL.

Method

The critical values are calculated by taking appropriate quantiles of the statistics obtained from the bootstrap samples. For the Wald and F statistics, and the "greater-than" tests of the contrasts or their t-statistics, this is the quantile for one minus the probability. For the "less-than" tests of the contrasts or their t-statistics, it is the quantile for the probability. For the two-sided tests, the quantiles are taken over the absolute values of the contrasts and their t-statistics, and are for one minus the probability.

With an equivalence test, you define a threshold h below which two treatments can be assumed to be equivalent. The contrast c would be the difference between the treatments, and the null hypothesis that the treatments are not equivalent is that either

$$c \leq -t$$

or

$$c \geq t$$

with the alternative hypothesis that they are equivalent, i.e.

$$-t < c < \mathrm{t}$$

This defines an *intersection-union* test, in which each component of the null hypothesis must be rejected separately. This implies performing two one-sided t-tests (this is known as a *TOST* procedure). If the significance level for the full test is to be α, each t-test must have significance level α (see Berger & Hsu 1996). The critical values are thus given by quantiles that are taken over the absolute values of the contrasts and their t-statistics, and are for one minus twice the probability. The hypothesis that the treatments are equivalent would be rejected if the absolute value of the estimated contrast was less than the critical value.

With a non-inferiority test, you again define the threshold t for the effect of the new treatment to be inferior to the standard treatment, and a contrast representing the effect of the new test minus the effect of the standard treatment. The null hypothesis is

$$-c \geq t$$

which represents a one-sided "less-than" t-test.

Reference

Berger, M.L. & Hsu, J.C. (1996). Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Statistical Science*, **11**, 283-319.

See also
Directive: REML, VCOMPONENTS, VSTRUCTURE.
Procedure: VBOOTSTRAP, VPOWER, VUVCOVARIANCE.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VDEFFECTS

Plots one- or two-way tables of effects estimated in a REML analysis (R.W. Payne).

Options

GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
METHOD = *string token*	What to plot (effects, lines); default effe
XFREPRESENTATION = *string token*	How to label the x-axis (levels, labels); default labels uses the XFACTOR labels, if available
PSE = *string*	What s.e. to plot to represent variation (differences, effects, alleffects); default diff
SAVE = *REML save structure*	Save structure of the analysis to display; the default is to take the most recent REML analysis

Parameters

XFACTOR = *factors*	Factor providing the x-values for each plot
GROUPS = *factors*	Factor identifying the different sets of points from a two-way table of effects
COVARIATES = *variates*	X-variates for regression coefficients or pointer
NEWXLEVELS = *variates*	Values to be used for XFACTOR instead of its existing levels
TITLE = *texts*	Title for the graph; default defines a title automatically
YTITLE = *texts*	Title for the y-axis; default ''
XTITLE = *texts*	Title for the x-axis; default is to use the identifier of the XFACTOR

Description

VDEFFECTS plots tables of effects estimated in a REML analysis. By default the effects are from the most recent analysis, but you use the SAVE option to specify the save structure from some other analysis.

The XFACTOR parameter indicates the factor against whose levels the effects are plotted. You can also specify a second factor, using the GROUPS parameter, to plot a two-way table of effects. A separate set of points is then plotted for every level of GROUPS.

By default, the effects will be for the model term XFACTOR (if GROUPS is not set) or XFACTOR.GROUPS (if GROUPS is set). You can also specify one, or more, variates for the term, using the COVARIATES parameter. If COVARIATES is set to a single variate, xvar say, the term will be XFACTOR.xvar or XFACTOR.GROUPS.xvar (representing regression coefficients for xvar). Alternatively, it can be set to a pointer containing several variates, for example x1var and x2var. The term will be then be XFACTOR.x1var.x2var or XFACTOR.GROUPS.x1var.x2var (representing regression coefficients for the product of the variates x1var and x2var).

The NEWXLEVELS parameter enables different levels to be supplied for XFACTOR if the existing levels are unsuitable. If XFACTOR has labels, these are used to label the x-axis unless you set option XFREPRESENTATION=levels.

Usually, each estimate is represented by a point (using pens 1, 2, and so on for each level in turn of the GROUPS factor). However, with high-resolution plots, the METHOD option can be set to lines to draw lines between the points. The GRAPHICS option controls whether a high-resolution or a line-printer graph is plotted; by default GRAPHICS=high.

The PSE option specifies how to represent the variability of the effects, as follows:

differences	plots an error bar showing the average standard error for

	differences between pairs of effects;
`effects`	plots an error bar showing the average standard error of the effects;
`alleffects`	plots a bar around each estimate showing plus and minus its standard error.

The `TITLE`, `YTITLE` and `XTITLE` parameters allow you to supply titles for the graph, the y-axis and the x-axis respectively.

Options: `GRAPHICS`, `METHOD`, `XFREPRESENTATION`, `PSE`, `SAVE`.
Parameters: `XFACTOR`, `GROUPS`, `COVARIATES`, `NEWXLEVELS`, `TITLE`, `YTITLE`, `XTITLE`.

Method
`VDEFFECTS` uses the `GET` directive, if necessary, to obtain the `REML` save structure, and `VKEEP` to obtain the tables of estimates.

See also
Procedures: `VGRAPH`, `AGRAPH`, `DTABLE`, `RGRAPH`, `RDESTIMATES`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VDFIELDRESIDUALS

Display residuals from a REML analysis in field layout (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (table); default * i.e. none
PLOT = *string tokens*	Controls the graphs that are displayed (contour, shade); default cont
RMETHOD = *string token*	Which random terms to use to calculate the residuals (final, all, notspline, stfinal, stall); default all
GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
MARGIN = *string token*	Whether to include margins in printed tables (yes, no); default no
YORIENTATION = *string token*	Y-axis orientation of the plot (reverse, normal); default norm
PENCONTOUR = *scalar*	Pen number to be used for the contours; default 1
PENFILL = *scalar* or *variate*	Pen number(s) defining how to fill the areas between contours; default 3
PENSHADE = *scalar* or *variate*	Pen(s) to use for the shade plot; default 3

Parameters

Y = *variates* or *factors*	Specifies the y-coordinates of the plots
X = *variates* or *factors*	Specifies the x-coordinates of the plots
SAVE = *REML save structures*	Save structure of the REML analysis from which to take the residuals; default is to take the most recent REML analysis
FIELDWIDTH = *scalars*	Field width for printing the residuals; default 12
DECIMALS = *scalars*	Number of decimal places to use when printing the residuals
TITLE = *texts*	Titles for the plots

Description

VDFIRLDRESIDUALS allows you to display residuals from a REML analysis in a two dimensional layout as, for example, from a field experiment. This can be useful to study the spatial pattern of the residuals, for example to see if there are any systematic trends in fertility.

The locations of the plots are defined by the Y and X parameters, specifying variates or factors containing their y- and x-coordinates respectively. By default the residuals are taken from the most recent REML analysis. However, you can take the residuals from some other analysis, by specifying its save structure using the SAVE parameter.

The RMETHOD option controls which random terms are used to calculate the residuals:

all	all the random effects (default),
final	only the final random term,
notspline	all except any random spline terms,
stall	standardized residuals using all the random effects, and
stfinal	standardized residuals using only the final random term.

Usually, the plots in the experiment will all have different coordinates. However, if there are several plots with the same coordinates, mean residuals are calculated for each location. Thus for example, if you wanted only to look at the block and whole-plot residuals in a split-plot design, you could form the residuals from all the random terms, and then set identical coordinates for the (sub-) plots within each whole plot.

VDFIELDRESIDUALS provides two types of graph, selected by the settings of the PLOT option

as follows:

`contour`	generates a contour plot if the plots are on a regular grid, or a line graph if they are arranged in a single line, and
`shade`	produces a shade plot for plots that are on a regular grid.

By default `PLOT=contour`. You can also set option `PRINT=table` to print the residuals in a table, whose structure corresponds to the field layout,

The `GRAPHICS` option determines the type of graphics that is used, with settings `highresolution` (the default) and `lineprinter`. No graphs can be produced if the plots are in an irregular 2-dimensional arrangement. High-resolution contour plots require more than three rows and columns, and line-printer contour plots require more than four rows and columns.

The way in which the lines are drawn in high-resolution contour plots is defined by the properties of the pen specified by the `PENCONTOUR` option, while the pen specified by the `PENFILL` parameter defines how to shade the areas between the contours. Their defaults are 1 and 3 respectively. Similarly, the pen or pens specified by the `PENSHADE` option control the colouring of the shade plot; the default is to use pen 3. For more information see the `DCONTOUR` and `DSHADE` directives.

The `MARGIN` option, with settings `no` (default) and `yes`, determines whether or not marginal means are included with the printed tables. The `FIELDWIDTH` and `DECIMALS` parameters can be used to specify the formats of the printed tables (as in the `PRINT` directive). The `TITLE` parameter can supply a title. If this is not set, a default title is formed.

The `YORIENTATION` option controls the orientation of the y-coordinates in the plots and tables. By default this is `normal`, so that they run upwards from the bottom of the page (as in a map).

Options: `PRINT`, `PLOT`, `RMETHOD`, `GRAPHICS`, `MARGIN`, `YORIENTATION` `PENCONTOUR`, `PENFILL`, `PENSHADE`.
Parameters: `Y`, `X`, `SAVE`, `FIELDWIDTH`, `DECIMALS`, `TITLE`.

Method

`VDFIELDRESIDUALS` obtains the residuals using the `VKEEP` directives, and standardizes them (if required) using standard errors from procedure `VFRESIDUALS`.

Action with `RESTRICT`

If either of `X` or `Y` is restricted, only the unrestricted field plots are displayed.

See also

Directive: `REML`.
Procedures: `AFIELDRESIDUALS`, `VPLOT`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VEQUATE

Equates values across a set of data structures (P.W. Goedhart).

No options

Parameters

OLDSTRUCTURES = *pointers*	Structures whose values are to be transferred – each pointer should contain a set of structures with the same length and type (either scalar, variate, matrix, diagonal matrix, symmetric matrix, table, text or pointer)
NEWSTRUCTURES = *pointers*	Structures to contain the transferred values – each pointer contains a set of either variates, texts or pointers, as relevant to the type of the OLDSTRUCTURES

Description

VEQUATE allows the values in a set of structures to be copied into another set of structures, one for each element of the original structures. The original structures are input in a pointer, using the OLDSTRUCTURES parameter. They must all be of the same type (scalar, variate, matrix, diagonal matrix, symmetric matrix, table, text or pointer), and have the same number of values.

The structures to take the values are returned in a pointer, whose identifier is specified by the NEWSTRUCTURES parameter. The values in the first element of each of the original structures are copied into the first structure in the NEWSTRUCTURES pointer, then those in the second element are copied into the second structure, and so on. If the old structures contain numbers, the new structures will be variates. If they are texts, the new structures will be texts. Finally, if they are pointers, the new structures will be pointers. If NEWSTRUCTURES has already been declared, it should be to a pointer of the correct length. The structures to which it points will be redefined, if necessary, to have the correct length.

Options: none.
Parameters: OLDSTRUCTURES, NEWSTRUCTURES.

Method

EQUATE is used to transfer values. If OLDSTRUCTURES points to restricted variates or texts, the values included in the subset are first copied to dummy structures.

Action with RESTRICT

If the OLDSTRUCTURES pointer consists of variates or texts, any restrictions will be taken into account and, if the NEWSTRUCTURES pointer is not declared in advance, its suffixes will be set to the units in the restricted set.

See also

Directive: EQUATE.
Procedures: APPEND, STACK.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

VFIXEDTESTS

Saves fixed tests from a REML analysis (R.W. Payne).

Options

FIXEDTESTS = *pointer*	Saves the fixed tests
FMETHOD = *string token*	Controls whether and how to calculate F-statistics (automatic, none, algebraic, numerical); default auto
WMETHOD = *string token*	Controls which tests are saved (add, drop); default drop
SAVE = *REML save structure*	Specifies the save structure from the required analysis; default * i.e. most recent one

No parameters

Description

VFIXEDTESTS saves the results of the fixed tests in a REML analysis. By default the results are from the most recent REML, but you use the SAVE option to specify the save structure from some other analysis.

The WMETHOD option controls whether the tests are from the table where terms are added sequentially to the model, or that where terms are dropped from the full fixed model.

The FMETHOD option specifies which algorithm to use to calculate the denominator numbers of degrees of freedom required for F tests. The default, automatic, will use any stored values that have been calculated for this analysis by earlier REML, VDISPLAY or VKEEP statements; otherwise it will choose automatically between the two available methods. (See REML for more details.)

The tests are saved, in a pointer, using the FIXEDTESTS option. The pointer is labelled by the headings from the tests for fixed tests that appear in the REML output. If the denominator degrees of freedom are available, the labels and their corresponding vectors are as follows:

Term	text containing the names of the fixed terms,
Wald statistic	variate containing the Wald statistics,
n.d.f.	variate containing the numerator degrees of freedom,
F statistic	variate containing the F statistics,
d.d.f.	variate containing the denominator degrees of freedom,
F pr.	variate containing the probabilities for the F tests.

If the denominator degrees of freedom are not available (either because they could not be calculated, or because FMETHOD has been set to none), the labels F statistic, d.d.f. and F pr. are omitted, and instead there is

Chi pr.	variate containing the probabilities for chi-square tests for the Wald statistics.

The vectors have an element for each fixed term, with missing values if its test results are unavailable. (For example, with the fixed model A*B, tests for the main effects A and B would be available only when WMETHOD=add.)

Options: FIXEDTESTS, FMETHOD, WMETHOD, SAVE.
Parameters: none.

See also
Directive: VKEEP.
Procedure: VSPREADSHEET.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VFLC

Performs an F-test of random effects in a linear mixed model based on linear combinations of the responses, i.e. an FLC test (V.M. Cave).

PRINT = *string tokens*	Controls printed output (summary, monitoring); default summ
PLOT = *string tokens*	What graphs to plot for the bootstrap and fast double bootstrap FLC tests (kerneldensity, histogram); default * i.e. none
TEST = *string tokens*	Type(s) of test to perform; (flc, bootstrap, fastdoublebootstrap); default flc
NBOOT = *scalar*	Number of bootstrap samples to take; default 99
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
WINDOW = *scalar*	Window to use for the graphs; default 3
SAVE = *REML save structure*	Specifies the save structure of the original analysis; default is to use the save structure from the most recent REML analysis

Parameters

TERMS = *formula*	Random terms to test
STATISTIC = *scalar*	Saves the FLC test statistic
BOOTSTATISTICS = *variate*	Saves the FLC test statistics from the original data set (i.e. the observed FLC test statistic), and then the bootstrap samples
FASTDOUBLE = *pointer*	Pointer to scalars and variates to save the first-level bootstrap probability value and FLC test statistics, and the second-level fast double bootstrap FLC test statistics and resulting critical value
PROBABILITIES = *pointer*	Pointer to scalar(s) to save the probability value(s) from the test(s)
TITLE = *text*	Title for the graphs

Description

The VFLC procedure performs an FLC test to assess whether random terms can be dropped from a linear mixed model, that has been fitted by REML. The FLC test is an F-test based on linear combinations of the responses. VFLC offers the standard FLC test as well its bootstrapped and fast double bootstrapped counterparts.

The original linear mixed model must be fitted using the REML, VCOMPONENTS and VSTRUCTURE directives, in the usual way. The random effects may be correlated, but the model must not contain any spline terms. The SAVE option supplies the save structure from the original analysis; if this is not set, the most recent REML analysis is used. The random term(s) to drop from the original model are defined by a model formula supplied by the TERMS parameter.

The types of FLC test to be performed are specified by the TEST option, with settings flc, bootstrap and fastdoublebootstrap. The default is to use the standard FLC test. For the bootstrap and fast double bootstrap FLC tests, the NBOOT option specifies the number of bootstrap samples to take (default 99), and the SEED option supplies the seed for the random number generator used to generate the bootstrap samples. The default SEED of zero continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, it initializes the seed automatically. If you use the same (non-zero) seed more than once, you will get the same random numbers, and hence the same

bootstrap samples.

Printed output is controlled by the PRINT option, with the following settings.

summary	prints a summary of the test results. For the standard FLC test, this is a table giving the test statistic (i.e. an F-value), its degrees of freedom and corresponding probability value. For the bootstrap and fast double bootstrap FLC tests, this is a table giving the number of bootstrap samples, the seed, the test statistic (i.e. the observed F-value) and the corresponding probability value.
monitoring	prints monitoring information, showing the progress of the bootstrapping.

The default is to print the summary.

The PLOT option controls the graphical output from the bootstrap and fast double bootstrap FLC tests, with settings:

histogram	to plot a histogram of the bootstrap FLC test statistics, and
kerneldensity	to produce a kernel density plot of the bootstrap FLC test statistics.

By default, nothing is plotted. If TEST=bootstrap, the observed FLC test statistic is included in the set of bootstrap FLC test statistics that are plotted. In addition, a reference line is added to indicate where it sits compared to those from the bootstrap samples. Conversely, if TEST=fastdoublebootstrap, the observed FLC test statistic is not included in the set of bootstrap FLC test statistics plotted, and the reference line indicates where the estimated fast double bootstrap critical value, QB, falls. The WINDOW option defines the window to use for the plots; default 3. The TITLE parameter can supply a title for the plots.

Results can be saved using the STATISTIC, BOOTSTATISTICS, FASTDOUBLE and PROBABILITIES parameters. The STATISTIC parameter saves the FLC test statistic in a scalar. The BOOTSTATISTICS parameter saves the bootstrap FLC statistics in a variate, whose first value is the test statistic from the original data set (i.e. the observed FLC test statistic). The FASTDOUBLE parameter saves the results from the fast double bootstrap FLC test in a pointer. The first element of the pointer, labelled 'B_FLC pr.', is a scalar storing the first-level bootstrap probability value. The second element, labelled 'B_FLC F', is a variate storing the first-level bootstrap FLC test statistics. The third element, labelled 'FDB_FLC F', is a variate storing the second-level fast double bootstrap FLC test statistics. The fourth element, labelled 'QB', is a scalar the storing the critical value from the fast double bootstrap FLC test.

Options: PRINT, PLOT, TEST, NBOOT, SEED, WINDOW, SAVE.
Parameters: TERMS, STATISTIC, BOOTSTATISTICS, FASTDOUBLE, PROBABILITIES, TITLE.

Method

VFLC uses the methods described in Hui *et al.* (2019) and O'Shaughnessy *et al.* (2018).

Action with RESTRICT

The REML analysis may be restricted in the usual way.

References

Hui, F.K.C., Müller, S., & Welsh, A.H. (2019). Testing random effects in linear mixed models: another look at the F-test. *Australia & New Zealand Journal of Statistics*, **61**, 61-84.

O'Shaughnessy, P.Y., Hui, F.K.C., Müller, S., & Welsh, A.H. (2018). Bootstrapping F-test for random effects in linear mixed models. arXiv:1812.03428.

See also
Directive: REML, VCOMPONENTS, VSTRUCTURE
Procedure: VBOOTSTRAP, VRPERMTEST.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VFMODEL

Forms a model-definition structure for a REML analysis (R.W. Payne).

Options

MODELSTRUCTURE = *pointer*	Specifies the model-definition structure; no default (must be specified)
DESCRIPTION = *text*	Description of the model (for output)
FIXED = *formula*	Fixed model terms; default *
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
CADJUST = *string token*	What adjustment to make to covariates before analysis (mean, none); default mean
CHANGEITEMS = *string tokens*	What changes to make to an existing model-definition structure (description, fixed, constant, factorial, cadjust, random, initial, constraints); if this is unset, the structure is redefined completely
IMODELSTRUCTURE = *pointer*	Specifies the initial model-definition structure, to modify when CHANGEITEMS is set; default is to modify the one specified by MODELSTRUCTURE
EXPERIMENTS = *factor*	Factor defining the different experiments in a multi-experiment (meta-) analysis

Parameters

RANDOM = *formula*	Random model terms
INITIAL = *scalars*	Initial values for each component
CONSTRAINTS = *string tokens*	How to constrain each variance component and the residual variance (none, positive, fixrelative, fixabsolute); must be set unless MODIFY=yes

Description

VFMODEL is one of a suite of procedures designed to simplify the assessment of alternative models for a REML analysis. The first step is to form a model-definition structure for each candidate model, using the VFMODEL and VFSTRUCTURE procedures (these define the model settings controlled by the VCOMPONENTS and VSTRUCTURE directives, respectively). The model-definition structures can then be used as input to procedures like VARANDOM, which assesses possible random models. VARANDOM uses VMODEL to specify each model, in turn, so that it can fit it using REML. The relevant results from each fit are saved by the VRACCUMULATE procedure, so that a decision about the recommended random model can be made once they have all been tried.

The model-definition structure is specified by the MODELDEFINITION option, which must be set. The DESCRIPTION option supplies a (brief, one-line) description to identify the model in the output.

Details of the model are specified by the FIXED, CONSTANT, FACTORIAL, CADJUST and EXPERIMENTS options, and the RANDOM, INITIAL, CONSTRAINTS parameters (which correspond to those options and parameters of the VCOMPONENTS directive).

You can set the CHANGEITEMS option to modify an existing model-definition structure, instead of redefining it. Its settings then specify which aspects are to be changed. By default, the existing definition structure is supplied by MODELSTRUCTURE (and the modified structure

replaces the existing one). Alternatively, if you want to keep the existing structure, you can specify it separately, using the IMODELSTRUCTURE option.

Options: MODELSTRUCTURE, DESCRIPTION, FIXED, CONSTANT, FACTORIAL, CADJUST, CHANGEITEMS, IMODELSTRUCTURE, EXPERIMENTS.
Parameters: RANDOM, INITIAL, CONSTRAINTS.

See also
Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VARANDOM, VFSTRUCTURE, VMODEL.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VFPEDIGREE

Prepares pedigree information to generate an inverse relationship matrix for use when fitting animal or plant breeding models by REML (S.A. Gezan & R.W. Payne).

Options

FREPRESENTATION = *string token*	Whether to match factor values by their levels or their labels (levels, labels); default leve
†SEX = *string token*	Possible sex categories of parents (fixed, either); default fixe
UNKNOWN = *scalar* or *string*	Value to be treated as unknown in the pedigree factors
†INVMETHOD = *string token*	How to represent the INVERSE (full, sparse); default spar

Parameters

INDIVIDUALS = *factors*	Individuals on which data have been measured
MALEPARENTS = *factors*	Male parents (or sires) of the progeny
FEMALEPARENTS = *factors*	Female parents (of dams) of the progeny
NEWINDIVIDUALS = *factors*	New individuals factor, with levels standardized for use in VPEDIGREE
NEWMALEPARENTS = *factors*	New males factor, with levels standardized to match those in the NEWINDIVIDUALS factor
NEWFEMALEPARENTS = *factors*	New females factor, with levels standardized to match those in the NEWINDIVIDUALS factor
OTHERFACTORS = *pointers*	Pointer containing additional factors, that may be used in the REML models, whose levels must also be standardized to match those in the NEWINDIVIDUALS factor
NEWOTHERFACTORS = *pointers*	Pointer containing new additional factors, with standardized levels
†INVERSE = *pointer*	Inverse relationship matrix in sparse matrix form
†POPULATION = *variates*	Full list of identifiers generated from the individuals and parents

Description

In the analysis of animal and plant breeding experiments it may be interesting to take account of the parentage of the animals or genotypes. This *pedigree* information is specified by three factors, one that identifies the individuals for which data are available, and two others that indicate their male parents and their female parents (if available). This information can be used to generate a sparse inverse relationship matrix that can be used by VSTRUCTURE to define a correlation model of the individual (or animal) effects for use in a REML analysis.

The matrix is formed using the VPEDIGREE directive. First, however, VFPEDIGREE needs to standardize the factors so that the levels and labels of the individual, male and female factors match, and that the levels are in ascending order with the parents defined in the individuals factor before their offspring. Otherwise VPEDIGREE will fail. If you are confident that your factors are already standardized, you can call VPEDIGREE direct (and use VFPEDIGREE instead if that fails).

The factors defining the individuals, the male parents (or sires) and, optionally, the female parents (or dams) in the pedigree data set are specified by the INDIVIDUALS, MALEPARENTS and FEMALEPARENTS parameters, respectively. The OTHERFACTORS parameter can specify a pointer containing additional factors, involving the individuals in the pedigree, that may also be needed in the REML models. You can use the NEWINDIVIDUALS, NEWMALEPARENTS, NEWFEMALEPARENTS and NEWOTHERFACTORS parameters to save the new standardized factors.

Otherwise, the original factors are redefined.

The FREPRESENTATION option indicates whether the factor values are to be matched by their levels (the default) or their labels. If the INDIVIDUALS, MALEPARENTS and FEMALEPARENTS factors are being matched by levels, and the number corresponding to each level needs to be redefined, the factors will be given labels to help identify the original values. If INDIVIDUALS has labels, these will be used. Otherwise the labels will be textual forms of the original levels.

The POPULATION option can save the levels of the standardized factors when FREPRESENTATION=levels, or their labels when FREPRESENTATION=labels.

By default, it is assumed that an individual can act as either a male or female parent but not both. Option SEX=either can be used to specify that individuals can act as both male and female parents. This may be useful, for example, in plant breeding analyses.

Missing values in any of the factors will be treated as coding for unknown individuals. Option UNKNOWN allows you to specify an additional code to represent unknown individuals. This should be a scalar (e.g. 0 or –1) when FREPRESENTATION=levels, or a single-valued text (e.g. '*' or '0') when FREPRESENTATION=labels.

The inverse relationship matrix can be saved by the INVERSE parameter. By default, this is held in a special sparse matrix form (that is, only non-zero values are stored), using a pointer. This is usable in the VSTRUCTURE directive but not elsewhere in Genstat. The second element of the pointer is a variate storing the non-zero values of the inverse matrix in lower-triangular order. The first element of the pointer is an integer index vector. Alternatively, you can set option INVMETHOD=full to store the full matrix as a symmetric matrix (which can also be used by VSTRUCTURE). However, this is not recommended for large pedigrees.

Options: FREPRESENTATION, SEX, UNKNOWN, INVMETHOD.

Parameters: INDIVIDUALS, MALEPARENTS, FEMALEPARENTS, NEWINDIVIDUALS, NEWMALEPARENTS, NEWFEMALEPARENTS, OTHERFACTORS, NEWOTHERFACTORS, INVERSE, POPULATION.

Action with RESTRICT

VFPEDIGREE ignores any restrictions on the factors.

See also

Directives: REML, VCOMPONENTS, VPEDIGREE, VSTRUCTURE, VRESIDUAL, VSTATUS.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VFRESIDUALS

Obtains residuals, fitted values and their standard errors from a REML analysis (S.J. Welham).

Options

RESIDUALS = *variate*	Saves the residuals
SERESIDUALS = *variate*	Saves standard errors of the residuals
FITTEDVALUES = *variate*	Saves the fitted values
SEFITTEDVALUES = *variate*	Saves prediction standard errors for the fitted values
RMETHOD = *string token*	Which random terms to use when calculating the residuals (final, all); default fina
MAXNUNITS = *scalar*	Maximum number of units for which the full variance-covariance matrix will be formed; default 1000
EXIT = *scalar*	Exit code set to zero if the saving was successful, one otherwise
SAVE = *REML save structure*	Save structure for the required analysis; default uses the save structure from the most recent REML

No parameters

Description

The VFRESIDUALS procedure saves residuals, fitted values and their standard errors from a REML analysis. The residuals are formed as differences between the data and the fitted model. The RMETHOD option controls which random terms are used to calculate the residuals, with settings:

all	uses all of the random effects, and
final	uses only the final random term (default).

The final setting thus provides conditional residuals, with the fitted model is calculated from all of the fixed and random terms in the model. The all setting provides marginal residuals, with the fitted model is calculated from the fixed terms alone.

The residuals and fitted values can be saved, in variates, using the RESIDUALS and FITTEDVALUES options, respectively. The SERESIDUALS option saves the standard errors of the residuals, and the SEFITTEDVALUES option saves the prediction standard errors of the fitted values (i.e. the square root of the prediction error variances).

The standard errors can be calculated in several different ways, and VFRESIDUALS will attempt to use the most efficient method. One method involves saving the full variance-covariance matrix for the data. This can be time-consuming for large data sets, so the MAXNUNITS option sets a limit (default 1000) on the size of data set for which this may be used.

By default, VFRESIDUALS forms the residuals etc. from the most recent REML analysis. However, you can form them from an earlier analysis, by using the SAVE option to specify its save structure (saved using the SAVE parameter of the REML command that performed the analysis).

VFRESIDUALS is currently unable to form standard errors for models containing spline terms.

Options: RESIDUALS, SERESIDUALS, FITTEDVALUES, SEFITTEDVALUES, RMETHOD, MAXNUNITS, EXIT, SAVE.
Parameters: none.

Method

The linear mixed model is

$$y = X\beta + Zu + \varepsilon$$

where

y is a vector of data,

β is a vector of fixed effects, with design matrix X,
u is a vector of random effects, with design matrix Z,
ε is a vector of random error

The conditional residuals take the form

$$\tilde{\varepsilon}_c = y - X\hat{\beta} - Z\tilde{u}$$

with variance matrix

$$\mathrm{var}(\tilde{\varepsilon}_c) = \sigma^2 (R - W C^{-1} W')$$

where

$$W\alpha = X\beta + Zu$$
$$\sigma^2 C^{-1} = \mathrm{var}(\hat{\alpha} - \alpha)$$

and R is the matrix of variances and covariances fitted to the residual.

The standard errors of the residuals are given by the square root of the diagonal of the variance matrix. The diagonal of $\sigma^2 (W C^{-1} W')$ can be obtained as the standard error of the predicted fitted values, and the matrix R can be derived from the fitted model.

The marginal residuals take the form

$$\tilde{\varepsilon}_m = y - X\hat{\beta}$$

with variance matrix

$$\mathrm{var}(\tilde{\varepsilon}_m) = \sigma^2 (H - X (X' H^{-1} X)^{-1} X')$$

where

$$\sigma^2 H = \mathrm{var}(y)$$

Again, the standard errors are given by the square root of the diagonal of this matrix. The diagonal of $\sigma^2 X (X' H^{-1} X)^{-1} X'$ can be obtained as the standard errors of the predicted fitted values. The matrix H can be derived from the fitted model, or obtained using the `UVCOV` option of `VKEEP`.

See also

Directives: `REML`, `VCOMPONENTS`, `VKEEP`.
Procedure: `VCHECK`, `VPLOT`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VFSTRUCTURE

Adds a covariance-structure definition to a REML model-definition structure (R.W. Payne).

Options

MODELSTRUCTURE = *pointer*	Supplies the model-definition structure; no default (must be specified)
EXPERIMENT = *scalar*	Level of the EXPERIMENTS factor for which a residual is to be defined (using the VRESIDUAL directive)
TERMS = *formula*	Model terms for which the covariance structure is to be defined
FORMATION = *string token*	Whether the structure is formed by direct product construction or by definition of the whole matrix (direct, whole); default dire
COORDINATES = *identifiers*	Coordinates of the data points to be used in calculating distance-based models (list of variates or matrix)

Parameters

MODELTYPE = *string tokens*	Type of covariance model associated with the term(s), or with individual factors in the term(s) if FORMATION=direct (identity, fixed, AR, MA, ARMA, power, banded, correlation, antedependence, unstructured, diagonal, uniform, FA, FAequal) default iden
ORDER = *scalar*	Order of model
HETEROGENEITY = *string token*	Heterogeneity for correlation matrices (none, outside); default none
METRIC = *string token*	How to calculate distances when MODELTYPE=power (cityblock, squared, euclidean); default city
FACTOR = *factors*	Factors over which to form direct products

Description

VFSTRUCTURE is one of a suite of procedures designed to simplify the assessment of alternative models for a REML analysis. The first step is to form a model-definition structure for each candidate model, using the VFMODEL and the VFSTRUCTURE procedures (these define the model settings controlled by the VCOMPONENTS and the VSTRUCTURE and VRESIDUAL directives, respectively). The model-definition structures can then be used as input to procedures like VARANDOM, which assesses possible random models. VARANDOM uses VMODEL to specify each model, in turn, so that it can fit it using REML. The relevant results from each fit are saved by the VRACCUMULATE procedure, so that a decision about the recommended random model can be made once they have all been tried.

The model-definition structure must be specified by the MODELDEFINITION option. Details of the model are specified by the TERMS, FORMATION, COORDINATES and EXPERIMENT options, and the MODELTYPE, ORDER, HETEROGENEITY, METRIC, and FACTOR parameters (which correspond to those options and parameters in the VSTRUCTURE and VRESIDUAL directives). If the EXPERIMENT option is not set, the specification will be used in a VSTRUCTURE statement within VMODEL. The EXPERIMENT option is relevant if you have used the EXPERIMENTS option in the original VFMODEL statement to define the experiments factor for a meta analysis. You can then set EXPERIMENT to a level of that factor to define a residual model for that experiment, using a VRESIDUAL statement within VMODEL.

Options: MODELSTRUCTURE, EXPERIMENT, TERMS, FORMATION, COORDINATES.
Parameters: MODELTYPE, ORDER, HETEROGENEITY, METRIC, FACTOR.

See also
Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VARANDOM, VFMODEL, VMODEL.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VFUNCTION

Calculates functions of variance components from a REML analysis (S.J. Welham).

Options

PRINT = *string token*	Output required (function); default func
RANDOM = *formula*	Random model (excluding residual stratum) used for the REML analysis
NCONSTANT = *scalar*	Value to be used as constant in the numerator function; default 0
DCONSTANT = *scalar*	Value to be used as constant in the denominator function; default 0
SAVE = *REML save structure*	Specifies the (REML) save structure from which the variance components are to be taken; by default they are taken from the save structure of the most recent REML analysis

Parameters

NUMERATOR = *variates*	Each variate contains a list of coefficients, one for each variance component, defining a linear combination of the components to use as the numerator of the function
DENOMINATOR = *variates*	Each variate contains coefficients defining a linear combination of the variance components to use as the denominator of the function
FUNCTIONVALUE = *scalars*	Saves the calculated value of the function
SE = *scalars*	Saves the approximate standard error of the function value

Description

VFUNCTION calculates linear combinations, reciprocals of linear combinations, or ratios of linear combinations of the estimates of variance components from a REML analysis. The approximate standard errors of the functions are also produced.

The estimated variance components are taken from the structure specified by the SAVE option. If this option is not set, the SAVE structure from the most recent REML analysis is used. The RANDOM option must be set to the random formula used by the REML analysis, but excluding the residual term.

The NUMERATOR parameter supplies a variate that defines the coefficient to use as a multiplier for each variance component in the linear combination of components that forms the numerator of the function. The order of the components is as given by the RANDOM option, with the residual term added at the end. If the variate contains fewer values than the number of components, the final coefficients are taken to be zero. However, random components that were constrained to be fixed in the REML analysis are ignored. The DENOMINATOR parameter similarly defines the linear combination of components in the denominator of the function. If only NUMERATOR is set the function will be linear; conversely if only DENOMINATOR is set it will be a reciprocal function, and if both NUMERATOR and DENOMINATOR are set the function will be the ratio of two linear functions. Options NCONSTANT and DCONSTANT allow a constant to be included in the numerator and denominator functions, respectively.

Printed output is controlled by the option PRINT; by default the calculated value of the function and its approximate standard error are printed. Parameters FUNCTIONVALUE and SE allow the function value and standard error to be saved.

Options: PRINT, RANDOM, NCONSTANT, DCONSTANT, SAVE.
Parameters: NUMERATOR, DENOMINATOR, FUNCTIONVALUE, SE.

Method
The components and their variance-covariance matrix are retrieved using VKEEP. The function is calculated as specified and its approximate standard error is calculated using a formula derived from a Taylor expansion (see, for example, Kendall & Stuart 1963, page 232):

$$\text{se}(f/g) = (1/g) \times \sqrt{\{ \text{var}(f) - 2 \times (f/g) \times \text{cov}(f,g) + (f/g) \times (f/g) \times \text{var}(g) \}}$$

Reference
Kendall, M. & Stuart, A. (1963). *The Advanced Theory of Statistics, Volume 1*. Griffin, London.

See also
Directive: VCOMPONENTS.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VGESELECT

Selects the best variance-covariance model for a set of environments (M.P. Boer, M. Malosetti, S.J. Welham & J.T.N.M. Thissen).

Options

PRINT = *string tokens*	What to print (summary, best, model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, waldtests, missingvalues, covariancemodels); default summ, best, comp, cova
VCMODELS = *string tokens*	Specifies the variance-covariance models that are to be compared for the set of environments (identity, diagonal, cs, hcs, outside, fa, fa2, unstructured); default iden, diag, cs, hcs, outs, fa, fa2, unst
CRITERION = *string token*	Defines which criterion is used to compare the different covariance structures (aic, sic); default sic
FIXED = *formula*	Defines extra fixed effects
UNITFACTOR = *factor*	Saves the units factor required to define the random model when UNITERROR is to be used
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default expl, yvar
MAXCYCLE = *scalar*	Limit on the number of iterations; default 100
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm; default 100

Parameters

TRAIT = *variates*	Quantitative trait to be analysed; must be set
GENOTYPES = *factors*	Genotype factor; must be set
ENVIRONMENTS = *factors*	Environment factor; must be set
UNITERROR = *variate*	Uncertainty on trait means (derived from individual unit or plot error) to be included in QTL analysis; default * i.e. omitted
SELECTEDMODEL = *texts*	VCMODELS setting for the best variance-covariance model
SAVE = *REML save structures*	Save the details of each REML analysis for use in subsequent VDISPLAY and VKEEP directives

Description

VGESELECT selects the best covariance structure for genetic correlations between environments. The quantitative trait is specified by the TRAIT parameter, and the environment and genotype factors are specified by the ENVIRONMENTS and GENOTYPES parameters respectively. The UNITERROR parameter allows you to specify uncertainty on the trait means (derived from individual unit or plot error) to include in the random model; by default this is omitted. The UNITFACTOR option allows you to save the factor that is needed to define the unit-error term (you would need this, for example, if you later wanted to save information about the term using VKEEP).

The settings of the VCMODELS option indicate which models to consider for the variance-covariance structure (see the *Method* Section for details). The CRITERION option specifies whether to assess the different covariance structures by using the Bayesian Information Criterion

(BIC), which is also known as the Schwarz Information Criterion (SIC), or by using Akaike's Information Criterion (AIC). The default is to use the Schwarz (Bayesian) criterion. The SELECTEDMODEL parameter can save the setting corresponding to the best covariance structure can be saved.

The PRINT option controls the printed output. The summary setting prints a summary of the analyses, and best prints details of the best model. The other settings correspond to the settings of the PRINT option of the REML directive. The specified output is printed for each model specified by the MODELS option.

The FIXED option can be used to include extra fixed effects, e.g. selected QTLs (genetic predictors). There are also MVINCLUDE, MAXCYCLE and WORKSPACE options which operate in the same way as these options in the REML directive.

Options: PRINT, VCMODELS, CRITERION, FIXED, UNITFACTOR, MVINCLUDE, MAXCYCLE, WORKSPACE.
Parameters: TRAIT, GENOTYPES, ENVIRONMENTS, UNITERROR, SELECTEDMODEL, SAVE.

Method

The method selects the best variance-covariance matrix to model the genetic correlations between environments, based on the Schwarz (Bayesian) Information Criterion (BIC) or Akaike Information Criterion (AIC), as described by Malosetti *et al*. (2004) and Boer *et al*. (2007). The AIC and BIC are defined by:

$$\text{AIC} = \text{deviance} + 2 \times p,$$

$$\text{BIC (or SIC)} = \text{deviance} + \log(N) \times p,$$

where N is the total number of observations, and p is the number of parameters in the variance-covariance matrix. The default is to use the Schwarz (Bayesian) criterion.

The variance-covariance models that can be specified by the VCMODELS option to be compared are as follows:

Setting	Description	Variance-covariance matrix	Number of parameters
identity	Identity	$\mathbf{I}\,\sigma_e^2$	1
cs	Compound symmetry	$\mathbf{J}\,\sigma_g^2 + \mathbf{I}\,\sigma_e^2$	2
diagonal	Diagonal matrix (heteroscedastic)	$\mathbf{D}$	n_{env}
hcs	Heterogeneous compound symmetry	$\mathbf{J}\,\sigma_g^2 + \mathbf{D}$	$n_{env} + 1$
outside	Heterogeneity outside	$\sqrt{\mathbf{D}}\,\mathbf{K}\,\sqrt{\mathbf{D}}$	$n_{env} + 1$
fa	First order factor-analytic model	$\lambda\lambda'\,\mathbf{T} + \mathbf{D}$	$2 \times n_{env}$
fa2	Second order factor-analytic model		$3 \times n_{env}$
unstructured		$\sqrt{\mathbf{D}}\,\mathbf{K}\,\sqrt{\mathbf{D}}$	

In this table n_{env} is the number of environments, σ2e and σ2g are scalars, and λ is a n_{env} dimensional vector. In addition, **I** is the $n_{env} \times n_{env}$ identity matrix, **J** is the $n_{env} \times n_{env}$ matrix with all values equal to one, **K** is the $n_{env} \times n_{env}$ matrix with one in its diagonal elements and θ in its

off-diagonal elements, and **D** is a diagonal matrix containing the variances (σ_{ei}^2:$i = 1...n_{env}$).

The analyses are performed by the REML directive, using the VSTRUCTURE directive to specify the covariance models. The table below summarizes how the models are specified in Genstat notation.

Setting	VSTRUCTURE parameters			
	Model	Heterogeneity	Order	Extra Random term
identity	identity	None		
cs	identity	None		GENOTYPES
diagonal	diagonal	None		
hcs	diagonal	None		GENOTYPES
outside	uniform	Outside		
fa	fa	None	1	
fa2	fa	None	2	
unstructured	unstructured	None		

Action with RESTRICT

Restrictions are not allowed.

References

Boer, M.P., Wright, D,, Feng, L., Podlich, D.W., Luo, L., Cooper, M. & van Eeuwijk F.A. (2007). A mixed-model quantitative trait loci (QTL) analysis for multiple-environment trial data using environmental covariables for QTL-by-environment interactions, with an example in maize. *Genetics*, **177**, 1801-1813.

Malosetti, M., Voltas, J., Romagosa, I., Ullrich, S.E. & van Eeuwijk, F.A. (2004). Mixed models including environmental covariables for studying QTL by environment interaction. *Euphytica*, **137**, 139-145.

See also

Procedures: QMVAF, QMBACKSELECT, QMESTIMATE, QMQTLSCAN.

Genstat Reference Manual 1 Summary sections on: REML analysis of linear mixed models, Statistical genetics and QTL estimation.

VGRAPH

Plots tables of means from REML (R.W. Payne).

Options

GRAPHICS = *string token*	Type of graph (highresolution, lineprinter); default high
METHOD = *string token*	What to plot (points, means, linesandpoints, onlylines, data, barchart, splines); default poin when XFACTOR is a factor, and only when it is a variate
XFREPRESENTATION = *string token*	How to label the x-axis (levels, labels); default labe uses the XFACTOR labels, if available
PSE = *string token*	What to plot to represent variation when points are plotted at the means (differences, lsd, means, allmeans); default diff
LSDLEVEL = *scalar*	Significance level (%) to use for approximate least significant differences; default 5
DFSPLINE = *scalar*	Number of degrees of freedom to use when METHOD=splines
YTRANSFORM = *string tokens*	Transformed scale for additional axis marks and labels to be plotted on the right-hand side of the y-axis (identity, log, log10, logit, probit, cloglog, square, exp, exp10, ilogit, iprobit, icloglog, root); default iden i.e. none
PENYTRANSFORM = *scalar*	Pen to use to plot the transformed axis marks and labels; default * selects a pen, and defines its properties, automatically
†KEYMETHOD = *string token*	What to use for the key descriptions when GROUPS specifies more than one factor (labels, namesandlabels); default name
†PLOTTITLEMETHOD = *string token*	What to use for the titles of the plots when TRELLISGROUPS specifies more than one factor (labels, namesandlabels); default name
†PAGETITLEMETHOD = *string token*	What to use for the titles of the pages when PAGEGROUPS specifies more than one factor (labels, namesandlabels); default name
†USEAXES = *string token*	Which aspects of the current axis definitions of window 1 to use (none, limits, marks, mpositions, nsubticks,); default none
SAVE = *REML save structure*	Save structure to provide the table of means; default uses the save structure from the most recent REML

Parameters

XFACTOR = *factors* or *variates*	Provides the x-values for each plot; by default this is chosen automatically
GROUPS = *factors* or *pointers*	Factor or factors identifying groups in each plot; by default chosen automatically
TRELLISGROUPS = *factors* or *pointers*	Factor or factors specifying the different plots of a trellis plot of a multi-way table
PAGEGROUPS = *factors* or *pointers*	Factor or factors specifying plots to be displayed on

	different pages
NEWXLEVELS = *variates*	Values to be used for XFACTOR; default uses the existing levels if XFACTOR is a factor, and the minimum and maximum values if it is a variate
TITLE = *texts*	Title for the graph; default is to define a title automatically if GROUPS is set, or to have none if it is unset
YTITLE = *texts*	Title for the y-axis; default is to use the identifier of the y-variate, or to have no title if this is unnamed
XTITLE = *texts*	Title for the x-axis; default is to use the identifier of the XFACTOR
PENS = *variates*	Defines the pen to use to plot the points and/or line for each group defined by the GROUPS factors

Description

VGRAPH plots tables of predicted means from REML. In its simplest form, the behaviour of VGRAPH depends on the model. If the fixed model contains only main effects, it plots the means for the first factor in the fixed model. Otherwise, it looks for the first fixed term involving two factors; it then plots the means with one of these factors as the x-axis, and the second as a grouping factor with levels identified by different plotting colours and symbols.

By default, the means are from the most recent REML. However, you can plot means from an earlier analysis, by using the SAVE option of VGRAPH to specify its save structure (saved using the SAVE parameter of the REML command that performed the analysis). VGRAPH uses the VPREDICT directive with default option settings to obtain the means. This should give the same means as those printed by REML or VDISPLAY. If you want to use VPREDICT with other option settings, you can plot these using the DTABLE procedure.

The GRAPHICS option controls whether a high-resolution or a line-printer graph is plotted; by default GRAPHICS=high.

The METHOD option controls how the predicted means are plotted in high-resolution graphics, with settings:

points	to plot a point at each mean;
means	synonym of points;
linesandpoints	to plot points and join them by lines;
onlylines	to draw lines between the means;
data	to draw lines between the means, and then also plot the original data values;
barchart	to plot the means as a barchart;
splines	to plot points at the means together with a smooth spline to show the trend over each group of means; the DFSPLINE specifies the degrees of freedom for the splines; if this is not set, 2 d.f. are used when there are up to 10 points, 3 if there are 11 to 20, and 4 for 21 or more.

The default is to plot points when XFACTOR is a factor, and onlylines when it is a variate. Only points are available in line-printer graphics.

The PSE option specifies the type of error bar to be plotted, when points are plotted for the means, with settings:

differences	average standard error of difference;
lsd	average approximate least significant difference (calculated using the VLSD procedure);
means	average effective standard error for the means;
allmeans	plots plus and minus the effective standard error around

every mean.

The LSDLEVEL option sets the significance level (%) to use for the approximate least significant differences (default 5). The allmeans setting is often unsuitable for plots other than barcharts when there are GROUPS, as the plus/minus e.s.e. bars may overlap each other.

You can define the table of means to plot explicitly, by specifying its classifying factors using the XFACTOR, GROUPS, TRELLISGROUPS and PAGEGROUPS parameters. The XFACTOR parameter can define a factor against whose levels the means are plotted. It can also specify a variate, and VPREDICT then sets up a factor automatically, to classify the table, with levels at the values specified by the NEWXLEVELS parameter. With a multi-way table, there will be a plot of means against the XFACTOR levels for every combination of levels of the factors specified by the GROUPS, TRELLISGROUPS and PAGEGROUPS parameters. The GROUPS parameter specifies factors whose levels are to be included in a single window of the graph. So, for example, if you specify

```
VGRAPH [METHOD=line] XFACTOR=A; GROUPS=B
```

VGRAPH will produce plot the means in a single window with factor A on the x-axis, and a line for each level of the factor B. You can set GROUPS to a pointer to specify several factors to define groups. For example

```
POINTER [VALUES=B,C] Groupfactors
VGRAPH [METHOD=line] XFACTOR=A; GROUPS=Groupfactors
```

to plot a line for every combination of the levels of factors B and C. Similarly, the TRELLISGROUPS option can specify one or more factors to define a trellis plot. For example,

```
VGRAPH [METHOD=line] XFACTOR=A; GROUPS=B; TRELLISGROUPS=C
```

will produce a plot for each level of C, in a trellis arrangement; each plot will again have factor A on the x-axis, and a line for each level of the factor B. Likewise, the PAGEGROUPS parameter can specify factors whose combinations of levels are to be plotted on different pages. So

```
VGRAPH [METHOD=line] XFACTOR=A; GROUPS=B; PAGEGROUPS=C
```

will produce a plot for each level of C, but now on separate pages. Multi-way tables can plotted even if the corresponding model term was not in the REML analysis. For example you can plot a two-way table even if the analysis contained only the main effects of the two factors; however, the lines will then all be parallel and no LSDs can be included.

The NEWXLEVELS parameter enables different levels to be supplied for an XFACTOR factor, if its existing levels are unsuitable. If the factor has labels, these are used to label the x-axis unless you set option XFREPRESENTATION=levels. When XFACTOR is a variate, NEWXLEVELS can specify the values where the predictions are to be made. By default, they are made at its minimum and maximum values.

Note that the values predicted by VPREDICT, for an XFACTOR variate, will not include any spline effects, nor can it take account of any relationships between different variates in the model. (For example, the model may include a variate and its square.) To take account of relationships like these, you should use VPREDICT directly, specifying the linked variables with the PARALLEL parameter. Save the table of predictions, and then plot it using DTABLE.

The TITLE, YTITLE and XTITLE parameters can supply titles for the graph, the y-axis and the x-axis, respectively. The symbols, colours and line styles that are used in a high-resolution plot are usually set up by VGRAPH automatically. If you want to control these yourself, you should use the PEN directive to define a pen with your preferred symbol, colour and line style, for each of the groups defined by combinations of the GROUPS factors. The pen numbers should then be supplied to VGRAPH, in a variate with a value for each group, using the PENS parameter.

The YTRANSFORM option allows you to include additional axis markings, transformed onto another scale, on the right-hand side of the y-axis. Suppose, for example, suppose you have analysed a variate of percentages that have been transformed to logits. You might then set YTRANSFORM=ilogit (the inverse-logit transformation) to include markings in percentages

alongside the logits. The settings are the same as those of the TRANSFORM parameter of AXIS (which is used to add the markings). You can control the colours of the transformed marks and labels, by defining a pen with the required properties, and specifying it with the PENYTRANSFORM option. Otherwise, the default is to plot them in blue.

When there is more than one GROUPS factor, the KEYMETHOD controls whether to use the factor names with their labels (or levels for factors with no labels) or just the labels (or levels) in the key descriptions. The default is to use the names and the labels (or levels). Similarly, the PLOTTITLEMETHOD specifies what to use for the titles of the plots when there is more then one TRELLISGROUPS factor, and the PAGETITLEMETHOD specifies what to use for the titles of the plots when there is more then one PAGEGROUPS factor. You can setKEYMETHOD=* to have no key at all.

The USEAXES option allows you to control various aspects of the axes. First you need to use the XAXIS and YAXIS directives to define them for window 1. Then specify which of the aspects of the axes in window 1 are to be used by DTABLE, by specifying USEAXES with the following settings:

limits	y- and x-axis limits (LOWER and UPPER parameters);
marks	location and labelling of the tick marks (MARKS, LABELS, LDIRECTION, LROTATION, DECIMALS, DREPRESENTATION, and VREPRESENTATION parameters);
mpositions	positions of the tick marks (MPOSITION parameter); and
nsubticks	number of subticks per interval (NSUBTICKS parameter).

By default none are used.

Options: GRAPHICS, METHOD, XFREPRESENTATION, PSE, LSDLEVEL, DFSPLINE, YTRANSFORM, PENYTRANSFORM, KEYMETHOD, PLOTTITLEMETHOD, PAGETITLEMETHOD, USEAXES, SAVE.
Parameters: XFACTOR, GROUPS, TRELLISGROUPS, PAGEGROUPS, NEWXLEVELS, TITLE, YTITLE, XTITLE, PENS.

See also
Procedures: VDEFFECTS, AGRAPH, DTABLE, RGRAPH, RDESTIMATES.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VHERITABILITY

Calculates generalized heritability for a random term in a REML analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (heritability); default heri
SAVE = *REML save structure*	Save structure of the analysis from which to calculate the heritabilities; default uses the most recent REML analysis

Parameters

TERMS = *formula*	Random terms whose heritabilities are to be calculated
HERITABILITY = *scalar* or *variate*	Saves the heritabilities
EXIT = *scalar* or *variate*	Exit status for the calculations: one if unsuccessful, otherwise zero

Description

VHERITABILITY can be used to calculate the generalized heritability for random terms in a REML analysis, using the definition of Cullis, Smith & Coombes (2006). This provides a natural extension of the standard concept of heritability, which was defined in the context of conventional designs like complete randomized block designs, to more complicated analyses like those with spatial correlation models (see VSTRUCTURE).

Heritability measures the proportion of variance that is attributable to the effects of the term. It is often used by plant breeders to assess the proportion of the variance of a phenotypic trait that is attributable to the effects of genotypes, thus providing an indication of potential benefits of selection. Technically, VHERITABILITY provides a broad-sense measure of heritability, on a mean-line basis, that comprises the sum of additive, dominance and epistatic effects. For more details see Falconer & Mackay (1996) or Piepho & Möhring (2007).

By default, the heritabilities are usually calculated from the most recent REML analysis. However, you can use the SAVE parameter to specify the save structure from an earlier analysis.

The TERMS parameter supplies a model formula to specify the terms whose heritabilities are to be calculated. These must all be in the random model of the analysis.

The HERITABILITY parameter allows you to save the heritabilities, in a scalar if there is a single term, or in a variate if there are several. Similarly the EXIT parameter can save a scalar or variate indicating whether each heritability was calculated successfully (zero for success and one for failure). Possible reasons for failures may include the fact that the term was not in the random model, or that it has a negative variance component.

By default, the heritabilities are printed, but you can set option PRINT=* to suppress this.

Options: PRINT, SAVE.
Parameters: TERMS, HERITABILITY, EXIT.

Method

Cullis, Smith & Coombes (2006) define the heritability as

$$1 - A_{tt} / (2 \gamma_g^2)$$

where γ_g^2 is the variance component of the term divided by the residual variance, and A_{tt} is the average variance for differences between the effects of the term divided by the residual variance. This can be simplified, by cancelling out the residual variance, to become

$$1 - V_{tt} / (2 \sigma_g^2)$$

where σ_g^2 is the variance component of the term, and V_{tt} is the average variance for differences between the effects of the term. These are obtained from VKEEP and VPREDICT, respectively.

References

Cullis, B.R., Smith, A. & Coombes, N. (2006). On the design of early generation variety trials with correlated data. *JABES*, **11**, 381–393.

Falconer, D. S. & Mackay, T. (1996). *Introduction to Quantitative Genetics, 4th ed.* Longman, Harlow.

Piepho, H-P & Möhring, J. (2007). Computing heritability and selection response from unbalanced plant breeding trials. Genetics, **177**, 1881-1888.

See also

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VHOMOGENEITY

Tests homogeneity of variances and variance-covariance matrices (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (test, variances); default test
GROUPS = *factors*	Define the groups whose variances are to be compared; these need be given only if DATA is set

Parameters

DATA = *variates* or *pointers*	Data variate from which variances are calculated, or pointer to a list of variates from which variance-covariance matrices are calculated
VARIANCES = *any numerical structures* or *pointers*	Supplies the variances (in any numerical structure) or variance-covariance matrices in a pointer to a list of symmetric matrices if the DATA parameter is not set, or saves variances (in a table) and variance-covariance matrices (in a pointer to a list of symmetric matrices) if they have been calculated from DATA and GROUPS
DF = *any numerical structures*	Supplies the degrees of freedom for variances (in any numerical structure) or for variance-covariance matrices (as a pointer to a list of scalars) if the DATA parameter is not set, or saves the degrees of freedom for variances (in a table) or variance-covariance matrices (as a pointer to a list of scalars) if they have been calculated from DATA and GROUPS
SAVE = *pointers*	Saves the results i.e. type of test, chi-square statistic, degrees of freedom and probability

Description

Equality of variances of residuals is an important requirement for the validity of analysis of variance and regression. VHOMOGENEITY allows the homogeneity of variances in different groups to be assessed using Bartlett's test. This test is rather sensitive to departures from Normality (another requirement for the validity of these analyses); so it is recommended that the residuals also be examined, for example using procedures RCHECK or APLOT.

To test homogeneity of variances, VHOMOGENEITY can take as input either the original data values together with factors defining the groups, or variances along with degrees of freedom. For the first method the DATA parameter should be set to a variate containing the data values; the factors are specified by the GROUPS option. For the second method the variances are input (in any numerical structure) using the VARIANCES parameter, and the degrees of freedom (in a structure of the same type as for VARIANCES) using the DF parameter.

With multivariate data, the analogous test for variance-covariance matrices is given by Box (1950). Again two methods of input are available. The original data variates can be supplied, in a pointer, using the DATA parameter and the factors can be listed by the GROUPS option (as for a single variate). Alternatively, the VARIANCES parameter can be set to a pointer containing the variance-covariance matrices to be tested, and the DF parameter to a pointer containing the corresponding degrees of freedom.

If the variances and degrees of freedom are to be calculated by the procedure (from DATA and GROUPS), the VARIANCES and DF parameters can be used to save the calculated values. When testing homogeneity of variances, the variances and degrees of freedom are saved in tables, classified by the GROUPS factors; these tables need not be declared in advance. With variance-

covariance matrices, VARIANCES is a pointer to the list of symmetric matrices that have been formed, and DF a pointer to a list of scalars.

Printed output is controlled by the PRINT option, with settings variances and test to print the variances and the test statistics respectively. By default, PRINT=test.

You can save the results of the test, in a pointer, using the SAVE parameter. The pointer has the following elements:

'test'	type of test (Bartlett or Box),
'chi-square'	chi-square statistic,
'd.f.'	the number of degrees of freedom, and
'probability'	the probability.

Options: PRINT, GROUPS. Parameters: DATA, VARIANCES, DF, SAVE.

Method

If the raw data have been given as input, the procedure uses TABULATE to form tables of variances, and of replications from which the degrees of freedom are calculated. The test statistic is calculated as M/C, where

$M = \sum n_i \times \log(\sum\{ n_i \times s_i \} / \sum\{ n_i \}) - \sum\{ n_i \times \log(s_i) \}$

$C = 1 + (1 / (3 \times (N - 1))) \times (\sum\{ 1/n_i \} - 1 / (\sum n_i))$

N = number of groups

n_i = degrees of freedom of group i

s_i = variance of group i

The number of degrees of freedom associated with the test statistic is the number of groups minus one. See, for example, Snedecor & Cochran (1980, pages 252-253).

The FSSPM directive is used to form variance-covariance matrices. The equivalent test of homogeneity is given by Box (1950).

Action with RESTRICT

If the DATA variates are restricted, only the units not excluded by the restriction will be used to calculate the variances and degrees of freedom.

References

Box, G.E.P. (1950). Problems in the analysis of growth and wear curves. *Biometrics*, **6**, 362-389.

Snedecor, G.W. & Cochran, W.G. (1980). *Statistical Methods (seventh edition)*. Iowa State University Press, Ames, Iowa.

See also

Directive: VSTRUCTURE.

Procedure: AREPMEASURES.

Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

VINTERPOLATE

Performs linear & inverse linear interpolation between variates (R.J. Reader).

Options

METHOD = *string token*	Type of interpolation required (interval, value): for METHOD=value, y-values are interpolated for each point in the NEWINTERVALS variates and stored in the NEWVALUES variates, while for METHOD=interval, x-values are estimated for the y-values in the NEWVALUES variates and stored in the NEWINTERVALS variates; default inte
RANGEMETHOD = *string token*	Whether the smallest value, largest value or the mean of the two is returned if more than one value is valid (first, middle, last); default midd

Parameters

OLDVALUES = *pointers*	Each one contains variates specifying the y-values (data values) with which an interpolation is to be carried out
NEWVALUES = *pointers*	For METHOD=value, each pointer contains variates to store the results of an interpolation; for METHOD=interval, it contains either variates or scalars to specify y-values for which inverse interpolation is to be carried out
OLDINTERVALS = *variates*	Contains the x-values (intervals) corresponding to the variates in the OLDVALUES pointer
NEWINTERVALS = *pointers*	For METHOD=interval, each pointer contains variates to store the results of an inverse interpolation; for METHOD=value, it contains either variates or scalars to specify x-values at which interpolation is to be performed

Description

VINTERPOLATE performs linear interpolation and inverse linear interpolation between variates, as was given by the Genstat 4 functions LINT and INLINT. The y-values (or data values) are given in a set of variates, contained in the pointer specified by the OLDVALUES parameter. The corresponding x-values (intervals) are specified by the OLDINTERVALS parameter, in a variate with one value for each variate of the OLDVALUES pointer. For interpolation (METHOD=value), values are interpolated at the x-values specified by the variates or scalars contained in the NEWINTERVALS pointer, and are stored in variates contained in the NEWVALUES parameter. For inverse interpolation (METHOD=interval), x-values are estimated for the y-values specified by the variates or scalars contained in the NEWVALUES pointer, and are stored in variates contained in the NEWINTERVALS pointer. Where two or more successive OLDINTERVALS or OLDVALUES are the same, there is no unique solution to the interpolation; the RANGEMETHOD option allows the smallest (RANGEMETHOD=first), largest (RANGEMETHOD=last) or the mean of these two (RANGEMETHOD=middle) values or intervals to be selected.

Options: METHOD, RANGEMETHOD.
Parameters: OLDVALUES, NEWVALUES, OLDINTERVALS, NEWINTERVALS.

Method
An estimate of the required value is calculated from each successive pair of points. If this estimate is between the two points from which it was calculated, it is a valid answer i.e. it was produced by interpolation not extrapolation. If it does not satisfy this condition it is set to missing. If the curve is horizontal or vertical at the point of interpolation, more than one point will satisfy the above condition. In this case the value returned will depend on the setting of the option RANGEMETHOD. This will determine whether the smallest value found, the largest value found, or the mean of these two values is returned. The default is to return the mean value.

Action with RESTRICT
If any of the variates in the OLDVALUES pointer is restricted, the other variates in the pointer must either have the same restriction or be unrestricted. Missing values are then returned for the units excluded by the restriction. The OLDINTERVALS variate must not be restricted.

See also
Directive: INTERPOLATE.
Procedure: VREGRESS.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

VLINEBYTESTER

Analyses a line-by-tester trial by REML (R.W. Payne).

Options

PRINT = *string tokens*	Specifies the output to be produced (model, components, effects, means, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels, aic, sic, bic, combinability, tests); default mode, comp, wald, comb, test
PRECOVERY = *string tokens*	Controls what summary output is produced about the models that are tried during recovery (deviance, aic, bic, sic, dffixed, dfrandom, change, exit, best); default devi, aic, sic, dfra, best
LINES = *factor*	Specifies the line (usually female parent); no default (must be specified)
TESTERS = *factor*	Specifies the tester (usually male parent); no default (must be specified)
CONTROLS = *factor*	Distinguishes between control and test (line × tester) genotypes; default is that there are no controls
FIXED = *formula*	Fixed model terms, in addition to the TESTERS main effect and any control comparisons; default * i.e. none
RANDOM = *formula*	Random model terms, in addition to the terms involving LINES, TESTERS and EXPERIMENTS that are included automatically; default * i.e. none
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
EXPERIMENTS = *factor*	Specifies the different experiments for a REML meta analysis; default is that the data are all from a single experiment
PCOMBINABILITYTERMS = *formula*	Terms whose combinability effects are to be printed, selected from LINES, LINES.TESTERS and their interactions with EXPERIMENTS; default is to print all of them
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be printed; default * implies all the fixed terms
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default diff
MVINCLUDE = *string tokens*	Whether to include units with missing values in the explanatory factors and variates and/or the y-variates (explanatory, yvariate); default * i.e. omit units with missing values in either explanatory factors or variates or y-variates
RECOVER = *string token*	Whether to try to recover with a simpler random model if REML cannot fit the model (yes, no); default no
METHOD = *string token*	How to choose the best model during recovery (aic, sic, bic); default sic

Parameters

Y = *variates*	Response variates
COMBINABILITY = *pointers*	Pointer to tables of combinability effects for each y-variate
SECOMBINABILITY = *pointers*	Pointer to tables of standard errors of combinability effects for each y-variate
DEVIANCES = *variates*	Saves deviances for LINES, LINES.TESTERS and their interactions with EXPERIMENTS
EXIT = *scalars*	Exit status for each y-variate (zero to indicate that the analysis was successful)
SAVE = *REML save structures*	Save structure from the analysis of each y-variate

Description

VLINEBYTESTER does a mixed-model analysis of data from a line-by-tester trial, using REML. These are trials in which several "lines" (usually female parents) are all mated with a smaller number of testers (usually male parents). Generally, all combinations of parent will be present in the trial, but incomplete arrangements, such as diallels, are also possible. Control, or check, genotypes may also be present.

The lines must be specified (in a factor) by the LINES option. Similarly, the testers must be specified (again in a factor) by the TESTERS option. If there are any control genotypes, these are specified by using the CONTROLS option to supply a factor with a different level for each control genotype, and a single level for all the line-by-tester genotypes.

You can use the FIXED and RANDOM options to specify fixed and random terms to be fitted in the analysis. The FACTORIAL option sets a limit on the number of factors and variates allowed in each fixed term (default), and the CONSTANT option can be set to omit the constant.

By default, the testers main effect is treated as fixed, so it will be added to any fixed model that you specify. However, you can put TESTERS into the random model (specified by RANDOM), if you would prefer it to be treated as random. Likewise, the main effect of the CONTROLS factor (comparisons amongst control genotypes, and between controls and the mean of the tests) is treated as fixed, but can be put into the random model if you prefer. The model terms LINES and LINES.TESTERS are treated as random, and so will be added automatically to any random model that is specified by RANDOM. If an EXPERIMENTS factor is specified, its main effect and its interactions with LINES and TESTERS are also added to the random model.

The PRINT option specifies the output to be produced. Its settings are mainly the same as those of the PRINT option of the REML directive. There are extra settings aic and sic (with a synonym bic) to print the Akaike and Schwarz (Bayesian) information coefficients, respectively. There is a setting combinability to print the BLUPs for LINES, LINES.TESTERS and any interations with EXPERIMENTS (within CONTROLS, if specified). By default, with this setting, BLUPs are printed for all these terms. However, you can set the PCOMBINABILITYTERMS option to a model formula specifying exactly which ones you want. Finally, there is an extra setting tests to print deviances for LINES, LINES.TESTERS and their interations with EXPERIMENTS, so that you can assess whether their effects are genuinely present.

The PTERMS option operates as in REML, to specify the terms whose means and effects are printed; the default is all the fixed terms. Likewise, the PSE option controls the type of standard error that is displayed with the means and effects; the default is to give a summary of the standard errors of differences.

The MVINCLUDE option controls whether units with missing values in the explanatory factors and variates and/or the y-variate are included in the analysis, as in the REML directive.

REML may be unable to achieve a successful fit if the model contains more random terms than

are actually needed to explain the random variation.(The REML likelihood may be too flat for any clear optimum to be found.) You can guard against having specified an over-complicated random model using the RANDOM option, by setting option RECOVER=yes. VLINEBYTESTER then tries models removing first one term from RANDOM, then two and so on, until successful. Note: it regards a model as successful, if the REML directive returns an exit status of zero (i.e. successful fitting) and there are no bound or aliased variance parameters.

The METHOD option specifies how to choose the random (and spatial) model if there is more than one possible model with the same number of random terms removed:

aic	uses their Akaike information coefficients,
sic or bic	uses their Schwarz (Bayesian) information coefficients (default).

The PRECOVERY option specifies the summary output to be produced about the models that are fitted during recovery. The settings are mainly the same as those of the VRACCUMULATE procedure (which is used to store and then print details of the analyses). There is an extra setting, best, to print the description of the best model. The default is to print the best description, together with the deviance, the Akaike and Schwarz (Bayesian) information coefficients and the number of degrees, for all the models.

The Y parameter specifies the response variate. The COMBINABILITY parameter can save a pointer to tables containing the combinability BLUPs, requested by the PCOMBINABILITYTERMS option. Similarly, the SECOMBINABILITY parameter can save a pointer to tables containing the standard errors of the BLUPs. The DEVIANCES parameter can save the deviances printed by the test setting of PRINT, in a variate. The EXIT parameter allows you to save a code from REML, giving the "exit status" of the fit (zero if successful). Finally, you can save the REML save structure from the analysis with the best model, using the SAVE parameter.

Options: PRINT, PRECOVERY, LINES, TESTERS, CONTROLS, FIXED, RANDOM, CONSTANT, FACTORIAL, EXPERIMENTS, PCOMBINABILITYTERMS, PTERMS, PSE, MVINCLUDE, RECOVER, METHOD.
Parameters: Y, COMBINABILITY, SECOMBINABILITY, DEVIANCES, EXIT, SAVE.

See also
Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedure: VALINEBYTESTER, VARECOVER.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VLSD

Prints approximate least significant differences for REML means (R.W. Payne).

Options

`PRINT` = *string tokens*	Controls printed output (`means`, `sed`, `lsd`, `df`); default lsd
`FACTORIAL` = *scalar*	Limit on the number of factors in each term; default 3
`LSDLEVEL` = *scalar*	Significance level (%) to use in the calculation of least significant differences; default 5
`DFMETHOD` = *string token*	Specifies which degrees of freedom to use for the t-statistics (`fddf`, `given`, `tryfddf`); default `fddf`
`DFGIVEN` = *scalar*	Specifies the number of degrees of freedom to use for the t-statistics when `DFMETHOD=given`, or if d.d.f. are unavailable when `DFMETHOD=tryfddf`
`FMETHOD` = *string token*	Controls how to calculate denominator degrees of freedom for the F-statistics, if these are not already available in the `REML` save structure (`automatic`, `algebraic`, `numerical`); default `auto`
`SAVE` = *REML save structure*	Save structure to provide the table of means; default uses the save structure from the most recent `REML`

Parameters

`TERMS` = *formula*	Treatment terms whose means are to be compared; default `*` takes the `REML` fixed model
`MEANS` = *pointer* or *table*	Saves the means for each term
`SED` = *pointer* or *symmetric matrix*	Saves standard errors of differences between means
`LSD` = *pointer* or *symmetric matrix*	Saves approximate least significant differences matrix for the means
`DF` = *pointer* or *scalar*	Saves the degrees of freedom used to calculate the t critical values for the LSDs
`DDF` = *pointer* or *scalar*	Saves the denominator degrees of freedom in the F test for the term
`DFRANGE` = *pointer* or *scalar*	Saves the range of denominator degrees of freedom in the F tests for the term and any terms that are marginal to the term (available only when denominator degrees of freedom of F-statistics are being used)

Description

`VLSD` calculates least significant differences (LSDs) for predicted means of fixed terms in a `REML` analysis. These are calculated by multiplying standard errors for differences by the t-statistic that would be used to assess whether those differences are non-zero.

The `TERMS` parameter specifies a model formula to define the fixed terms whose predicted means are to be compared. The means are usually taken from the most recent analysis performed by `REML`, but you can set the `SAVE` option to a save structure from another `REML` if you want to examine means from an earlier analysis. As in `VCOMPONENTS`, the `FACTORIAL` option sets a limit on the number of factors in each term (default 3).

The `DFMETHOD` option specifies how to obtain the degrees of freedom for the t-statistics. The default is to use the numbers of denominator degrees of freedom printed by `REML` in the `d.d.f.` column in the table of tests for fixed tests (produced by setting option `PRINT=wald`). The degrees of freedom are relevant for assessing the fixed term as a whole, and may vary over the contrasts amongst the means of the term. So the LSDs should be used with caution. (If you are

interested in a specific comparison, you should set up a 2-level factor to fit this explicitly in the analysis.) The FMETHOD option controls how the denominator degrees of freedom should be calculated, if they are not already available in the REML save structure (e.g. because they were printed in the original analysis). The settings are the same as in the REML and VKEEP directives, except that there is no none setting. (You would set this option only if you really do want to calculate them.)

In some of the more complicated analyses, REML may be unable to calculate the denominator degrees of freedom. You might then want to supply the number of degrees of freedom yourself, using the DFGIVEN option, rather than having no least significant differences at all. For example, you could use the number of denominator degrees of freedom from the analysis of an earlier similar design. However, the results will only be as good as the degrees of freedom that you have supplied, and thus should be used with caution! You can set option DFMETHOD=tryfddf to use the denominator degrees of freedom, if these can be calculated, or those specified by DFGIVEN otherwise. The setting DFMETHOD=given always uses the degrees of freedom specified by DFGIVEN.

Printed output is controlled by the PRINT option, with settings:

means	prints the means;
sed	prints standard errors for differences between the means;
lsd	prints least significant differences for the means;
df	prints the degrees of freedom used to calculate the t critical value required for the LSD, together with the denominator degrees of freedom in the F test for the term if these are not the same.

The significance level to use in the calculation of the least significant differences can be changed from the default of 5% using the LSDLEVEL option.

The MEANS parameter can save the means. If the TERMS parameter specifies a single term, MEANS must be undeclared or set to a table. If TERMS specifies several terms, you must supply a pointer which will then be set up to contain as many tables as there are terms. Similarly the SED parameter can save the standard errors of differences, the LSD parameter can save the approximate least significant differences, the DF parameter can save the degrees of freedom used to calculate the t-statistics, and the DDF parameter can save the denominator degrees of freedom in the F tests.

When a term involves several factors, its means may be be formed from the effects of several terms. For example, the means for the term A.B will involve the effects for the terms A and B (if these are in the model), as well as those for the term A.B. Different contrasts between the means will then have different denominator degrees of freedom. For caution, if VLSD is using the number of denominator degrees of freedom, it uses the smallest number over the terms that are involved in calculating each table of the means. (This corresponds to the largest t-statistic.) If the difference in the t-statistics calculated from smallest and largest numbers of degrees of freedom differ by more than 1%, VLSD prints a warning message. If the denominator degrees of freedom are being used, their range for each term can be saved by the DFRANGE parameter.

Options: PRINT, FACTORIAL, LSDLEVEL, DFMETHOD, DFGIVEN, FMETHOD, SAVE.
Parameter: TERMS, MEANS, SED, LSD, DF, DDF, DFRANGE.

See also
Directive: VDISPLAY.
Procedure: VMCOMPARISON.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VMATRIX

Copies values and row/column labels from a matrix to variates or texts (D.A.Murray).

No options

Parameters

MATRIX = *matrices*, *symmetric matrices* or *diagonal matrices*	Matrices to copy into variates
VARIATE = *variates*	Saves the values from each matrix
ROWS = *variates*	Saves the row coordinates
COLUMNS = *variates*	Saves the column coordinates
ROWLABELS = *texts*	Saves the row labels
COLLABELS = *texts*	Saves the column labels

Method

VMATRIX allows the values and row/column labels in a matrix, symmetric matrix or diagonal matrix to be copied into a set of variates (and texts). The matrix, symmetric matrix or diagonal matrix is supplied using the MATRIX parameter. The values are saved using the VARIATE parameter. The ROWS and COLUMNS parameters save the row and column coordinates, in variates. Similarly the ROWLABELS and COLLABELS allow the row and column labels to be saved, in texts.

Options: none.
Parameters: MATRIX, VARIATE, ROWS, COLUMNS, ROWLABELS, COLLABELS.

Method

EQUATE is used to transfer values.

See also

Procedure: VTABLE.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

VMCOMPARISON

Performs pairwise comparisons between REML means (D.M. Smith).

Options

PRINT = *string tokens*	Controls printed output (comparisons, critical, description, lines, letters, plot, mplot, pplot); default lett
METHOD = *string token*	Test to be performed (fplsd, fulsd, bonferroni, sidak); default fuls
FACTORIAL = *scalar*	Limit on the number of factors in each term; default 3
DIRECTION = *string token*	How to sort means (ascending, descending); default asce
PROBABILITY = *scalar*	The required significance level; default 0.05
STUDENTIZE = *string token*	Whether to use the alternative LSD test where the Studentized Range statistic is used instead of Student's t (yes, no); default no
DFMETHOD = *string token*	Specifies which degrees of freedom to use for the tests (fddf, given, tryfddf); default fddf
DFGIVEN = *scalar*	Specifies the number of degrees of freedom to use for the tests when DFMETHOD=given, or if d.d.f. are unavailable when DFMETHOD=tryfddf
FMETHOD = *string token*	Controls how to calculate denominator degrees of freedom for the F-statistics, if these are not already available in the REML save structure (automatic, algebraic, numerical); default auto
SAVE = *REML save structure*	Save structure to provide the table of means and associated information; default uses the save structure from the most recent REML

Parameters

TERMS = *formula*	Treatment terms whose means are to be compared
MEANS = *pointer* or *variate*	Saves the (sorted) means
DIFFERENCES = *pointer* or *symmetric matrix*	Saves differences between the (sorted) means
LABELS = *pointer* or *text*	Saves labels for the (sorted) means
LETTERS = *pointer* or *text*	Saves letters indicating groups of means that do not differ significantly
SIGNIFICANCE = *pointer* or *symmetric matrix*	Indicators to show significant comparisons between (sorted) means
CIWIDTH = *pointer* or *symmetric matrix*	Saves the width of the confidence interval for the absolute differences between the (sorted) means

Description

VMCOMPARISON calculates comparisons between means estimated in a REML analysis, and tests them with t-statistics using the approximate numbers of residual degrees of freedom that can be printed by REML with the Wald statistics. This corresponds to Fisher's unprotected LSD test, or you can set option METHOD=fplsd to request Fisher's protected LSD test (so that the comparisons are not tested if the fixed term generating the means is not significant). Alternatively, the METHOD settings bonferroni or sidak allow you to use adjusted critical

probability values for the t-statistic that take account of the numbers of comparisons that are being made; see Hsu (1996) page 65.

The TERMS parameter specifies a model formula to define the fixed terms whose predicted means are to be compared. The means (and the necessary associated information) are usually taken from the most recent analysis performed by REML, but you can set the SAVE option to a save structure from another REML if you want to examine means from an earlier analysis. As in VCOMPONENTS, the FACTORIAL option sets a limit on the number of factors in each term (default 3).

The DFMETHOD option specifies how to obtain the degrees of freedom for the tests. The default is to use the numbers of denominator degrees of freedom printed by REML in the d.d.f. column in the table of tests for fixed tests (produced by setting option PRINT=wald). The degrees of freedom are relevant for assessing the fixed term as a whole, and may vary over the contrasts amongst the means of the term. So the results should be used with caution. (If you are interested in a specific comparison, you should set up a 2-level factor to fit this explicitly in the analysis.) The FMETHOD option controls how the denominator degrees of freedom should be calculated, if they are not already available in the REML save structure (e.g. because they were printed in the original analysis). The settings are the same as in the REML and VKEEP directives, except that there is no none setting. (You would set this option only if you really do want to calculate them.)

In some of the more complicated analyses, REML may be unable to calculate the denominator degrees of freedom. You might then want to supply the number of degrees of freedom yourself, using the DFGIVEN option, rather than having no tests at all. For example, you could use the number of denominator degrees of freedom from the analysis of an earlier similar design. However, the results will only be as good as the degrees of freedom that you have supplied, and thus should be used with caution! You can set option DFMETHOD=tryfddf to use the denominator degrees of freedom, if these can be calculated, or those specified by DFGIVEN otherwise. The setting DFMETHOD=given always uses the degrees of freedom specified by DFGIVEN.

Printed output is controlled by the PRINT option, with settings:

comparisons	prints the differences between the pair of means, upper and lower confidence limits for the differences, t-statistics and an indication of whether or not they are significant;
critical	gives critical values for the t-statistic;
description	provides a description including information such as the experiment-wise and compartment-wise error rates;
lines	gives the means, with lines joining those that do not differ significantly;
letters	gives the means, with identical letters (a, b etc.) alongside those that do not differ significantly;
mplot	does a mean-mean scatter plot (synonym plot);
pplot	displays the probabilities in a shade plot.

By default, PRINT=letters.

The means are usually sorted into ascending order, but you can set option DIRECTION=descending for descending order, or DIRECTION=* to leave them in their original order. Note, though, that the lines joining means with non-significant differences may then be broken.

In most REML analyses the standard errors for the differences between the means will be unequal, and the memberships of the groups defined by the lines or letters may then be inconsistent. Suppose, for example, you have ordered means A, B and C. If the s.e.d. for A vs. C is large compared to those for A vs. B and B vs C, you might find that there is no significant difference between A and C, but there are significant differences between A and B, and between B and C. So treatments A and B and treatments B and C would be in different groups. However,

treatments A and C (which are further apart) would be in the same group. This contradicts the idea behind multiple comparisons, where you expect that if two means are in the same group, than any mean between them should be in that group too. If VMCOMPARISON finds inconsistencies like this, it gives a diagnostic and suppresses the printing of lines and letters (but not the other types of output).

The mean-mean scatter plot allows you to assess the confidence region for the difference between each pair of means visually. It has grid lines from both the x- and y-axis at the position of each mean, and a diagonal line at 45 degrees marking y=x. The confidence interval for each pair of means is plotted as a line at an angle of −45 degrees and centred on the intersection above the line y=x of the grid lines for the two means (so the y grid line is for the larger of the two means, and the x grid line is for the smaller mean). The difference between the means is significant if their confidence line does not intersect the line y=x. For more details, see Hsu (1996) pages 151-153.

The shade plot displays the probabilities in a symmetric matrix. The colour of each cell represents the probability for the difference between the means for the treatments in the corresponding row and column.

The PROBABILITY option allows the experiment-wise significance level for the intervals from the Bonferroni and Sidak tests to be changed from the default 0.05 (e.g. to 0.01). For the Fisher's tests, it changes the pair-wise significance level. The STUDENTIZE option can specify that the tests should use the Studentized Range statistic rather than Student's t (for further information see Hsu 1996, page 139).

The MEANS parameter can save the means, sorted according to the DIRECTION option and omitting any that were non-estimable. If the TERMS parameter specifies a single term, MEANS should be set to a variate. If TERMS specifies several terms, you must supply a pointer which will then be set up to contain as many variates as there are terms. Similarly the LABELS parameter can save labels to identify the means, in either a text (for a single term) or in a pointer of texts (for several). Likewise the LETTERS parameter can save texts with the letters identifying means that do not differ significantly, and the SIGNIFICANCE parameter can save symmetric matrices containing ones or zeros according to whether the various comparisons were significant or non-significant. The DIFFERENCES parameter can save symmetric matrices containing the differences between the (sorted) means, and the CIWIDTH parameter can save symmetric matrices containing the widths of the confidence intervals for the differences.

Options: PRINT, METHOD, FACTORIAL, DIRECTION, PROBABILITY, STUDENTIZE, DFMETHOD, DFGIVEN, FMETHOD, SAVE.
Parameters: TERMS, MEANS, DIFFERENCES, LABELS, LETTERS, SIGNIFICANCE, CIWIDTH.

Method
The methodology implemented is based on that described and reviewed in Hsu (1996).

Reference
Hsu, J.C. (1996). *Multiple Comparisons Theory and Methods*. Chapman & Hall, London.

See also
Directive: VDISPLAY.
Procedures: VLSD, AMCOMPARISON, AUMCOMPARISON, MCOMPARISON.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VMETA

Performs a multi-treatment meta analysis using summary results from individual experiments (V.M. Cave).

Options

PRINT = *string tokens*	Controls printed output from the REML analysis (model, components, effects, means, monitoring, vcovariance, deviance, Waldtests, covariancemodels); default mode, comp, cova, mean
PSE = *string token*	Standard errors to be printed with tables of effects and means (differences, estimates, alldifferences, allestimates, none); default alle
EMETHOD = *string token*	Specifies whether the EXPERIMENTS main effect is fitted as a fixed or random term in the REML model; default fixe
VCMODEL = *string token*	Specifies the between-experiment variance-covariance model (identity, diagonal, cs, hcs, unstructured, faequal1, faequal2, fa1); default iden for fixed EXPERIMENTS effects and cs for random effects
INITIAL = *scalars*, *variates*, *matrices*, *symmetric matrices* or *pointers*	Initial parameter values for the variance-covariance model specified by VCMODEL (supplied in the structures appropriate for the model concerned); default generates values automatically
MAXCYCLE = *scalar*	Sets a limit on the number of iterations in the REML analysis; default 30

Parameters

MEANS = *variates*	Supplies the TREATMENTS by EXPERIMENTS means
TREATMENTS = *factors*	Identifier of the treatments factor
EXPERIMENTS = *factors*	Identifier of the experiments factor
SEDMEANS = *variates*	Supplies the (average) standard error of differences in each experiment
VARIANCES = *variates*	Identifier for the variate containing the sampling variance for each experiment
MODERATOR = *factors* or *variates*	Identifier for a moderator variable
SAVE = *REML save structures*	Saves the details of each analysis for use in subsequent VDISPLAY and VKEEP directives

Description

VMETA uses REML to perform a multi-treatment meta analysis, when only the summary results for each treatment (i.e. means and standard error of the difference or sampling variance) are available from the individual experiments. The estimated treatment means for each experiment are supplied, in a variate, using the MEANS parameter. The TREATMENTS and EXPERIMENTS parameters must specify factors indicating the treatment and experiment, respectively, corresponding to each mean.

You must use either the SEDMEANS parameter to supply the (average) standard error of differences for the experiments, or the VARIANCES parameter to supply their sampling variances. You can specify these in a variate with the same length as the number of experiments. Alternatively, you can supply them in a variate with the same length as MEANS. However, this

must contain the same value for the treatments in each experiment.

The EMETHOD option specifies whether the experiment effects are fitted as fixed or random; default fixed. The VCMODEL option specifies the variance-covariance structure used to model the variation and correlation of the between-experiment treatment effects. (See the *Method* Section for details.) The variance-covariance models available depend on the EMETHOD option. When EMETHOD=fixed, the possibilities are identity (default) and diagonal. When EMETHOD=random, they are cs (default), hcs, unstructured, faequal1, faequal2 and fa1. Initial values for the parameters of the variance-covariance model can be supplied by the INITIAL option, which corresponds to the INITIAL parameter of the VSTRUCTURE directive. Default values are generated automatically. For all models, except unstructured, the number of initial values is the number of parameters. However, for the unstructured model, a full covariance matrix of initial values must be given. The initial values must be supplied in the structures appropriate for the model concerned. See the VSTRUCTURE directive for details.

The MODERATOR parameter can be used to supply either an experiment-level factor or a variate that is to be incorporated into the linear mixed model to account for experiment-specific effects on the estimated treatment means. You can specify these in a variate with the same length as the number of experiments. Alternatively, you can supply them in a variate with the same length as MEANS. However, this must contain the same value for the treatments in each experiment.

The PRINT and PSE options controls the output from the REML analyses, with the same settings as the PRINT and PSE options of REML, respectively. The default setting of PRINT gives a description of the model and covariance models that have been fitted, plus estimates of the variance components and the predicted means. The default setting of PSE=allestimates gives the all the standard errors.

The MAXCYCLE option sets a limit on the number of iterations in the REML analysis (default 30). The SAVE parameter can be used to name the REML save structure for later use with the VKEEP and VDISPLAY directives.

Options: PRINT, PSE, EMETHOD, VCMODEL, INITIAL, MAXCYCLE.
Parameters: MEANS, TREATMENTS, EXPERIMENTS, SEDMEANS, VARIANCES, MODERATOR, SAVE.

Method

VMETA uses the methods described in Madden *et al.* (2016). The multi-treatment meta analysis (also known as network meta analysis) is performed on summary results for each treatment (i.e. means and standard error of the difference or sampling variance) using a linear mixed model, fitted by VMETA using the REML, VCOMPONENTS and VSTRUCTURE directives in the usual way.

The treatment term, and if supplied, the moderator term are fitted as fixed, but the experiment term may be fitted as either fixed or random.

The variance-covariance models that can be specified by the VCMODEL option, and subsequently fitted by REML using the VSTRUCTURE directive, are:

Setting	Description	Variance-covariance matrix	Number of parameters
Fixed experiment effects			
identity	Identity	$C_{i,i} = \sigma_\mu^2$ $C_{i,j} = 0$, for $i \neq j$	1
diagonal	Diagonal matrix (heteroscedastic)	$C_{i,i} = \sigma_{\mu(i)}^2$ $C_{i,j} = 0$, for $i \neq j$	m
Random experiment effects			
cs	Compound symmetry	$C_{i,i} = \sigma_\beta^2 + \sigma_\mu^2$ $C_{i,j} = \sigma_\beta^2$, for $i \neq j$	2
hcs	Heterogeneous compound symmetry	$C_{i,i} = \sigma_{T(i)}^2$ $C_{i,j} = \rho\sigma_{T(i)}\sigma_{T(j)}$, for $i \neq j$	$m + 1$
unstructured	Unstructured	$C_{i,i} = \sigma_{T(i)}^2$ $C_{i,j} = \sigma_{T(ij)}$, for $i \neq j$	$m(m + 1)/2$
faequal1	First order factor analytic model with common variance	$C_{i,i} = \gamma_i^2 + \sigma_v^2$ $C_{i,j} = \gamma_i\gamma_j$, for $i \neq j$	$m + 1$
faequal2	Second order factor analytic model with common variance	$C_{i,i} = \gamma_i^{(1)2} + \gamma_i^{(2)2} + \sigma_v^2$ $C_{i,j} = \gamma_i^{(1)}\gamma_j^{(1)} + \gamma_i^{(2)}\gamma_j^{(2)}$, for $i \neq j$	$2m$
fa1	First order factor analytic model	$C_{i,i} = \gamma_i^2 + \sigma_{v(i)}^2$ $C_{i,j} = \gamma_i\gamma_j$, for $i \neq j$	$2m$

In this table $i, j = 1 \ldots m$, where m is the number of treatments.

Action with RESTRICT

VMETA will work with restrictions. However, if more than one variate or factor is restricted, they must be restricted in the same way. In addition, parameters SEDMEANS, VARIANCES and MODERATOR may only be restricted if they supply vectors of the same length as the MEANS variate.

Reference

Madden, L.V., Piepho, H.-P., & Paul, P.A. (2016). Statistical models and methods for network meta-analysis. *Phytopathology*, **106**, 792-806.

See also

Directives: REML, VCOMPONENTS, VSTRUCTURE.

Procedures: META, VAMETA, VASMEANS.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VMODEL

Specifies the model for a REML analysis using a model-definition structure defined by VFMODEL (R.W. Payne).

Option

PRINT = *string tokens*	Controls printed output (model, structure); default * i.e. none

Parameter

MODELSTRUCTURE = *pointer*	Model-definition structure

Description

VMODEL is one of a suite of procedures designed to simplify the assessment of alternative models for a REML analysis. The first step is to form a model-definition structure for each candidate model, using the VFMODEL and VFSTRUCTURE procedures (these define the model settings controlled by the VCOMPONENTS and VSTRUCTURE directives, respectively). The model-definition structures can then be used as input to procedures like VARANDOM, which assesses possible random models. VARANDOM uses VMODEL to specify each model, in turn, so that it can fit it using REML. The relevant results from each fit are saved by the VRACCUMULATE procedure, so that a decision about the recommended random model can be made once they have all been tried.

Printed output is controlled by the PRINT option, with settings:

model	uses the VSTATUS directive to print details of the model that has been specified, and
structure	shows the contents of the model-definition structure (to help check for any errors in the definition).

The model-definition structure is supplied by the MODELDEFINITION parameter.

Option: PRINT.
Parameter: MODELSTRUCTURE.

See also

Directives: REML, VCOMPONENTS, VSTRUCTURE.
Procedures: VARANDOM, VFMODEL, VFSTRUCTURE.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VNEARESTNEIGHBOUR

Analyses a field trial using nearest neighbour analysis (D.B. Baird).

Options

PRINT = *string tokens*	Controls printed output (model, wald, components, means, effects, sed); default mode, wald, comp, mean, effe, sed
NDIFFERENCES = *scalar*	Specifies the number of neighbours to use in differencing the plots, either 1 for first or 2 for second differences; default 1
TMETHOD = *string token*	Indicates how the treatments effects are to be included in the model (fixed, random); default fixe
UMETHOD = *string token*	Whether to include a unit-error term in the model (include, omit); default incl
SEDMETHOD = *string token*	Specifies how the estimates of standard errors of differences of treatment effects are to be calculated (REML, simulation); default REML
NTIMES = *scalar*	Specifies the number of simulations to make; default 100

Parameters

Y = *variates*	Variates to be analysed
TREATMENTS = *factors*	Treatment factor for each y-variate
BLOCKS = *factors*	Block factor for each y-variate, defining groups of plots to be detrended independently
UNITS = *factors*	Unit-within-block factor for each y-variate, defining the order of plots within each block
MEANS = *tables*	Saves the estimated treatment means from each analysis
EFFECTS = *tables*	Saves the estimated treatment effects from each analysis
SED = *matrices* or *symmetric matrices*	Saves the estimated standard errors of differences between treatments
COMPONENTS = *variates*	Saves the estimated variance components from the fitted model
SEED = *scalars*	Seed for the random number generator used in the simulations to calculate standard error of differences; default 0 continues from the previous generation or (if none) initializes the seed automatically

Description

VNEARESTNEIGHBOUR analyses a field trial, whose plots are arranged linearly in blocks, using a one-dimensional nearest neighbour analysis, similar to that of Stroup & Mulitze (1991). However, to avoid bias, VNEARESTNEIGHBOUR estimates the variance parameters by residual maximum likelihood (REML) rather than ordinary maximum likelihood. The original method of nearest-neighbour analysis, due to Papadakis (Papadakis 1937, Bartlett 1938), can be approximated by setting NDIFFERENCES=1 and UMETHOD=omit. However, this is an improvement on the Papadakis method, as the treatments effects and the trend are estimated jointly, instead of estimating the spatial effects from the unadjusted treatment effects (i.e. ignoring any trend).

The NDIFFERENCES option controls the differencing used to detrend the data: 1 for first difference, (y_i - y_{i-1}); or 2 for second differences ($2y_i - y_{i-1} - y_{i+1}$), which give a stronger form of detrending. The first difference model with fixed treatment effects is equivalent to the extended

first difference model of Besag & Kempton (1986) fitted by LVARMODEL.

The model allows for local trends within a row, that the analysis attempts to remove by using a form of smoothing. In the full nearest neighbour model (UMETHOD=include), the degree of smoothing is estimated from the data. Alternatively the reduced model (UMETHOD=omit) applies a full detrending to the data.

The method for fitting the treatment effects is controlled by the TMETHOD option. The random setting treats them as random effects so that best linear unbiased predictors (*BLUP*s) are formed. The fixed setting treats them as fixed effects, thus forming best linear unbiased estimates (*BLUE*s).

The nearest neighbour model treats the data as the sum of up to three components: the treatment effects, a trend component, and a unit-error (i.e. measurement error) term. The unit-error term is included by default, but you can set option UMETHOD=omit to exclude it.

The variable to be analysed is specified by the Y parameter, and the factor defining the treatment on each plot is specified by the TREATMENTS parameter. The BLOCKS parameter specifies the block factor, which defines the groups of plots that are to be detrended separately. The blocks need not all be the same length. The UNITS parameter specifies the units-within-blocks factor, which defines the order of the plots within each block. For example, if the plots are on a rectangular grid and trends are to be removed along rows, the BLOCKS and UNITS factors would be the row and column factors, respectively. If BLOCKS and UNITS are not set, the plots are assumed to be in a single line (and specified sequentially down the line). The procedure can handle missing values in the y-variate but not in the TREATMENTS, BLOCKS or UNITS factors.

The other parameters allow information to be saved from the analysis: MEANS for the table of estimated treatment means; EFFECTS for the table of estimated treatment effects; SED for the standard errors of differences between treatments effects (in either a matrix or a symmetric matrix); and COMPONENTS for the estimated variance parameters. The first variance component is the treatment variance (if TMETHOD=random), the next component is the variance of the plot errors (if UMETHOD=include), and the final component is trend variance component.

Printed output is controlled by the PRINT option with the following settings:

model	prints the fitted model,
wald	prints Wald tests of fixed effects,
components	prints the estimated variance components,
means	prints the estimated treatment means,
effects	prints the estimated treatment effects, and
sed	prints the standard errors of differences of effects.

The option SEDMETHOD controls the estimator used for the standard error of differences between treatment means or effects. The REML setting uses the normal REML estimator. The simulation setting uses an estimator based on simulation; this randomly samples plot and trend components from a Normal model using the estimated variance components. The SEED parameter specifies the seed for the random number generator used in the simulations. The default of zero continues from the previous generation or (if none) initializes the seed automatically. The NTIMES option specifies the number of simulations to make; default 100.

Options: PRINT, NDIFFERENCES, TMETHOD, UMETHOD, SEDMETHOD. NTIMES.
Parameters: Y, TREATMENTS, BLOCKS, UNITS, MEANS, EFFECTS, SED, COMPONENTS, SEED.

Method

A difference matrix is constructed and applied to both the treatment design matrix and Y variate. The model is set up with the VCOMPONENTS directive then estimated by REML.

Action with RESTRICT

The procedure ignores any restrictions, for example, on Y, TREATMENTS, BLOCKS and UNITS.

References

Bartlett, M. (1938). The approximate recovery of information from replicated field experiments with large blocks. *The Journal of Agricultural Science*, **28**, 418-427.

Besag, J.E. & Kempton, R.A. (1986). Statistical analysis of field experiments using neighbouring plots. *Biometrics*, **42**, 231-251.

Papadakis, J.S. (1937). Méthode statistique pour des expériences sur champ. *Bull. Inst. Amel. Plantes a Salonique*, **23**, 13-29.

Stroup, W.W. & Mulitze, D.K. (1991). Nearest neighbor adjusted best linear unbiased prediction. *The American Statistician*, **45**, 194-200.

See also

Directives: REML, VSTRUCTURE.

Procedure: LVARMODEL.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VORTHPOLYNOMIAL

Forms orthogonal polynomials over time for repeated measures (J.T.N.M. Thissen).

Options

TIMEPOINTS = *variate*	Variate of timepoints; default uses the suffixes of the DATA pointer
MAXDEGREE = *scalar*	The number of contrasts (excluding the mean); default is the number of identifiers in the CONTRAST pointer minus 1

Parameters

DATA = *pointers*	Each pointer contains the data variates (observed at successive times); must be set
CONTRAST = *pointers*	To save the calculated contrasts: the first variate contains the means, the second the linear polynomial contrasts, the third the quadratic polynomial contrasts etc; must be set

Description

A repeated measures experiment is one in which the same set of units, or subjects, is observed at a sequence of times to investigate treatment effects over a period of time. VORTHPOLYNOMIAL calculates orthogonal polynomial contrasts in time for each experimental unit. These contrasts can then be analysed given the block and treatment structure at each timepoint.

The observed data is specified in a pointer containing a set of variates, each one containing the measurements made on the subjects at one of the occasions on which they were observed, and input to the procedure using the DATA parameter. The variate in option TIMEPOINTS specifies the actual times when the measurements were taken. If this is not specified, the suffixes of the DATA pointer are taken as values for the timepoints.

The calculated contrasts are saved in a pointer which must be specified by the CONTRAST parameter. This points to a list of variates: the first variate saves the means over the DATA variates, the second variate saves the linear polynomial contrast, the third the quadratic polynomial, and so on. Provided the MAXDEGREE option is specified, the CONTRAST need not be declared in advance. The suffixes of the pointer are then defined to be 0, 1, 2 ...

The number of contrasts can be specified using option MAXDEGREE and should be less than the number of timepoints. The default setting is the number of identifiers in the pointer specified by the CONTRAST parameter minus 1. If MAXDEGREE is set, and the CONTRAST pointer has been declared to be of length less than MAXDEGREE+1, a fault message is produced.

If an experimental unit has a missing value in one of the DATA variates each contrast in the CONTRAST pointer (including the mean) gets a missing value for this unit.

Options: TIMEPOINTS, MAXDEGREE.
Parameters: DATA, CONTRAST.

Method

Procedure ORTHPOLYNOMIAL gets orthogonal polynomial coefficients which are used to form the contrasts.

Action with RESTRICT

Each variate in the DATA pointer should be restricted in the same way. The saved orthogonal polynomial contrasts are restricted accordingly. The TIMEPOINTS variate must not be restricted.

See also
Procedure: ORTHPOLYNOMIAL.
Functions: POL, POLND, REG.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Repeated measurements.

†VPERMTEST

Does random permutation tests for the fixed effects in a REML analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (prwald, criticalwald, ownstatistics, monitoring); default prwa, crit
NTIMES = *scalar*	Number of permutation samples to make; default 99
NRETRIES = *scalar*	Maximum number of extra samples to take when some REML analyses fail to converge; default NTIMES
BLOCKSTRUCTURE = *formula*	Model formula defining any blocking to consider during the randomization; default none
EXCLUDE = *factors*	Factors in the block formula whose levels are not to be randomized
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
WMETHOD = *string token*	Controls which Wald statistics are used (add, drop); default add
OWNMETHOD = *string token*	Type of test required for own statistics (twosided, greaterthan, lessthan); default twos
CIPROBABILITY = *scalar*	Probability level for the confidence interval for own statistics; default 0.95

Parameters

SAVE = *REML save structures*	Specifies the (REML) save structure of the original analysis; default * uses the SAVE structure from the most recent REML analysis
WALD = *pointers*	Wald statistics saved in a pointer with a variate for each term
PRWALD = *pointers*	Critical values for Wald statistics saved in a pointer with a scalar for each term
CRITICALWALD = *pointers*	Saves a pointer with variates for the 5%, 1% and 0.1% significance levels containing the corresponding critical values for the fixed terms, obtained from the quantiles of the Wald statistics from the permuted data sets
NNOTCONVERGED = *scalars*	Saves the number of permutations whose REML analysis failed to converge
OWNDATA = *pointers*	Data required to calculate own statistics
OWNOBSERVEDVALUES = *variates*	Saves observed values of the own statistics
OWNPROBABILITIES = *variates*	Saves probabilities for the own statistics
OWNESTIMATES = *variates*	Saves estimates for the own statistics
OWNSES = *variates*	Saves standard errors for the own statistics
OWNLOWERCIS = *variates*	Saves lower values of the confidence intervals for the own statistics
OWNUPPERCIS = *variates*	Saves upper values of the confidence intervals for the own statistics
OWNSTATISTICS = *pointers*	Saves the own statistics obtained from the permutation samples, in a pointer with a variate for each statistic

Description

VPERMTEST performs a random permutation test for fixed effects in a REML analysis. The SAVE parameter can supply the save structure from the original analysis; if this is not set, the tests are done for the most recent REML analysis.

The test probabilities are calculated by taking the proportion of Wald statistics in the permutation samples that are larger than the observed Wald statistic of each fixed term. (As a result these should not suffer from the bias that is found in the probabilities for the Wald statistics themselves, which tend to be too low.) The WMETHOD option controls whether the Wald statistics are obtained from the table where terms are added sequentially (the default), or from the table where suitable terms are dropped from the full fixed model. Note that, if you use the table where terms are dropped, the only terms that can be tested are those that are not marginal to any other term in the fixed model: for example, the main effect A cannot be tested if the model contains an interaction, such as A.B.

The NTIMES option defines how many random permutations to perform; by default there are 99 (as well as the "null" permutation where the data keep their original order). The NRETRIES option specifies the maximum number of extra samples to take when some REML analyses fail to converge; the default is to use the same number as specified by NTIMES. The SEED option allows you to specify the seed to use for the random-number generator that is used to construct the permutation samples. The default, SEED=0, continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, it initializes the seed automatically. If NTIMES exceed the maximum possible number of permutations for the data, an "exact" test is performed in which every permutation is used once. This is feasible only for small datasets. There are *n*! (n factorial) permutations of *n* units: 3!=6, 4!=24, 5!=120, 6!=720, 7!=5040, 8!=40320, and so on. The NNOTCONVERGED parameter can save the number of samples whose analyses did not converge, in a scalar.

If the data are from a designed experiment, you may need to use the BLOCKSTRUCTURE option to specify a block model to define how to do the randomization. The EXCLUDE option can then restrict the randomization so that one or more of the factors in the block model is not randomized. See the RANDOMIZE directive for further details.

Output is controlled by the PRINT option, with settings:

prwald	to print the probabilities calculated from the distribution of the Wald statistics;
criticalwald	to print a table giving estimated critical values for each of the Wald statistics, formed from the permutation samples;
ownstatistics	to print the estimates, standard errors and confidence intervals for the own statistics, and
monitoring	to monitor the progress of the test.

The Wald statistics from the permutation tests can be saved, in a pointer with a variate for each of the FIXEDTERMS, using the WALD parameter. The probabilities calculated from the tests can be saved, in a pointer with a scalar for each of the FIXEDTERMS, using the PRWALD parameter.

You can define your own statistics to be assessed by the test. They are calculated by a procedure _VPERMownstatistics, which is called by VPERMTEST following the REML analysis of each permutation sample. Its use is shown in the VPERMTEST example, which can be modified to calculate your own statistics instead. The information required by _VPERMownstatistics to do the calculations is supplied, in a pointer, by the OWNDATA parameter. The OWNMETHOD option specifies the type of test to be made. The default, twosided tests whether the statistics differ from zero. The greaterthan setting tests whether they are greater than zero, and the lessthan setting tests whether they are less than zero. Standard errors and confidence intervals are also calculated, The CIPROBABILITY option specifies the probability for the confidence intervals (default 0.95). The OWNOBSERVEDVALUES parameter can save a variate containing the

values of the own statistics from the original data set. The OWNPROBABILITIES can save a variate containing the probabilities from the tests. The OWNESTIMATES can save a variate containing the bootstrap estimates of the statistics (calculated as the mean of the values obtained from the bootstrap samples) The OWNSES can save a variate containing standard errors of bootstrap estimates. The OWNLOWERCIS and OWNUPPERCIS parameters can save variates containing the lower and upper values, respectively, of the confidence intervals. Finally, the OWNSTATISTICS can save the values of the own statistics obtained from the bootstrap samples, in a pointer with a variate for each statistic.

The maximum number of iterations (MAXCYCLE) and number of blocks of internal memory to be (WORKSPACE) to be used in the REML analyses can be set by a call to the VAOPTIONS procedure before you use VPERMTEST.

Options: PRINT, NTIMES, NRETRIES, BLOCKSTRUCTURE, EXCLUDE, SEED, WMETHOD, OWNMETHOD, CIPROBABILITY.

Parameters: SAVE, WALD, PRWALD, CRITICALWALD, NNOTCONVERGED, OWNDATA, OWNOBSERVEDVALUES, OWNPROBABILITIES, OWNESTIMATES, OWNSES, OWNLOWERCIS, OWNUPPERCIS, OWNSTATISTICS.

See also

Directive: REML.

Directives: REML, VCOMPONENTS.

Procedures: VAOPTIONS, VBOOTSTRAP, VRPERMTEST.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VPLOT

Plots residuals from a REML analysis (S.J. Welham).

Options

RMETHOD = *string token*	Which random terms to use when calculating the residuals (final, all, notspline, stfinal, stall); default uses the setting from the REML statement
INDEX = *variate* or *factor*	X-variable for an index plot; default !(1,2...)
GRAPHICS = *string token*	What type of graphics to use (lineprinter, highresolution); default high
TITLE = *text*	Overall title for the plots; if unset, the identifier of the y-variate is used
SAVE = *REML save structure*	Specifies the (REML) save structure from which the residuals and fitted values are to be taken; default * uses the SAVE structure from the most recent REML analysis

Parameters

METHOD = *string tokens*	Type of residual plot (fittedvalues, normal, halfnormal, histogram, absresidual, index); default fitt, norm, half, hist
PEN = *scalars*, *variates* or *factors*	Pen(s) to use for each plot

Description

Procedure VPLOT provides up to four types of residual plots from a REML analysis. These are selected using the METHOD parameter, with settings: fitted for residuals versus fitted values, normal for a Normal plot, halfnormal for a half-Normal plot, histogram for a histogram of residuals, absresidual for a plot of the absolute values of the residuals versus the fitted values, and index for a plot against an "index" variable (specified by the INDEX option). The PEN parameter can specify the graphics pen or pens to use for each plot. The TITLE option can supply an overall title. If this is not set, the identifier of the y-variate is used.

The residuals and fitted values are accessed automatically from the analysis specified by the SAVE option. If the SAVE option has not been set, they are taken from the SAVE structure from the most recent REML analysis.

The RMETHOD option controls which random terms are used to calculate the residuals:

all	all the random effects,
final	only the final random term,
notspline	all except any random spline terms,
stall	standardized residuals using all the random effects, and
stfinal	standardized residuals using only the final random term.

The default takes the setting from the REML directive that produced the analysis. Note that residuals based on the final random term will not be calculated when any of the variance components are negative, as the associated negative correlations can generate very misleading patterns. VPLOT will then generate a warning that all the residuals are missing, and you should use RMETHOD=all instead.

By default, high-resolution graphics are used. Line-printer graphics can be used by setting option GRAPHICS=lineprinter.

Options: RMETHOD, INDEX, GRAPHICS, TITLE, SAVE. Parameters: METHOD, PEN.

Method
Residuals and fitted values effects are accessed, using VKEEP or VFRESIDUALS, from the REML analysis specified by the SAVE option. The plots are produced using the DRESIDUALS procedure.

Action with RESTRICT
If the y-variate in the REML analysis was restricted, then only units included by the restriction will be used in the graphs.

See also
Procedures: VDEFFECTS, VDFIELDRESIDUALS, VFRESIDUALS, VGRAPH, APLOT, DRESIDUALS, RCHECK.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VPOWER

Uses a parametric bootstrap to estimate the power (probability of detection) for terms in a REML analysis (R.W. Payne & C.J. Brien).

Options

PRINT = *string tokens*	Controls printed output (power, nnotconverged, monitoring); default powe
VPRINT = *string tokens*	Controls the output from the REML analyses (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default * i.e. none
TERM = *formula*	Fixed term to be assessed in the analysis
UVCOVARIANCE = *symmetric matrix*	Specifies the variances and covariances of the units; default is to take this from the SAVE structure
PROBABILITY = *scalar*	Significance level at which the response is to be detected; default 0.05
TMETHOD = *string token*	Type of test to be made (fratio, wald, twosided, greaterthan, lessthan, equivalence, noninferiority); default frat
XCONTRASTS = *variate*	X-variate defining a contrast to be detected
CONTRASTTYPE = *string token*	Type of contrast (regression, comparison) default rege
CRITICALVALUE = *scalar*	Supplies a critical value for the test statistic
NBOOT = *scalar*	Number of bootstrap samples to analyse; default 500
NRETRIES = *scalar*	Maximum number of extra samples to take when some REML analyses fail to converge; default NBOOT
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
METHOD = *string token*	Indicates whether to use the standard Fisher-scoring algorithm or the new AI algorithm with sparse matrix methods (Fisher, AI); default AI
MAXCYCLE = *scalar*	Sets a limit on the number of iterations in the REML analyses; default 30
FMETHOD = *string token*	Controls whether and how to calculate F statistics for fixed terms (automatic, none, algebraic, numerical); default auto
WMETHOD = *string token*	Controls which Wald statistics are saved (add, drop); default add
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for use by the REML algorithm
SAVE = *vsave*	REML save structure to provide the unit-by-unit variance-covariance matrix if UVCOVARIANCE is not specified

Parameters

RESPONSE = *scalars*, *variates* or *tables*	Specifies the response to be detected
POWER = *scalars*	Saves the power (i.e. probability of detection) for RESPONSE
NCONVERGED = *scalars*	Saves the number of bootstrap samples whose REML

	analyses converged
NNOTCONVERGED = *scalars*	Saves the number of bootstrap samples whose REML analyses failed to converge

Description

When assessing an experimental design, it can be useful to know how likely a fixed response of a specified size is to be detected. This probability of detection, known as the *power* of the design with respect to the response of interest, helps to determine whether the experiment is sufficiently large or accurate to achieve its purpose.

VPOWER performs a parametric bootstrap to allow the power to be estimated, for designs whose results will be analysed by REML. The model to be fitted must be defined using the VCOMPONENTS and VSTRUCTURE directives, in the usual way. The bootstrap samples are generated from a multivariate Normal distribution with dimension equal to the number of units in the analysis. The UVCOVARIANCE option supplies the variances and covariances of the units. If UVCOVARIANCE is not specified, the default is the unit-by-unit variance-covariance matrix from the REML analysis supplied by the SAVE option, or from the most recent REML if SAVE is not set. (See the UVCOVARIANCE option of VKEEP). Note: you can use the VUVCOVARIANCE procedure to form the variance-covariance matrix, if you know the variance components for a REML model that contains no covariance models.

The NBOOT option specifies the number of bootstrap samples to take (default 500). The NRETRIES option specifies the maximum number of extra samples to take when some REML analyses fail to converge; the default is to use the same number as specified by NBOOT. The SEED option supplies the seed for the random number generator used to form the samples; default 0 continues from the previous generation or (if none) initializes the seed automatically.

The fixed term to be tested is specified using the TERM option of VPOWER, and the response to be detected is specified by the RESPONSE parameter. This can supply a scalar to specify the maximum difference between the effects of the term, it can supply a table, to specify the anticipated effects themselves, or it can supply a variate with the effects entered in to the relevant units of the design. As an alternative to detecting a difference between its effects, you can ask to detect a contrast. RESPONSE must then supply a scalar, and TERM must be a main effect (that is, it must involve just one factor). The XCONTRASTS option must specify a variate or table containing the coefficients defining the contrast, and the CONTRASTTYPE option indicates whether this is a regression contrast (as specified by the REG function) or a comparison (as specified by COMPARISON).

The TMETHOD option specifies the type of test that is to be used to assess the term, with the following settings.

fratio	assumes that the term will be tested using its F ratio (default).
wald	assumes that the term will be tested by a Wald test.
twosided	assumes a two-sided test to assess whether a contrast of the term differs from zero (default).
lessthan	assumes a one-sided test to assess whether a contrast of the term is less than zero.
greaterthan	assumes a one-sided test to assess whether a contrast of the term is greater than zero.
noninferiority	assumes a test to check that a contrast of the term is not significantly less then zero. (See Method for more details.)
equivalence	assumes a one-sided test to check that a contrast of the term does not differ significantly from zero; see Method for more details.

The PROBABILITY option specifies the significance level to be used in the test; the default

is 0.05, i.e. 5%. The CRITICALVALUE option can supply the critical value to be used in the test. (The VCRITICAL procedure can obtain this using a similar parametric bootstrap process to that used by VPOWER.). If CRITICALVALUE is not set, the critical value is obtained in the conventional way, using an F, chi-square or t-distribution, according to the type of test.

The VPRINT option controls the output from the REML analyses of the bootstrap samples, with the same settings as the PRINT option of REML. By default, nothing is printed.

The MAXCYCLE option sets a limit on the number of iterations in the REML analyses (default 30). The METHOD option controls whether REML uses the Fisher-scoring algorithm, or the AI algorithm with sparse matrix methods (the default). The WMETHOD option controls whether the Wald and F statistics are obtained from the table where terms are added sequentially (the default), or from the table where suitable terms are dropped from the full fixed model. Note that, if you use the table where terms are dropped, the TERM must not be not marginal to any other term in the fixed model: for example, the main effect A cannot be tested if the model contains an interaction, such as A.B. The FMETHOD option controls how to estimate the denominator degrees of freedom for the F tests. (This is relevant if TMETHOD=fratio, or if tests for fixed effects are being printed in the REML analyses of the bootstrap samples.) The WORKSPACE option specifies the number of blocks of internal memory to be set up for use by the REML algorithm. The default is to use the same value as in the SAVE structure, if SAVE has been set. Otherwise, it uses the value from the most recent REML analysis, or the standard REML default if there has been no analysis.

Printed output is controlled buy the PRINT option, with the following settings.

power	prints the estimated power.
nnotconverged	prints the number of bootstrap samples whose analysis failed to converge.
monitoring	prints monitoring information, showing the progress of the bootstrap sampling.

By default, the power is printed.

The POWER parameter can save the power, in a scalar. The NCONVERGED and NNOTCONVERGED parameters can save the number of samples whose analyses converged, or failed to converge, respectively.

Options: PRINT, VPRINT, TERM, UVCOVARIANCE, PROBABILITY, TMETHOD, XCONTRASTS, CONTRASTTYPE, CRITICALVALUE, NBOOT, NRETRIES, SEED, METHOD, MAXCYCLE, FMETHOD, WMETHOD, WORKSPACE, SAVE.
Parameters: RESPONSE, POWER, NCONVERGED, NNOTCONVERGED.

Method

The power is estimated by seeing how frequently the relevant test would be significant in the analyses of the bootstrap samples.

With an equivalence test, you define a threshold h below which two treatments can be assumed to be equivalent. The contrast c would be the difference between the treatments, and the null hypothesis that the treatments are not equivalent is that either

$$c \leq -t$$

or

$$c \geq t$$

with the alternative hypothesis that they are equivalent, i.e.

$$-t < c < \mathrm{t}$$

This defines an *intersection-union* test, in which each component of the null hypothesis must be rejected separately. This implies performing two one-sided t-tests (this is known as a *TOST* procedure). If the significance level for the full test is to be α, each t-test must have significance level α (see Berger & Hsu 1996).

With a non-inferiority test, you again define the threshold *t* for the effect of the new treatment to be inferior to the standard treatment, and a contrast representing the effect of the new test minus the effect of the standard treatment. The null hypothesis is

$$-c \geq t$$

which represents a one-sided "less-than" t-test.

Reference

Berger, M.L. & Hsu, J.C. (1996). Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Statistical Science*, **11**, 283-319.

See also

Directive: `REML`.

Procedures: `APOWER`, `RPOWER`, `VCRITICAL`, `VSAMPLESIZE`, `VUVCOVARIANCE`.

Genstat Reference Manual 1 Summary sections on: REML analysis of linear mixed models, Design of experiments.

VRACCUMULATE

Forms a summary accumulating the results of a sequence of REML random models (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (deviance, aic, bic, sic, dffixed, dfrandom, change, exit); default devi, aic, sic, dfra
METHOD = *string token*	How to accumulate the current analysis (add, printonly, restart); default add
INCLUDE = *string tokens*	Which constants to include that depend only on the fixed model (determinant, pi); default pi
DMETHOD = *string token*	Method to use to calculate log(determinant(X′X)) (choleski, lrv); default chol
ACCUMULATED = *pointer*	Saves the summary

Parameters

DESCRIPTION = *text*	Single-line text to describe the analysis; default lists the random terms added or deleted from the previous model
SAVE = *REML save structure*	Save structure for the REML analysis to put into the summary; default uses the save structure from the most recent REML

Description

Random models in a REML analysis can be assessed by examining their Akaike or Schwarz (Bayesian) information coefficients or, if one random model is a generalization of another random model, you can look at the change in their deviances. These statistics can be calculated and printed after each analysis using the VAIC procedure. However, making comparisons can be inconvenient if there are many models to compare. Thus this procedure, VRACCUMULATE, allows you to accumulate results from a sequence of models, so that you can view them all at once. You can do this by giving the command

```
VRACCUMULATE [PRINT=*]
```

following all except the last analysis. Then, after the last analysis, give another VRACCUMULATE command, but with the PRINT option now set to request the desired output, using the following settings:

deviance	prints the deviances;
aic	prints the Akaike information coefficients;
bic or sic (synonyms)	print the Schwarz (Bayesian) information coefficients;
dffixed	prints the number of parameters fitted in the fixed models;
dfrandom	prints the number of parameters fitted in the random models (and any covariance models);
change	prints changes in the deviance and number of random d.f. between successive lines of the summary and their (chi-square) probabilities; and
exit	exit codes (from VKEEP) indicating whether each analysis was fitted successfully (the deviance and information coefficients are set to missing values for unsuccessful fits).

The output indicates any point during the sequence of analyses where the fixed model has changed. It is not valid to compare random models unless one of the models is an extension of the other one, and the fixed model remained unchanged; if VRACCUMULATE detects that a comparison is invalid, the change in deviance is set to a missing value. It also flags any lines

where it detects that there have been changes in the variance models (defined by VSTRUCTURE); before you use the change in deviance between these lines, you should check that the variance model defined in one of the lines is an extension of the model defined in the other one.

To print the information without adding another line to the summary, you can set option METHOD=printonly. Setting METHOD=restart clears the existing summary and then adds the current analysis. The default, METHOD=add, continues the existing summary by adding another line.

The deviance provided by REML omits some constants that depend on the fixed model. In fact the full deviance is given by

full-deviance = REML-deviance + $(n-p)$*log(2π) - log(det(X′X))

where X is the design matrix of the fixed model. Other software systems tend to include the first term, involving π, but omit the log-determinant term which is more time-consuming to calculate. The inclusion of these terms in the calculation is controlled by the INCLUDE option, with settings

determinant	−log(det(X′X))
pi	+(n−p)*log(2π)

The DMETHOD option controls how −log(det(X′X)) is calculated when this is included. However, the default is INCLUDE=pi. The INCLUDE option also affects the values of the Akaike or Schwarz (Bayesian) information coefficients, which depend on the deviance; see VAIC for details.

By default, the first line of the summary is labelled by the list of random terms in the model; subsequent lines list the random terms added or deleted from the previous model. Alternatively, you can supply your own labels using the DESCRIPTION parameter.

VRACCUMULATE usually adds a line to the summary for the most recent REML analysis. However, you can use the SAVE parameter to specify the save structure from an earlier analysis.

The ACCUMULATED option allows you to save the summary in a pointer, with elements labelled 'description', 'deviance', 'aic', 'sic', 'dffixed', 'dfrandom', 'deviance change', 'd.f. change', 'fixed changed', 'var-mod. changed' and 'exit'. ACCUMULATED['description'] is a text. The other elements are variates. The saved values of the deviances and information coefficients all take account of the settings of the INCLUDE option.

Options: PRINT, METHOD, INCLUDE, DMETHOD, ACCUMULATED.
Parameters: DESCRIPTION, SAVE.

See also
Procedure: VAIC.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRADD

Adds terms from a REML fixed model into a Genstat regression (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated); default mode, summ, esti, accu
FACTORIAL = *scalar*	Limit for expansion of terms; default 3
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
SELECTION = *string tokens*	One or two criteria to be printed with the models (%variance, %ss, adjustedr2, r2, dispersion, aic, sic, bic); default %var, aic, sic

Parameter

TERMS = *formula*	Fixed terms to be added

Description

VRADD is one of several procedures designed to improve the process of determining the appropriate fixed terms to include in a REML analysis. (The others are VRFIT, VRDROP, VRDISPLAY, VRKEEP, VRSETUP, VRSWITCH and VRTRY.) They do this by a generalized regression analysis, with a weight matrix based on variances estimated from the original REML analysis (with the full fixed model). See VRFIT for details.

Before fitting any terms, the VRSETUP procedure must be called to make some checks, and initialize the regression by specifying a MODEL command with the necessary weight matrix and a TERMS command with the full fixed model. It also uses the WORKSPACE directive to set up a Genstat workspace structure to store control information and results. However, VRFIT will call VRSETUP for you, if you have not done so already. The analysis will then be based on the most recent REML analysis. To use an earlier analysis, you should call VRSETUP yourself, setting its SAVE option set to the save structure of the required REML analysis.

In principle the VRFIT procedure should also be called before VRADD is used. However, VRADD will call VRFIT with a null model (i.e. only the constant) if VRFIT has not been used already. So you can start investigating the fixed model just by calling VRADD (and VRFIT and VRSETUP will be called for you, automatically).

The terms to be fitted are specified by the TERMS parameter, in a similar way to the FIT directive. The FACTORIAL option sets a limit (by default 3) on the number of factors and variates in each term. Terms containing more than that number are omitted.

The PRINT option controls printed output as in the regression directives, except that some irrelevant settings are omitted. (For example, grid is relevant only to the fitting of nonlinear models.) See VRDISPLAY for more details.

The DENOMINATOR option specifies how the residual is selected for the accumulated analysis of variance By default it is taken from the model with the smallest number of residual degrees of freedom. However, you can set DENOMINATOR=ms to take it from the model with the smallest residual mean square.

The SELECTION option specifies the statistics to be displayed in the summary of analysis as in the regression directives, except that again some irrelevant settings are omitted. See VRDISPLAY for more details.

Options: PRINT, FACTORIAL, DENOMINATOR, SELECTION.
Parameter: TERMS.

Action with RESTRICT
Any restriction applied to vectors used in the REML analysis will apply also to the results from VRADD.

See also
Directives: FIT, REML.
Procedures: VRFIT, VRDISPLAY, VRDROP, VRKEEP, VRSETUP, VRSWITCH, VRTRY.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRCHECK

Checks effects of a random term in a REML analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (largeblups, stability); default larg
TERM = *formula*	Random term whose BLUPs are to be assessed; must be set
RMETHOD = *string token*	Which random terms to use to form the residuals that are subtracted from the y-variate to provide the fitted values (all, term); default all
RLIMIT = *scalar*	Limit for detection of large standardized BLUPs; if this is not set, the limit is set automatically according to the number of BLUPs
NLARGEBLUPS = *scalar*	Saves the number of large standardized BLUPs that have been detected
LARGEBLUPUNITS = *pointer*	Saves the factor levels of the large standardized BLUPs
STABILITYTEST = *pointer*	Saves the results of the Levene test for stability of the variance of the standardized BLUPs
SAVE = *REML save structure*	Specifies the analysis from which the BLUPs are to be taken; by default this will be the most recent REML

No parameters

Description

Procedure VRCHECK checks effects (i.e. BLUPs) of a random term from a REML analysis. The TERM option must be set to specify the random term to check. By default, its BLUPs are taken from the recent REML analysis. However, you can use an earlier analysis, by using the SAVE option of VRCHECK to specify its save structure (saved using the SAVE parameter of the earlier REML command).

Output is controlled by the PRINT option, with the following settings.

largeblups	reports any large standardized BLUPs.
stability	performs two Levene tests to check whether the variance of the random term differs according to the size of the response. The BLUPs are divided into three groups (small, intermediate and large) according to the sizes of the corresponding fitted values. The tests compare the variance of the standardized BLUPs in the first (small) group with those in the third (large) group, and the variance of the second (intermediate) group with the variance of other two groups combined.

By default PRINT=largeblups.

The RMETHOD option specifies how to form the residuals that are subtracted from the y-variate to provide the fitted values. The available settings are:

all	uses all of the random effects (default), and
term	uses only the random term specified by the TERM option.

It is important to realise that the estimated BLUPs will be correlated. The Levene tests assume that they are independent Normally-distributed observations. Their test probabilities may therefore be too low – and generate too many significant results. They should thus be interpreted with care.

The RLIMIT option specifies the limit that must be exceeded by the absolute value of a

standardized BLUP for it to be identified as large. If this is not set, the default is taken as 2.0 if the number of BLUPs is less than 20, and 4.0 if *d* is greater than 15773. For other values of *d*, the default is the critical value of the Normal distribution for a two-sided test with significance probability 1/*d*. These calculations are the same as those used in regression and analysis of variance, and are intended to ensure that a report should appear for any extreme BLUP, but that reports should not appear too often just as a result of random variation.

The NLARGERESIDUALS option saves the number of large standardized BLUPs that have been found. The LARGEBLUPUNITS option can save a pointer containing their factor levels. The pointer contains a factor for every factor in the table of BLUPs, and its elements are labelled by the factor names. The results of the Levene tests for stability of the variance of the standardized residuals can be saved, in a pointer, by the STABILITYTEST option.

Options: PRINT, TERM, RMETHOD, RLIMIT, NLARGEBLUPS, LARGEBLUPUNITS, STABILITYTEST, SAVE.
Parameters: none.

Method

Details about Levene tests can be found in Snedecor & Cochran (1989); also see O'Neill & Mathews (2002) for information about the issues that arise in their use in balanced analysis of variance.

Action with RESTRICT

If the y-variate in the REML was restricted, only the BLUPs not excluded by the restriction will be included in the checks.

References

O'Neill, M.E. & Mathews, K.L. (2002) Levene tests of homogeneity of variance for general block and treatment designs. *Biometrics*, **58**, 216-224.

Snedecor, G.W. & Cochran, W.G. (1989). *Statistical Methods (eighth edition)*. Iowa State University Press, Ames.

See also

Directive: REML.
Procedures: ACHECK, VCHECK, VPLOT, VSOM.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRDISPLAY

Displays output for a REML fixed model fitted in a Genstat regression (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated); default mode, summ, esti, accu
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
SELECTION = *string tokens*	One or two criteria to be printed with the models (%variance, %ss, adjustedr2, r2, dispersion, aic, sic, bic); default %var, aic, sic

No parameters

Description

VRDISPLAY displays output from models fitted by procedures VRFIT, VRADD, VRDROP, VRSWITCH and VRTRY, which are designed to improve the process of determining the appropriate fixed terms to include in a REML analysis. They do this by a generalized regression analysis, with a weight matrix based on variances estimated from the original REML analysis (with the full fixed model). See VRFIT for details.

The PRINT option controls printed output as in the regression directives, except that some irrelevant settings are omitted. The available settings are as follows.

model	description of the currently fitted model, including response and explanatory variates,
deviance	abbreviated summary analysis of variance,
summary	summary analysis of variance,
estimates	estimates of the parameters in the model,
correlations	correlation matrix of the parameter estimates,
fittedvalues	table with unit labels, values of response variate, fitted values, standardized residuals and leverages, and
accumulated	analysis of variance table showing the various changes that have been made to the model.

The DENOMINATOR option specifies how the residual is selected for the accumulated analysis of variance By default it is taken from the model with the smallest number of residual degrees of freedom. However, you can set DENOMINATOR=ms to take it from the model with the smallest residual mean square.

The SELECTION option specifies the statistics to be displayed in the summary of analysis as in the regression directives, except that again some irrelevant settings are omitted. The available settings are as follows.

%variance	percentage variance accounted for by the currently fitted model,
%ss	percentage sum of squares accounted for,
adjustedr2	proportion of variance accounted for (i.e. %variance / 100),
r2	proportion of the sum of squares accounted for (i.e. %ss / 100),
dispersion	dispersion parameter (which is equal to the residual mean square for an ordinary regression like this),
aic	Akaike information criterion, and

`sic` or `bic` Schwarz (Bayesian) information criterion.

Options: `PRINT`, `DENOMINATOR`, `SELECTION`.
Parameters: none.

See also
Directives: `RDISPLAY`, `REML`.
Procedures: `VRADD`, `VRDROP`, `VRFIT`, `VRKEEP`, `VRSETUP`, `VRSWITCH`, `VRTRY`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRDROP

Drops terms in a REML fixed model from a Genstat regression (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated); default mode, summ, esti, accu
FACTORIAL = *scalar*	Limit for expansion of terms; default 3
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
SELECTION = *string tokens*	One or two criteria to be printed with the models (%variance, %ss, adjustedr2, r2, dispersion, aic, sic, bic); default %var, aic, sic

Parameter

TERMS = *formula*	Fixed terms to be dropped

Description

VRDROP is one of several procedures designed to improve the process of determining the appropriate fixed terms to include in a REML analysis. (The others are VRFIT, VRADD, VRDISPLAY, VRKEEP, VRSETUP, VRSWITCH and VRTRY.) They do this by a generalized regression analysis, with a weight matrix based on variances estimated from the original REML analysis (with the full fixed model). See VRFIT for details.

The terms to be dropped are specified by the TERMS parameter, in a similar way to the FIT directive. The FACTORIAL option sets a limit (by default 3) on the number of factors and variates in each term. Terms containing more than that number are omitted.

The PRINT option controls printed output as in the regression directives, except that some irrelevant settings are omitted. (For example, grid is relevant only to the fitting of nonlinear models.) See VRDISPLAY for more details.

The DENOMINATOR option specifies how the residual is selected for the accumulated analysis of variance By default it is taken from the model with the smallest number of residual degrees of freedom. However, you can set DENOMINATOR=ms to take it from the model with the smallest residual mean square.

The SELECTION option specifies the statistics to be displayed in the summary of analysis as in the regression directives, except that again some irrelevant settings are omitted. See VRDISPLAY for more details.

Options: PRINT, FACTORIAL, DENOMINATOR, SELECTION.
Parameter: TERMS.

Action with RESTRICT

Any restriction applied to vectors used in the REML analysis will apply also to the results from VRADD.

See also

Directives: FIT, REML.
Procedures: VRFIT, VRADD, VRDISPLAY, VRKEEP, VRSETUP, VRSWITCH, VRTRY.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VREGRESS

Performs regression across variates (M.W. Patefield & D. Tandy).

No options

Parameters

`Y` = *pointers*	Pointers each containing a set of y-variates for each of whose units a regression is to be done
`X` = *pointers*	Pointer containing x-variates for each set of y-variates
`SLOPE` = *variates*	Variate to save the estimated slopes from each set of regressions
`INTERCEPT` = *variates*	Variate to save the estimated intercepts from each set of regressions

Description

Given a pointer containing a set of y-variates and another containing a set of x-variates, `VREGRESS` performs a separate regression for the data in each unit of the variates. The pointers are specified using the `Y` and `X` parameters. There must be an equal number of x- and y-variates, and the variates must all be of the same length. The `SLOPE` parameter must supply a variate to receive the regression coefficients, and the `INTERCEPT` parameter can give a variate to save the intercepts. These variates will have the same length as the x- and y-variates.

Options: none. Parameters: `Y`, `X`, `SLOPE`, `INTERCEPT`.

Method

The procedure propagates missing values in any of the x-variates into the appropriate unit of the corresponding y-variate, and vice-versa. The regressions are calculated using matrix operations and variate functions in `CALCULATE`. The vectors of means across the x- and y-variates are subtracted, and then the sums of squares of `X` and the sums of products of `Y` and `X` are calculated across the variates, to obtain the estimated slope coefficients. The estimated intercepts are calculated directly from the slope coefficients and the vectors of means.

The action taken with missing values is the same as would be given by the `FIT` directive. Units with all values of `Y` and `X` missing after propagation will have missing values in both `SLOPE` and `INTERCEPT`. Units with a single non-missing value will have a missing value in `SLOPE` and the corresponding element of `INTERCEPT` equal to the non-missing value of `Y`.

It is considerably faster to use `VREGRESS` than to use `FIT` on each unit after re-arrangement of the data. However, there may be a slight loss of accuracy resulting from single-precision calculations on machines where double-precision is used for the calculations within `FIT`.

Action with `RESTRICT`

All the data variates in `Y` and `X` must be subject to the same restriction (if any). If this is not so a fault (`VA 1 - incompatible restrictions`) will occur during the calculations. If `SLOPE` or `INTERCEPT` is restricted prior to use of `VREGRESS`, the restriction must be the same as that on the data variates. On exit from `VREGRESS`, the computed `SLOPE` and `INTERCEPT` variates will be restricted in the same way as the data variates.

See also
Procedure: VINTERPOLATE.
Functions: VSUMS, VTOTALS, VMEANS, VMEDIANS, VMINIMA, VMAXIMA, VRANGE, VCOVARIANCE, VCORRELATION, VSD, VSEMEANS, VSKEWNESS, VKURTOSIS, VVARIANCES, VNOBSERVATIONS, VNVALUES, VNMV, VPOSITIONS.
Genstat Reference Manual 1 Summary section on: Regression analysis.

VRFIT

Fits terms from a REML fixed model in a Genstat regression (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated); default mode, summ, esti, accu
FACTORIAL = *scalar*	Limit for expansion of terms; default 3
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
SELECTION = *string tokens*	One or two criteria to be printed with the models (%variance, %ss, adjustedr2, r2, dispersion, aic, sic, bic); default %var, aic, sic

Parameter

TERMS = *formula*	Fixed terms to be fitted

Description

VRFIT is one of several procedures designed to improve the process of determining the appropriate fixed terms to include in a REML analysis. (The others are VRADD, VRDROP, VRDISPLAY, VRKEEP, VRSETUP, VRSWITCH and VRTRY.) They do this by a generalized regression analysis, with a weight matrix based on variances estimated from the original REML analysis (with the full fixed model). You can use the mean square of the current model to assess each change as in ordinary regression. However, as this is a weighted regression, the mean square from the full fixed model is one. A convenient alternative might therefore be to use this mean square (of one), to assess the terms with the same measure of random variation as in analysis of variance. (Conversely, if you were to assess the fixed model by changing the fixed model in a sequence of VCOMPONENTS commands, the fixed terms that are not fitted will be included in the random variation. This will then vary from fit to fit, making it difficult to reach a clear and consistent conclusion.) Having used these procedures to decide on the important fixed terms, you can use VPREDICT to form predicted means.

Before fitting the terms, the VRSETUP procedure must be called to make some checks, and initialize the regression by specifying a MODEL command with the necessary weight matrix and a TERMS command with the full fixed model. It also uses the WORKSPACE directive to set up a Genstat workspace structure to store control information and results. VRFIT will call VRSETUP for you, if you have not done so already. The analysis will then be based on the most recent REML analysis. To use an earlier analysis, you should call VRSETUP yourself, setting its SAVE option set to the save structure of the required REML analysis.

The terms to be fitted are specified by the TERMS parameter, in a similar way to the FIT directive. The FACTORIAL option sets a limit (by default 3) on the number of factors and variates in each term. Terms containing more than that number are omitted.

The PRINT option controls printed output as in the regression directives, except that some irrelevant settings are omitted. (For example, grid is relevant only to the fitting of nonlinear models.) See VRDISPLAY for more details.

The DENOMINATOR option specifies how the residual is selected for the accumulated analysis of variance By default it is taken from the model with the smallest number of residual degrees of freedom. However, you can set DENOMINATOR=ms to take it from the model with the smallest residual mean square.

The SELECTION option specifies the statistics to be displayed in the summary of analysis as in the regression directives, except that again some irrelevant settings are omitted. See

VRDISPLAY for more details.

Options: PRINT, FACTORIAL, DENOMINATOR, SELECTION.
Parameter: TERMS.

Method
VRFIT calls the VRSETUP procedure to initialize the regression, if this has not been done already. It then uses the directives FIT and ADD to fit the terms one at a time, storing the results in the workspace together with the denominator degrees of freedom for each term if available from the original REML analysis. (It is the need to use the REML denominator degrees of freedom regression for the terms, instead of the residual degrees of freedom from the regression, that prevents you from using the regression commands directly.) Finally, VRFIT calls VRDISPLAY to print the results.

Action with RESTRICT
Any restriction applied to vectors used in the REML analysis will apply also to the results from VRFIT.

See also
Directives: FIT, REML.
Procedures: VRADD, VRDROP, VRDISPLAY, VRKEEP, VRSETUP, VRSWITCH, VRTRY, VALLSUBSETS, VSCREEN.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRKEEP

Saves output for a REML fixed model fitted in a Genstat regression (R.W. Payne).

Options

FACTORIAL = *scalar*	Limit for expansion of terms; default 3
RESIDUALS = *variate*	Residuals, as specified by the RMETHOD option
FITTEDVALUES = *variate*	Fitted values
RMETHOD = *string token*	Type of residuals to form (simple, standardized); default simp
RDF = *scalar*	Residual degrees of freedom
RSS = *scalar*	Residual sum of squares
ACCUMULATED = *pointer*	Accumulated analysis-of-variance table
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss

Parameters

TERMS = *formula*	Terms whose information is to be saved
ESTIMATES = *table*, *scalar* or *pointer* to *tables* or *scalars*	Estimated regression coefficients for each term
SE = *table*, *scalar* or *pointer* to *tables* or *scalars*	Standard errors of estimated regression coefficients for each term
VCOVARIANCE = *symmetric matrix* or *pointer* to *symmetric matrices*	Variances and covariances between the estimates of each term
NDF = *scalar* or *pointer* to *scalars*	Numerator degrees of freedom for each term
DDF = *scalar* or *pointer* to *scalars*	Denominator degrees of freedom for each term

Description

VRKEEP saves output from models fitted by procedures VRFIT, VRADD, VRDROP, VRSWITCH and VRTRY, which are designed to improve the process of determining the appropriate fixed terms to include in a REML analysis. They do this by a generalized regression analysis, with a weight matrix based on variances estimated from the original REML analysis (with the full fixed model). See VRFIT for more details.

The TERMS parameter specifies terms about which you wish to save information. As in FIT, the FACTORIAL option sets a limit on the number of factors and variates in each term. Any term containing more than that limit is deleted. The subsequent parameters allow you to specify identifiers of data structures to store the various types of information for each of the terms that you have specified. The ESTIMATES parameter saves estimates for each term, in a table if the term involves factors or in a scalar if it involves only variates. Similarly the SE parameter saves standard errors for the estimates. The VCOVARIANCE parameter saves the variances and covariances between the estimates of each term, in a symmetric matrix if the term involves factors or in a scalar if it involves only variates. The NDF and DDF parameters saves the number of numerator and denominator degrees of freedom for the terms, in scalars. If you have a single term, you can supply a table, scalar or symmetric matrix for each of these parameters, as appropriate. However, if you have several terms, you must supply a pointer which will then be set up to contain as many tables, scalars or symmetric matrices as there are terms.

The RESIDUALS and FITTEDVALUES options save the residuals and fitted values, respectively. The RMETHOD option controls the type of residuals that are formed.

The RDF and RSS options save the number of residual degrees of freedom and the residual sum

of squares.

The ACCUMULATED parameter saves the accumulated analysis-of-variance table, as a pointer with elements labelled 'Change', 's.s.', 'n.d.f.', 'm.s.', 'Wald','F', 'Wald pr.', 'd.d.f.' and 'F pr.'. The last two elements will contain missing values if the denominator degrees of freedom of the terms could not be estimated in the original REML analysis. The pointer is defined so that the case of the labels is not significant.

The DENOMINATOR option specifies how the residual is selected for the accumulated analysis of variance By default it is taken from the model with the smallest number of residual degrees of freedom. However, you can set DENOMINATOR=ms to take it from the model with the smallest residual mean square.

Options: FACTORIAL, RESIDUALS, FITTEDVALUES, RMETHOD, RDF, RSS, ACCUMULATED, DENOMINATOR.
Parameters: TERMS, ESTIMATES, SE, VCOVARIANCE, NDF, DDF.

See also
Directives: RKEEP, REML.
Procedures: VRADD, VRDISPLAY, VRDROP, VRFIT, VRSETUP, VRSWITCH, VRTRY.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRMETAMODEL

Forms the random model for a REML meta analysis (R.W. Payne).

Options

RANDOM = *formula structure*	Saves the random model
EXPERIMENTSFACTOR = *factor*	Factor defining which units are in each experiment
TERMS = *formula*	Specifies terms, if any, to be fitted over the whole data set; default * i.e. none

Parameters

EXPERIMENT = *scalars*, *variates* or *texts*	Experiments on which additional random terms are to be fitted
LOCALTERMS = *formula structures*	Random terms that are to be fitted only on the corresponding experiment
SAVEVECTORS = *pointers*	Saves the factors (and/or any variates) defined to represent the local terms on each experiment

Description

In REML meta analyses the designs used in the various experiments need not be identical and, even if they are all the same, the same random model may not be appropriate for every one. REML does allow you to fit different random terms in the different experiments, but their definition can be tedious. For example, if you wanted to include the term Blocks only in experiments 1 and 2 (and with a different variance component in each case), you would need to take two copies of the factor, giving them names (e.g. Blocks1 and Blocks2) that will be recognisable in the output. Then, set Blocks1 to missing except within experiment 1, and Blocks2 to missing except in experiment 2. If you now add Blocks1 + Blocks2 to the overall random model, and set option MVINCLUDE=explanatory in the REML statement, the terms Blocks1 and Blocks2 will each be fitted only in the desired experiment (1 or 2, respectively), and ignored elsewhere. An example is shown in Chapter 2 of the *Guide to REML*.

The process of forming the modified copies of the factors and devising names to label them clearly on the output can be inconvenient. So procedure VRMETAMODEL has been provided to make this clearer and more straightforward. In the output a term like Reps.Blocks, that is to be fitted only e.g. at Rothamsted, will be labelled

```
Reps@Rothamsted.Blocks@Rothamsted
```

The random model is formed automatically, and can be saved in a formula structure by the RANDOM option. The EXPERIMENTSFACTOR option must specify a factor to indicate which units of the data set belong to each experiment, and the TERMS option can specify random terms that are to be fitted over the whole data set.

The EXPERIMENT parameter lists the experiments where additional random terms are to be fitted, using either the levels or the labels of EXPERIMENTSFACTOR. You can specify a variate or a text with several values, if the terms are to be fitted with the same variance components in more than one experiment.

The LOCALTERMS parameter specifies a formula structure for each experiment to define its additional terms. The factors (and any variates) in the additional terms for each experiment are copied, the required missing values are inserted, and the terms are added to the random model.

By default, the modified copies of the factors and variates that are formed to represent the additional random terms will be unnamed, and exist only as part of the RANDOM model. (The labels that appear in the output are attached to the factors by setting the EXTRA parameter in the FACTOR statement or VARIATE statement that defined them inside VRMETAMODEL.) The SAVEVECTORS parameter allows you to supply a pointer for each experiment, to save its factors

(and any variates), so that you use them to refer to the additional random terms e.g. in the VKEEP directive. The elements of each pointer are labelled by the identifiers of the factors or variates in the corresponding local term to simplify their subsequent use.

Options: RANDOM, EXPERIMENTSFACTOR, TERMS.
Parameters: EXPERIMENT, LOCALTERMS, SAVEVECTORS.

See also
Directive: REML, VCOMPONENTS, VRESIDUAL.
Procedure: META, VAMETA.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRPERMTEST

Performs permutation tests for random terms in REML analysis (V.M. Cave).

Options

PRINT = *string tokens*	Controls printed output (summary, monitoring, vdiagnostics); default summ
VPRINT = *string tokens*	Controls the output from the REML analysis of the full and reduced models (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default * i.e. none
PLOT = *string tokens*	What graphs to plot (kerneldensity, histogram); default *
MODELDEFINITION = *pointer*	REML model definition structure, defined using the VFMODEL and VFSTRUCTURE procedures, to specify the full model; no default, must be set
RDROP = *formula*	Random term(s) to drop from the full model; no default, must be set
NTIMES = *scalar*	Number of permutations to make; default 99
NRETRIES = *scalar*	Maximum number of extra permutations to make when some REML analyses fail to converge; default NTIMES
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
WINDOW = *scalar*	Window to use for the graphs; default 3

Parameters

Y = *variates*	Variates to be analysed
STATISTICS = *scalars* or *pointers*	Saves the test statistics
PROBABILITIES = *scalars* or *pointers*	Saves the p-values
TITLE = *text*	Title for the graphs
SAVE = *pointers*	Saves the test statistics and permuted values

Description

VRPERMTEST performs permutation tests to assess whether random terms can be dropped from a linear mixed model, fitted using REML. The procedure implements a pair of permutation tests: one based on the best linear unbiased predictors (BLUP-based), and one based on the residual (or restricted) likelihood ratio test statistic (rLR-based). The rLR-based test enables the simultaneous testing of multiple random terms, whereas the BLUP-based approach can be used only to test the significance of dropping a single random term.

The full model is specified by forming a model-definition structure using the VFMODEL and VFSTRUCTURE procedures. These define the model settings controlled by the VCOMPONENTS and VSTRUCTURE directives, respectively. The VAOPTIONS procedure can be used to control some options of the REML commands used by VRPERMTEST.

The model-definition structure is supplied to VRPERMTEST by the MODELDEFINITION option. The random terms to drop from the full model are defined by a model formula supplied by the RDROP option. If more than one random term is specified, only the rLR-based permutation test is performed.

The Y parameter specifies the variate that is to be modelled. Restrictions and missing values

are not allowed in either the y-variate or the explanatory variates or factors.

Results can be saved using the STATISTICS, PROBABILITIES and SAVE parameters. When a single random term is tested, STATISTICS and PROBABILITIES save pointers that store the rLR-based and BLUP-based test statistics and their p-values, respectively. Alternatively, when more than one random term is tested, STATISTICS and PROBABILITIES save scalars that store the rLR-based test statistic and its p-value, respectively. The SAVE parameter can supply a pointer to store the test statistic and its permuted values. The first element of the pointer, indexed by 'permutedT_rLR', is a variate storing the rLR-based test statistic and its permuted values; the first value in the variate is the test statistic from the original data set. When a single random term is tested, a second element, indexed by 'permutedT_BLUP', stores the BLUP-based test statistic (first value) and its permuted values.

The NTIMES option defines how many random permutations to perform; default 99. The NRETRIES option specifies the maximum number of extra samples to take when some REML analyses fail to converge; the default is to use the same number as specified by NTIMES. The SEED option specifies the seed for the random-number generator, used by RANDOMIZE to make the permutations. The default of zero continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, it initializes the seed automatically. If you use the same (non-zero) seed more than once, you will get the same random numbers, and hence the same permutations.

Printed output is controlled by the PRINT option, with the following settings.

monitoring	prints monitoring information, showing the progress of the analysis.
summary	prints a summary of the test results: first the seed, the number of permutations and percentage of successful permutations, then a table showing the random term(s) tested, test statistic(s) and the corresponding p-value(s). This is the default.
vdiagnostics	prints any error diagnostics from the REML analyses.

The VPRINT option controls output from the REML analyses of the null and alternative models. The settings are the same as those of the PRINT option of the REML directive. By default, nothing is printed.

The PLOT option controls graphical output, with settings:

histogram	to plot a histogram of the permuted test statistics, and
kerneldensity	to produce a kernel density plot of the permuted test statistics.

The WINDOW option defines the window to use for the plots; default 3. By default, nothing is plotted. The TITLE parameter can supply a title for the plots.

Options: PRINT, VPRINT, PLOT, MODELDEFINITION, RDROP, NTIMES, NRETRIES, SEED, WINDOW.

Parameters: Y, STATISTICS, PROBABILITIES, TITLE, SAVE.

Method

VRPERMTEST is based on the methods of Lee & Braun (2012). Two permutation tests are available. The first test, based on the best linear unbiased predictors (BLUP-based), can be used for inference about a single random effect. The second test, based on the restricted likelihood ratio test statistic (rLR-based), can simultaneously test for the presence of multiple random effects. Both methods involve permuting the weighted marginal residuals. The weights, determined by the Cholesky decomposition of the unit-by-unit variance-covariance matrix, ensure that the marginal residuals are exchangeable under the null hypothesis.

The permutation test proceeds as follows:

1 The full model (M_1) and reduced model (M_0), which omits the random terms specified by RDROP, are fitted using REML.
2 The observed test statistic is calculated.
BLUP-based: $T_{BLUP} = \sum_{i=1,N} b_i^2$
where $b_1 \ldots b_N$ are the estimated BLUPs of the single random term being tested.
rLR-based: $T_{rLR} = -2 \log(L_{M0} - L_{M1})$
where L_{M0} and L_{M1} are the restricted likelihoods under the reduced and full models, respectively.
3 The marginal residuals, estimated from the full model, are weighted by $(\mathbf{U}_0')^{-1}$, where $\mathbf{U}_0$ is the Cholesky decomposition of the unit-by-unit variance-covariance matrix from the reduced model.
4 The weighted errors are permuted using RANDOMIZE.
5 The permuted residuals are unweighted, by multiplication by $\mathbf{U}_0'$, and a permuted Y variate (**Y***) is obtained.
6 The full and reduced models are refitted with the permuted Y variate, **Y***, and permuted values of the test statistics (T_{BLUP} and T_{rLR}) are calculated.
7 Steps 4-6 are repeated a maximum of NTIMES + NRETRIES times.
8 The p-value is given by the proportion of test statistics (including the observed test statistic) greater than the observed test statistic.

The kernel density plot is generated by the KERNELDENSITY procedure, using the method of Sheather & Jones (1991), the default number of grid points, and quantiles calculated at 0.025, 0.25, 0.5, 0.75 and 0.975. The permuted test statistics are plotted using red + symbols along the x-axis, and the location of the test statistic is denoted by a blue line. As the observed test statistic contributes to the null distribution, it is included in the calculation of both the kernel density and histogram.

Action with RESTRICT

Restrictions are not allowed.

References

Lee, O.E. & Braun, T.M. (2012). Permutation tests for random effects in linear mixed models. *Biometrics*, **68**, 486-493.

Sheather, S.J. & Jones, M.C. (1991). A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society, Series B*, **53**, 683-690.

See also

Directives: REML, VCOMPONENTS.

Procedures: VBOOTSTRAP, VAOPTIONS, VFLC, VFMODEL, VFSTRUCTURE, VPERMTEST.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRSETUP

Sets up Genstat regression to assess terms from a REML fixed model (R.W. Payne).

Option

SAVE = *REML save structure*	Specifies the analysis whose fixed terms are to be tested; by default this will be the most recent REML

No parameters

Description

VRSETUP sets up Genstat regression to enable procedures VRFIT, VRADD, VRDROP, VRSWITCH, VRTRY, VRDISPLAY and VRKEEP, to be used to assess the terms in a REML fixed model. See VRFIT for details.

By default, VRSETUP takes the most recent REML analysis. However, you can take an earlier analysis, by using the SAVE option of VRTSETUP to specify its save structure (saved using the SAVE parameter of the earlier REML command).

VRSETUP first makes some checks to ensure that the REML analysis was successful, and that it is feasible to use regression. This cannot be used if the REML analysis did not estimate the constant term, or if it included any units with missing explanatory units (i.e. if the original REML command had option CONSTANT = omit, or option MVINCLUDE = explanatory). VRSETUP then obtains the unit-by-unit variance-covariance matrix (using the UVCOVARIANCE option of VKEEP), and inverts it to provide the weight matrix for the regression. It initializes the regression by specifying a MODEL command with the necessary weight matrix, and a TERMS command with the full fixed model. It also uses the WORKSPACE directive to set up a Genstat workspace structure to store control information and results for use by the other procedures.

Option: SAVE.
Parameters: none.

Action with RESTRICT

Any restriction applied to vectors used in the REML analysis will apply also to the results from VRFIT etc.

See also

Directives: FIT, REML.
Procedures: VRADD, VRDROP, VRFIT, VRDISPLAY, VRKEEP, VRSWITCH, VRTRY.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRSWITCH

Adds or drops terms from a REML fixed model in a Genstat regression (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (model, deviance, summary, estimates, correlations, fittedvalues, accumulated); default mode, summ, esti, accu
FACTORIAL = *scalar*	Limit for expansion of terms; default 3
DENOMINATOR = *string token*	Whether to base ratios in accumulated summary on rms from model with smallest residual ss or smallest residual ms (ss, ms); default ss
SELECTION = *string tokens*	One or two criteria to be printed with the models (%variance, %ss, adjustedr2, r2, dispersion, aic, sic, bic); default %var, aic, sic

Parameter

TERMS = *formula*	Fixed terms to be added or dropped

Description

VRSWITCH is one of several procedures designed to improve the process of determining the appropriate fixed terms to include in a REML analysis. (The others are VRFIT, VRADD, VRDISPLAY, VRDROP, VRKEEP, VRSETUP and VRTRY.) They do this by a generalized regression analysis, with a weight matrix based on variances estimated from the original REML analysis (with the full fixed model). See VRFIT for details.

Before fitting any terms, the VRSETUP procedure must be called to make some checks, and initialize the regression by specifying a MODEL command with the necessary weight matrix and a TERMS command with the full fixed model. It also uses the WORKSPACE directive to set up a Genstat workspace structure to store control information and results. However, VRFIT will call VRSETUP for you, if you have not done so already. The analysis will then be based on the most recent REML analysis. To use an earlier analysis, you should call VRSETUP yourself, setting its SAVE option set to the save structure of the required REML analysis.

In principle the VRFIT procedure should also be called before VRSWITCH is used. However, VRSWITCH will call VRFIT with a null model (i.e. only the constant) if VRFIT has not been used already. So you can start investigating the fixed model just by calling VRSWITCH (and VRFIT and VRSETUP will be called for you, automatically).

The TERMS parameter specifies the terms to be added or dropped, in a similar way to the SWITCH directive. Any term that is not already in the current model is added to the model. Conversely, those that are already in the model are dropped. The FACTORIAL option sets a limit (by default 3) on the number of factors and variates in each term. Terms containing more than that number are omitted.

The PRINT option controls printed output as in the regression directives, except that some irrelevant settings are omitted. (For example, grid is relevant only to the fitting of nonlinear models.) See VRDISPLAY for more details.

The DENOMINATOR option specifies how the residual is selected for the accumulated analysis of variance By default it is taken from the model with the smallest number of residual degrees of freedom. However, you can set DENOMINATOR=ms to take it from the model with the smallest residual mean square.

The SELECTION option specifies the statistics to be displayed in the summary of analysis as in the regression directives, except that again some irrelevant settings are omitted. See VRDISPLAY for more details.

Options: PRINT, FACTORIAL, DENOMINATOR, SELECTION.
Parameter: TERMS.

Action with RESTRICT
Any restriction applied to vectors used in the REML analysis will apply also to the results from VRSWITCH.

See also
Directives: FIT, REML.
Procedures: VRFIT, VRADD, VRDISPLAY, VRDROP, VRKEEP, VRSETUP, VRTRY.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VRTRY

Tries the effect of adding and dropping individual terms from a REML fixed model in a Genstat regression (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (changes); default chan
FACTORIAL = *scalar*	Limit for expansion of terms; default 3
CHANGES = *pointer*	Saves details of the changes

Parameter

TERMS = *formula*	Fixed terms to be added or dropped

Description

VRTRY is one of several procedures designed to improve the process of determining the appropriate fixed terms to include in a REML analysis. (The others are VRFIT, VRADD, VRDISPLAY, VRDROP, VRKEEP, VRSETUP and VRSWITCH.) They do this by a generalized regression analysis, with a weight matrix based on variances estimated from the original REML analysis (with the full fixed model). See VRFIT for details.

Before fitting any terms, the VRSETUP procedure must be called to make some checks, and initialize the regression by specifying a MODEL command with the necessary weight matrix and a TERMS command with the full fixed model. It also uses the WORKSPACE directive to set up a Genstat workspace structure to store control information and results. However, VRFIT will call VRSETUP for you, if you have not done so already. The analysis will then be based on the most recent REML analysis. To use an earlier analysis, you should call VRSETUP yourself, setting its SAVE option set to the save structure of the required REML analysis.

In principle the VRFIT procedure should also be called before VRTRY is used. However, VRTRY will call VRFIT with a null model (i.e. only the constant) if VRFIT has not been used already. So you can start investigating the fixed model just by calling VRTRY (and VRFIT and VRSETUP will be called for you, automatically).

The TERMS parameter specifies the terms to be added or dropped, in a similar way to the TRY directive. Any term that is not already in the current model is added to the model. The effect of the change is recorded, and then the term is taken back out of the model. Conversely, any term that is already in the current model is dropped from the model. Again, the effect of the change is recorded, before the term is added back into the model. The FACTORIAL option sets a limit (by default 3) on the number of factors and variates in each term. Terms containing more than that number are omitted.

By default VRTRY prints a table showing the effect of adding and dropping the various terms. However, you can suppress that by setting option PRINT=*.

You can use the CHANGES option to save information about the changes in a pointer. The first elements of the pointer, labelled 'Change', 's.s.', 'n.d.f.' and 'd.d.f.', save the corresponding columns of the table in a text ('Change') and three variates. The final element, 'term', is a pointer storing a model formula for the term associated with each change. The final line of the table (Residual of initial model) is not included. The pointer is defined so that the case of the labels is not significant.

Options: PRINT, FACTORIAL, CHANGES.
Parameter: TERMS.

Action with RESTRICT

Any restriction applied to vectors used in the REML analysis will apply also to the results from VRTRY.

See also
Directives: FIT, REML.
Procedures: VRFIT, VRADD, VRDISPLAY, VRDROP, VRKEEP, VRSETUP, VRTRY.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VSAMPLESIZE

Estimates the replication to detect a fixed term or contrast in a REML analysis, using parametric bootstrap (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (power, replication, monitoring); default powe, repl, moni
TERM = *formula*	Fixed term to be assessed in the analysis
REPLICATES = *factor*	Factor identifying the replication in the design
TRYREPLICATION = *variate*	Replication values to try first; default !(2,4)
MAXREPLICATION = *scalar*	Maximum feasible replication; default * i.e. not defined
FIXED = *formula*	Fixed terms in the analysis; if unset, determined automatically from the most recent VCOMPONENTS
RANDOM = *formula*	Random terms in the analysis; if unset, determined automatically from the most recent VCOMPONENTS
COMPONENTS = *variate* or *scalar*	Variate of variance components of the random terms; must be set
FACTORIAL = *scalar*	Limit on the number of factors or variates in fixed terms; default 3
PROBABILITY = *scalar*	Significance level at which the term is required to be detected (assuming a one-sided test); default 0.05
POWER = *scalar*	The required power (i.e. probability of detection) of the test; default 0.9
TMETHOD = *string token*	Type of test to be made (fratio, wald, twosided, lessthan, greaterthan, equivalence, noninferiority; default frat
XCONTRASTS = *variate*	X-variate defining a contrast to be detected
CONTRASTTYPE = *string token*	Type of contrast (regression, comparison) default rege
CRITICALVALUE = *scalar*	Supplies a critical value for the test statistic
NBOOT = *scalar* or *variate*	Number of bootstrap samples to analyse, in a variate with 2 values if there is to be preliminary search, otherwise in a scalar; default 1000
NRETRIES = *scalar* or *variate*	Maximum number of extra samples to take when some REML analyses fail to converge, in a variate with 2 values if there is to be preliminary search, otherwise in a scalar; default NBOOT
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
METHOD = *string token*	Indicates whether to use the standard Fisher-scoring algorithm or the new AI algorithm with sparse matrix methods (Fisher, AI); default AI
MAXCYCLE = *scalar*	Sets a limit on the number of iterations in the REML analyses; default 30
FMETHOD = *string token*	Controls whether and how to calculate F statistics for fixed terms (automatic, none, algebraic, numerical); default auto
WMETHOD = *string token*	Controls which Wald statistics are saved (add, drop); default add
WORKSPACE = *scalar*	Number of blocks of internal memory to be set up for

use by the REML algorithm

Parameters

RESPONSE = *scalars* or *tables*	Specifies the response to be detected
NREPLICATES = *scalars*	Number of replicates required to detect RESPONSE

Description

When designing an experiment, it is usually possible to vary the replication of the treatments. For example, in a resolvable design, you may be able to include additional (duplicate) replicates. Alternatively, in some situations, it may be possible to improve precision by taking replicate measurements within the basic experimental units: for example, increasing the number of independent samples taken from a field plot. VSAMPLESIZE estimates the replication required to detect a specified response or contrast in a REML analysis.

The FIXED option defines the fixed model for the REML analysis. The RANDOM option specifies the random model. This may also contain the residual term, but that is not essential. If it is not present, the residual is added as the final model term. The COMPONENTS option specifies the variance components for the random terms, including the residual variance (at the end, if this had to be added this to the RANDOM formula). If either FIXED or RANDOM is not specified, their defaults are taken from the most recent VCOMPONENTS statement. The FACTORIAL option sets a limit (default 3) on the number of factors or variates in a fixed term; any containing more than that number are deleted. VSAMPLESIZE cannot be used for designs whose analysis to include covariance structures (specified by VSTRUCTURE).

The REPLICATES option must be set to the factor in the random model whose number of levels is to be increased or decreased to change the replication of the treatments. The factors in the fixed and random models must be defined to contain the values for a single replicate of the design. You can set the TRYREPLICATION option to a variate containing the number of replicates to try first. There must be at least two of these. The default is a variate containing the numbers 2 and 4. The MAXREPLICATION option can specify the maximum feasible number of replicates. The NREPLICATES parameter can save the estimate of the number of replicates that is required.

The fixed term to be tested is specified using the TERM option, and the response to be detected is specified by the RESPONSE parameter. This can supply a scalar to specify the maximum difference between the effects of the term, or it can supply a table, to specify the anticipated effects themselves. As an alternative to detecting a difference between its effects, you can ask to detect a contrast. RESPONSE must then supply a scalar, and TERM must be a main effect (that is, it must involve just one factor). The XCONTRASTS option must specify a variate or table containing the coefficients defining the contrast, and the CONTRASTTYPE option indicates whether this is a regression contrast (as specified e.g. by the REG function in ANOVA) or a comparison (as specified e.g. by the COMPARISON function in ANOVA).

The TMETHOD option specifies the type of test that is to be used to assess the term, with the following settings.

fratio	assumes that the term will be tested using its F ratio.
wald	assumes that the term will be tested by a Wald test.
twosided	assumes a two-sided test to assess whether a contrast of the term differs from zero (default).
lessthan	assumes a one-sided test to assess whether a contrast of the term is less than zero.
greaterthan	assumes a one-sided test to assess whether a contrast of the term is greater than zero.
noninferiority	assumes a test to check that a contrast of the term is not significantly less then zero. (See Method for more details.)

`equivalence`	assumes a one-sided test to check that a contrast of the term does not differ significantly from zero; see Method for more details.

The settings `fratio` and `wald` are not appropriate for contrasts. The default is `twosided` when there are contrasts, and `fratio` otherwise. Note: the specified response must be negative when `TMETHOD` is set to `lessthan` or `noninferiority`.

`VSAMPLESIZE` uses the `VPOWER` procedure to estimate of the power with which the response will be detected for each number of replicates that is tried. `VPOWER` performs a parametric bootstrap, in which random data variates are generated and analysed by `REML` ro see how often the term's response is significant.

The `MAXCYCLE` option sets a limit on the number of iterations in the `REML` analyses (default 30). The `METHOD` option controls whether `REML` uses the Fisher-scoring algorithm, or the AI algorithm with sparse matrix methods (the default). The `WMETHOD` option controls whether the Wald and F statistics are obtained from the table where terms are added sequentially (the default), or from the table where suitable terms are dropped from the full fixed model. Note that, if you use the table where terms are dropped, the `TERM` must not be not marginal to any other term in the fixed model: for example, the main effect `A` cannot be tested if the model contains an interaction, such as `A.B`. The `FMETHOD` option controls how to estimate the denominator degrees of freedom for the F tests. (This is relevant if `TMETHOD=fratio`, or if tests for fixed effects are being printed in the `REML` analyses of the bootstrap samples.) The `WORKSPACE` option specifies the number of blocks of internal memory to be set up for use by the `REML` algorithm.

The `NBOOT` option specifies the number of bootstrap samples to take. The `NRETRIES` option specifies the maximum number of extra samples to take when some `REML` analyses fail to converge. These can be either a scalar, or a variate with one or two values. If two values are supplied, the first is used during an initial search to find a replication value to provide at least enough power. The second is then used for a more precise search, The default for `NBOOT` is to the single value 1000. The default for `NRETRIES` is to use the same number as specified by `NBOOT`. The `SEED` option supplies the seed for the random number generator used to form the samples; default 0 continues from the previous generation or (if none) initializes the seed automatically.

The `PROBABILITY` option specifies the significance level to be used in the test; the default is 0.05, i.e. 5%. The `CRITICALVALUE` option can supply the critical value to be used in the test. (The `VCRITICAL` procedure can be used to obtain this, with a similar parametric bootstrap process to that used by `VPOWER`.). Note: the specified critical value must be negative when `TMETHOD` is set to `lessthan` or `noninferiority`. If `CRITICALVALUE` is not set, the critical value is obtained in the conventional way, using an F, chi-square or t-distribution, according to the type of test.

The `PRINT` option controls the printed output, with settings:

`power`	prints a table giving the estimated power for the numbers of replicates that have been tried in the second phase of the search;
`replication`	prints the required replication; and
`monitoring`	prints monitoring information showing the numbers of replicates and corresponding estimated powers obtained during the search.

By default all are printed.

Options: `PRINT`, `TERM`, `REPLICATES`, `TRYREPLICATION`, `MAXREPLICATION`, `FIXED`, `RANDOM`, `COMPONENTS`, `FACTORIAL`, `PROBABILITY`, `POWER`, `TMETHOD`, `XCONTRASTS`, `CONTRASTTYPE`, `CRITICALVALUE`, `NBOOT`, `NRETRIES`, `SEED`, `METHOD`, `MAXCYCLE`, `FMETHOD`, `WMETHOD`, `WORKSPACE`.

Parameters: RESPONSE, NREPLICATES.

Method

The power is estimated for each number of replicates in the search, using the VPOWER procedure. This sees how frequently the relevant test would be significant in the analyses of a set of bootstrap samples. The variance-covariance matrix required to generate the samples is formed by the VUVCOVARIANCE procedure.

With an equivalence test, you define a threshold h below which two treatments can be assumed to be equivalent. The contrast c would be the difference between the treatments, and the null hypothesis that the treatments are not equivalent is that either

$$c \leq -t$$

or

$$c \geq t$$

with the alternative hypothesis that they are equivalent, i.e.

$$-t < c < \mathrm{t}$$

This defines an *intersection-union* test, in which each component of the null hypothesis must be rejected separately. This implies performing two one-sided t-tests (this is known as a *TOST* procedure). If the significance level for the full test is to be α, each t-test must have significance level α (see Berger & Hsu 1996).

With a non-inferiority test, you again define the threshold t for the effect of the new treatment to be inferior to the standard treatment, and a contrast representing the effect of the new test minus the effect of the standard treatment. The null hypothesis is

$$-c \geq t$$

which represents a one-sided "less-than" t-test.

Reference

Berger, M.L. & Hsu, J.C. (1996). Bioequivalence trials, intersection-union tests and equivalence confidence sets. *Statistical Science*, **11**, 283-319.

See also

Directive: REML.

Procedures: ASAMPLESIZE, VPOWER, VCRITICAL, VUVCOVARIANCE.

Genstat Reference Manual 1 Summary sections on: REML analysis of linear mixed models, Design of experiments.

VSCREEN

Performs screening tests for fixed terms in a REML analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (ftests, waldtests); default ftes, wald
EXCLUDEHIGHER = *string token*	Whether to exclude higher-order interactions in the conditional models (yes, no); default no
FORCED = *formula*	Terms that must always be included in the model (no tests on these terms); default *
FSAVE = *pointer*	Saves the F tests
WSAVE = *pointer*	Saves the Wald tests
SAVE = *REML save structure*	Specifies the analysis whose fixed terms are to be tested; by default this will be the most recent REML

No parameters

Description

VSCREEN calculates marginal and conditional tests for fixed terms in a REML analysis. By default, these are from the recent REML analysis. However, you can take an earlier analysis, by using the SAVE option of VSCREEN to specify its save structure (saved using the SAVE parameter of the earlier REML command).

In the marginal test, the term is added to the simplest possible model. For example, the main effect of A would be added to the null model, and the interaction A.B would be added to a model containing only the main effects A and B.

In the conditional test, the term is added to the most complex possible model that contains no terms involving the term to be tested. For example, interaction A.B would be added to the model containing all terms except those involving A.B (such as the interaction A.B.C). By default, the most complex model includes terms with more factors or variates than the term being tested. For example, the interaction C.D.E would be included when testing A.B. You can exclude these higher-order terms by setting option EXCLUDEHIGHER=yes (and VSCREEN will print a message to remind you that this has been done).

You can specify terms that should always be included in the model by using the FORCED option. These terms are fitted first, and are not tested.

The PRINT option controls printed output, with the following settings.

ftests	presents F statistics for the terms. If denominator degrees of freedom (ddf) are available from the earlier REML analysis, probabilities are also given. Note, however, that these ddf are correct only for models that correspond to those in the sequential Wald table in the REML analysis. They should be acceptable for the other models, but you should be cautious when probabilities are close to critical values.
waldtests	presents Wald statistics for the terms. These suffer from the usual biases of Wald tests in REML analyses, and so should again be used with caution.

You can save the results of the F tests and the Wald tests, in pointers, using the FSAVE and WSAVE options, respectively. The elements of the pointers are labelled by the headers of the columns used in the printed output.

An advantage of using VSCREEN to assess the fixed model, rather than running a succession of REML analyses with different fixed models, is that the fixed terms are assessed against

identical estimates of the random variation (as in an analysis of variance). When terms are dropped from (or added to) the fixed model in a REML analysis, the random variation will change. For example, it will increase if a term with a Wald statistics greater than its number of degrees of freedom is dropped. It may therefore be difficult to reach consistent decisions about which fixed terms are genuinely required.

Once you have used VSCREEN to decide which terms to keep in the fixed model, you can use only those terms for prediction, by specifying them in the MODEL option of VPREDICT.

Options: PRINT, EXCLUDEHIGHER, FORCED, FSAVE, WSAVE, SAVE.
Parameters: none.

Method
VSCREEN defines a weighted regression, with weight matrix given by the inverse of the unit-by-unit variance-covariance matrix (obtained using the UVCOVARIANCE option of VKEEP.) It then calls the RSCREEN procedure to calculate the tests.

Action with RESTRICT
Any restriction applied to vectors used in the REML analysis will apply also to the results from VSCREEN.

See also
Directive: REML.
Procedures: ASCREEN, RSCREEN, VALLSUBSETS, VRFIT.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VSOM

Analyses a simple REML variance components model for outliers using a variance shift outlier model (S.J. Welham, F.N. Gumedze & D.B. Baird).

Options

PRINT = *string tokens*	Specifies the output to be produced (fdr, outliers); default fdr, outl
VPRINT = *string tokens*	Controls the output from the REML analysis of the baseline model (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default mode, comp, Wald, cova
PLOT = *string tokens*	Controls which plots are produced (indexplots, residual); default inde, resi
INDEXPLOT = *string tokens*	Selects the index plots to produce (omega, sigma2, tsquared, lrt, method, all); default meth
TERM = *formula*	Random term to scan for outliers; default is the residual term
METHOD = *string token*	Method for calculating the statistics used to indicate an outlier (full, partial, t); default t
THRMETHOD = *string token*	Method for obtaining the threshold statistics (approximate, bootstrap); default appr for METHOD=full and boot otherwise
NBOOT = *scalar*	Number of bootstrap samples to take to form the threshold statistics; default 99 for METHOD=full and 499 otherwise
FIXED = *formula*	Fixed model terms
RANDOM = *formula*	Random model terms
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
VCONSTRAINTS = *string token*	How to constrain the variance components and the residual variance (none, positive, fixrelative, fixabsolute); default posi
INITIAL = *variate*	Initial values for the variance components; default 1
SEED = *scalar*	Seed for random number generation; default 0 continues an existing sequence or, if none, selects a seed automatically
SAVEITEMS = *string tokens*	Selects the items to save (residuals, omega, sigma2, gamma, tsquared, lrt, fdr, approxthresholds, thresholdstats, outliers, method, all); default resi, omeg, sigm, meth, fdr, outl

Parameters

Y = *variates*	Response variates
TITLE = *texts*	Specifies the title or titles to use for the plots
SAVE = *pointer*s	Saves information from the analysis of each y-variate

Description

VSOM uses a mixed-model analysis with a variance shift outlier model (VSOM) to search for potential outliers. By default, the VSOM is used to assess the residuals. However, you can set the TERM option to a random term in the analysis, to assess its effects: i.e. to see whether any of the groups of observations defined by the random term seem to be aberrant. The model defines an extra component of variation for each unit (an individual or a group), in turn, and estimates the extra variance associated with it. The METHOD option specifies how the extra variance is estimated, with the following settings.

full	refits the full model with the added variance term for each unit; this can be very time-consuming.
partial	approximates the change in likelihood by a partial likelihood, where the baseline model parameters are held fixed, and only the extra variance component for each unit is estimated; this is much faster than re-estimating the full model.
t	uses the squared *t*-statistics (i.e. squared standardized residuals) to approximate the change in likelihood (default); this is the fastest approach.

To assess whether a unit is outside its expected distribution, thresholds are calculated at various levels of significance. The THRMETHOD option specifies the method to use:

approximate	uses the asymptotic distribution to calculate the thresholds; and
bootstrap	uses parametric bootstrap samples, with the variance components in the baseline model, to calculate the thresholds from the percentiles of the order statistics.

Each bootstrap sample is formed by taking the sum of the fitted fixed effects from the baseline model, together with simulated effects for the random terms in the model. Each random effect is simulated by Normal random numbers, with a mean of zero and the variance that was estimated for that term in the baseline model. The NBOOT option defines how many random samples to perform; the default is 99 for METHOD=full, and 499 otherwise. The SEED option specifies the seed for the random number generator, used by the GRNORMAL function to make the bootstrap samples. The default of zero continues the sequence of random numbers from a previous generation or, if this is the first use of the generator in this run of Genstat, it initializes the seed automatically from the computer clock. If you repeat the analysis with the same (non-zero) seed, you will get the same random numbers, and hence the same results.

The FIXED and RANDOM options specify the fixed and random terms to be fitted in the analysis, and the FACTORIAL option sets a limit on the number of factors and variates allowed in each fixed term. If neither FIXED nor RANDOM is specified, their settings are taken from the most recent VCOMPONENTS command. Its FACTORIAL setting is also taken if VCOMPONENTS is providing the fixed model. A fault is given if neither a fixed nor a random model is supplied. Note that the analysis cannot handle covariance models (which would be specified by the VSTRUCTURE directive). The VCONSTRAINTS option specifies constraints on the variance components, using the same settings as the CONSTRAINTS parameter of VCOMPONENTS. The CONSTANT option allows you to omit the constant.

Printed output is controlled by the PRINT option, with the following settings:

outliers	prints a summary of the potential outliers, as measured against the threshold statistics, at various levels of significance; and
fdr	prints the estimated false discovery rates for the potential outliers.

The false discovery rates (FDR) are estimated from the distribution of p-values calculated with

the *t*-statistics from the asymptotic model. This uses the FDRMIXTURE procedure, or else the FDRBONFERRONI procedure if that fails. The FDR estimates the probability that the outlier is generated by noise. If this is small, it is likely that the outlier is genuine. However, if it is larger than 0.5, there is more chance that it was generated by noise. The FDR probabilities do not allow for correlations between the estimates. So, if there are only 2-3 replicates of the fixed terms, these may be too small, and should be interpreted with caution.

The VPRINT option controls the output from the REML analysis of the baseline model (as specified by the FIXED and RANDOM options). This has the same settings and default as the PRINT option of REML.

Graphical output is controlled by the PLOT option, with the following settings.

residual	when TERM is set, the DRESIDUALS procedure is used to plot histograms and Normal plots of the specified random effects; when TERM is not set, DRESIDUALS is used to plot histograms and Normal plots of the residuals together with a plot of the residuals against the fitted values.
indexplots	plots the statistics, selected by the INDEXPLOT option, against their index (i.e. their position in the y-variate).

For residual and indexplots, points are plotted in red if they are greater than their 5% bootstrap threshold, and in purple or green if greater than the 1% or 5% asymptotic thresholds respectively. The index plot also displays reference lines for the order statistics (OS 1, OS 2...) when THRMETHOD=bootstrap, or the 5%, 1% and 0.1% and 0.01% asymptotic thresholds when THRMETHOD=approximate.

The plots that are produced as components of the index plot can be controlled by the INDEXPLOT option, with the following settings:

omega	variance shift as a ratio to the residual variance,
sigma2	estimated residual variance under VSOM,
tsquared	squared *t*-statistic,
lrt	likelihood ratio test,
method	the statistic associated with the setting of the METHOD option, i.e. lrt for full or partial, and tsquared for t (default), and
all	all the statistics.

The Y parameter specifies the response variate. The TITLE parameter can supply a text, with either one or three values, to label the graphs. If the text has a single value, this is used to prefix the standard descriptions for the three graphs. If it has three values, these give (in full) the titles for the comparison, indexplots, residual plots, respectively.

The SAVE parameter can save a pointer containing variates, storing the statistics calculated for each group or individual. The labels of the pointer, and the corresponding statistics, are as follows:

'residuals'	the standardized residuals,
'omega'	the variance shift as a ratio to the residual variance,
'sigma2'	the estimated residual variance under VSOM,
'gamma'	the estimated variance component for TERM under VSOM,
'tsquared'	the squared *t*-statistic,
'LRT'	the partial likelihood ratio test if THRMETHOD=partial or the full likelihood ratio test otherwise,
'method'	the statistic associated with the setting of the METHOD option (lrt for full or partial, and tsquared for t),
'FDR'	the false discovery rate base on the *t*-statistics,
'approxthresholds'	the approximate thresholds used to indicate significant departures,

`'thresholdstats'`	the 95 percentiles of the order statistics from the bootstrap samples in decreasing order, and
`'outliers'`	the unit numbers of outliers above the thresholds.

The `SAVEITEMS` option controls which of the above items are saved.

Options: `PRINT`, `VPRINT`, `PLOT`, `INDEXPLOT`, `TERM`, `METHOD`, `THRMETHOD`, `NBOOT`, `FIXED`, `RANDOM`, `CONSTANT`, `FACTORIAL`, `VCONSTRAINTS`, `INITIAL`, `SEED`, `SAVEITEMS`.
Parameters: `Y`, `TITLE`, `SAVE`.

Method
`VSOM` uses the method of Gumedze *et al.* (2010).

Action with `RESTRICT`
The `Y` parameter can be restricted. All output estimates will then be based only on the unrestricted units.

Reference
Gumedze, F.N., Welham, S.J., Gogel, B.J. & Thompson, R. (2010). A variance shift model for detection of outliers in the linear mixed model. *Computational Statistics and Data Analysis*, **54**, 2128-2144.

See also
Directives: `REML`, `VCOMPONENTS`, `VSTRUCTURE`.
Procedure: `VCHECK`, `VRCHECK`, `VPLOT`, `VDFIELDRESIDUALS`, `VFRESIDUALS`, `DRESIDUALS`. `FDRBONFERRONI`, `FDRMIXTURE`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VSPECTRALCHECK

Forms the spectral components from the canonical components of a multitiered design, and constrains any negative spectral components to zero (C.J. Brien).

Options

PRINT = *string tokens*	Controls printed output (relationshipsmatrix, canonicalcomponentestimates, spectralcomponentestimates, nconstrainedcomponents, all); default spec
VPRINT = *string tokens*	Controls the output from the final REML refit (model, components, effects, means, stratumvariances, monitoring, vcovariance, deviance, Waldtests, missingvalues, covariancemodels); default * i.e. none
INITIALMETHOD = *string token*	Whether to use the estimates from the unconstrained fit as initial values in constrained fits or the default REML initial values (remldefault, unconstrainedanalysis); default unco
MAXCYCLE = *scalar*	Sets a limit on the number of iterations in the REML analyses; default 30
TOLERANCE = *scalar*	Tolerance for zero values; default 10^{-10}
DPRINT = *string tokens*	Controls output of diagnostic information (spectralcomponents, canonicalcomponents, relationshipmatrix, all); default * i.e. none

Parameters

Y = *variates*	Response variates
CORRESPONDENCE = *matrices*	Upper-triangular matrix giving the spectral components in terms of the canonical components
SPECTRALESTIMATES = *variates*	Saves estimates of the spectral components
CANONICALESTIMATES = *variates*	Saves estimates of the canonical components
NCONSTRAINEDCOMPONENTS = *scalars*	Saves the number of spectral components constrained to zero, returns a missing value if some components could not be constrained
EXIT = *scalars*	Exit status of the final REML refit
SAVE = *REML save structures*	Supplies the save structure from the prior analysis of each Y variate; this need not be set, if that was the most recent REML analysis

Description

Randomization-based models, as described by Brien & Bailey (2006) and Bailey & Brien (2013), include the constraint that the spectral components are non-negative, even if the canonical components are allowed to be negative. While the estimates of the spectral components for two-tiered experiments are guaranteed to be non-negative, this not the case for multitiered experiments. VSPECTRALCHECK forms estimates of the spectral components from the canonical components, or unconstrained variance components, that are estimated from fitting a mixed model using the REML directive. It then checks for negative spectral components and, if any are found, imposes relationships between the canonical components so that the spectral components are constrained to be zero.

VSPECTRALCHECK expects that a mixed model has been fitted using the VCOMPONENTS and

REML directives only. It checks that the random model contains only gammas and σ^2, and that there are no spline models. In the random model (specified by the RANDOM parameter of VCOMPONENTS), the terms must be ordered so that, for each term, all the terms to which it is marginal follow it. All canonical components should be specified as unconstrained in the preceding REML analysis (this being the default for the VCOMPONENTS directive).

If VSPECTRALCHECK detects a negative spectral component, it redefines the random model, specifying a matrix of constraints using the RELATIONSHIP parameter of VCOMPONENTS. It then refits the model using REML. Because relationships are to be imposed between the canonical components, the standard Fisher-scoring algorithm (option METHOD=fisher in REML) must be used in the refits. The new estimates for the canonical components are extracted after the refit, and these are used to form new estimates of the spectral components. This process continues until all the spectral components are non-negative.

The Y parameter specifies the variate that was analysed by the preceding REML command. The SAVE parameter can supply the corresponding REML save structure; if this is not set, it is assumed that the y-variate is the one analysed in the most recent REML analysis. A warning is given if the Y variate seems to be different from that in the SAVE structure.

The CORRESPONDENCE parameter specifies a matrix giving coefficients of equations specifying the spectral components in terms of the canonical components. It must be a square, upper triangular matrix with rows corresponding to spectral components, and columns to canonical components. The rows and columns are considered to be in the same order as terms in the random model specified previously, by the VCOMPONENTS directive. The upper triangular form implies that the terms in the random model must be ordered, so that each term occurs before any terms to which it is marginal. In particular, the unit term will be in the last row and column of the matrix. The element (i, j) of this matrix is non-zero if $j \geq i$, and the term for row i is marginal to or equal to the term in column j; in this case, it is equal to the number of replicates of a combination of the levels of the factors in the term in column j (see Bailey & Brien 2013, Equation 5).

The SPECTRALESTIMATES and CANONICALESTIMATES parameters save the constrained estimates of the spectral and canonical components, respectively, in variates. The NOCONSTRAINEDCOMPONENTS parameter saves the number of constrained spectral components, in a scalar. The EXIT parameter can specify a scalar to save the exit status of the final REML fit.

Printed output is controlled by the PRINT option with settings:

relationshipsmatrix	to print the matrix of relationships imposed on the canonical components in the REML refits,
canonicalcomponentestimates	to print the estimates of the canonical components under the imposed relationships,
spectralcomponentestimates	to print the estimates of the spectral components without and, if applicable, also with the constraints imposed,
noconstrainedcomponents	to print the number of constrained components, with missing values indicating that a constraint could not be imposed, and
all	to print all of the above.

You can set the VPRINT option to print information from the final REML refit. This operates in the same way as the PRINT option of REML, except that the default is to print nothing. There is also a DPRINT option to print diagnostic information.

The INITIALMETHOD option control how the initial values are calculated for the REML refits. By default, the estimates from the unconstrained fit are used as initial values for the refits. Alternatively, you can set INITIALMETHOD=remldefault, to get REML to form the initial values automatically, in the usual way.

The MAXCYCLE option sets a limit on the number of iterations (default 30). The TOLERANCE option specifies the tolerance for zero. This is used do determine whether a component is small enough to be considered zero, and in the checking of the Y variate against that in the SAVE structure.

Options: PRINT, VPRINT, INITIALMETHOD, MAXCYCLE, TOLERANCE, DPRINT.
Parameters: Y, CORRESPONDENCE, SPECTRALESTIMATES, CANONICALESTIMATES, NCONSTRAINEDCOMPONENTS, EXIT, SAVE.

Method
Estimates of the canonical components are obtained from a prior REML analysis, and the estimates of the spectral components are obtained using the CORRESPONDENCE matrix. If a spectral component is negative, then relationships between the canonical components, determined from the row in the CORRESPONDENCE matrix for the spectral component, are imposed in a refit of the mixed model by the REML directive. It is possible that some random terms may be removed from the mixed model. After VSPECTRALCHECK has been run, the latest REML analysis will be the one that VSPECTRALCHECK has performed to constrain the components. So, for example, VDISPLAY can be used to display additional information, and VKEEP can be used to save information, in the usual way.

References
Bailey, R. A. & Brien C. J. (2013). Randomization-based models for multitiered experiments. I. A chain of randomizations. arXiv preprint arXiv:1310.4132: 30.
Brien, C.J. (2015). Randomization inference for randomizations in a chain. Submitted for publication.
Brien, C.J. & Bailey, R.A. (2006). Multiple randomizations. *Journal of the Royal Statistical Society, Series B*, **68**, 571-609.
Brien, C.J. & Payne, R.W. (1999). Tiers, structure formulae and the analysis of complicated experiments. *The Statistician*, **48**, 41-52.

See also
Procedure: AMTIER.
Directives: REML, VCOMPONENTS.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VSPREADSHEET

Saves results from a REML analysis in a spreadsheet (R.W. Payne).

Options

COMPONENTS = *variate*	Variate to contain the variance components; default components
MEANS = *pointer*	Pointer to tables to contain the means; default means
SEDMEANS = *pointer*	Pointer to matrices to contain the standard errors of differences of the means; default sedmeans
VARMEANS = *pointer*	Pointer to matrices to contain the variance-covariance matrices of the means; default varmeans
EFFECTS = *pointer*	Pointer to tables to contain the effects; default effects
SEDEFFECTS = *pointer*	Pointer to matrices to contain the standard errors of differences of the effects; default sedeffects
VAREFFECTS = *pointer*	Pointer to matrices to contain the variance-covariance matrices of the effects; default vareffects
REPLICATIONS = *pointer*	Pointer to tables of replications; default replication
WALDTABLE = *pointer*	Pointer to a text and variates containing the information in the table of tests for fixed effects; default waldtable
PTERMS = *formula*	Terms (fixed or random) for which effects or means are to be saved; default * implies all the fixed terms
FMETHOD = *string token*	Controls whether and how to calculate F-statistics for fixed terms (automatic, none, algebraic, numerical); default auto
SPREADSHEET = *string tokens*	What to include in the spreadsheet (components, waldtable, effects, sedeffects, vareffects, means, sedmeans, varmeans, replications); default comp, wald, mean, sedm, repl
OUTFILENAME = *texts*	Name of Genstat workbook file (.gwb) or Excel (.xls or .xlsx) file to create
SAVE = *REML save structure*	Specifies which REML analysis to save; default * i.e. most recent one

No parameters

Description

VSPREADSHEET puts results from a REML analysis into a spreadsheet. By default the results are from the most recent REML, but you use the SAVE option to specify the save structure from some other analysis.

The SPREADSHEET option specifies which pages of the spreadsheet to form, with settings:

components	variance components,
waldtable	tests for fixed effects,
effects	tables of effects,
sedeffects	standard errors of differences of effects,
vareffects	variance-covariance matrices of effects,
means	tables of means,
sedmeans	standard errors of differences of means,
varmeans	variance-covariance matrices of means,
replications	replication tables.

(Note: this includes only the information readily assembled from VKEEP. So, for example, parameters of correlation models are not available.) By default, SPREADSHEET = comp, wald,

mean, sedm, repl.

To help avoid clashes between the columns of the spreadsheets if you want to save results from more than one analysis, the parameters COMPONENTS, WALDTABLE, EFFECTS, SEDEFFECTS, VAREFFECTS, MEANS, SEDMEANS, VARMEANS and REPLICATIONS allow you to specify identifiers for the columns (or sets of columns) that will store the corresponding results in the current spreadsheet.

You can save the data in either a Genstat workbook (.gwb) or an Excel spreadsheet (.xls or .xlsx), by setting the OUTFILENAME option to the name of the file to create. If the name is specified without a suffix, '.gwb' is added (so that a Genstat workbook is saved). If OUTFILENAME is not specified, the data are put into a spreadsheet opened inside Genstat.

Options: COMPONENTS, MEANS, SEDMEANS, VARMEANS, EFFECTS, SEDEFFECTS, VAREFFECTS, REPLICATIONS, WALDTABLE, PTERMS, FMETHOD, SPREADSHEET, OUTFILENAME, SAVE.
Parameters: none.

Action with RESTRICT

If the Y variate is restricted, that restriction will carry over into the fitted-values spreadsheet.

See also

Directive: SPLOAD.
Procedures: ASPREADSHEET, AUSPREADSHEET, RSPREADSHEET, FSPREADSHEET.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

†VSUMMARY

Summarizes a variate, with classifying factors, into a data matrix of variates and factors (D.B. Baird).

Options

PRINT = *string token*	What to print (summaries); default * i.e. none
CLASSIFICATION = *factors*	Factors classifying the summary groups
NEWCLASSIFICATION = *factors*	Factors in the data matrix to classify the output variates
REDEFINE = *string token*	Whether to redefine the CLASSIFICATION factors and DATA variates, if NEWCLASSIFICATION or NEWDATA are not set (yes, no); default no
CMETHOD = *string token*	How to form levels for carried factors (median, minimum, maximum); default median
MVINCLUDE = *string token*	Whether to include factor combinations with no observations in summaries (yes, no); default no
WARNING = *string token*	What warnings to output (carry); default carry warns when carried factors have varying values within classification groups

Parameters

DATA = *variates*, *factors* or *pointers*	Data to be summarized
STATISTIC = *texts*	What statistic to calculate (carry, counts, sums, totals, nobservations, means, minima, maxima, variances, quantiles, sds, skewness, kurtosis, semeans, seskewness, sekurtosis); default mean
PERCENTILE = *scalars* or *variates*	Percentile to be used for quantiles; default 50.
NEWDATA = *variates*, *factors* or *pointers*	Summary statistics as variates or factors for STATISTIC=carry

Description

VSUMMARY forms data matrices containing summary statistics rather than the usual tables created by TABULATE. This can be useful if the summary statistics are to be used in a further analysis (e.g. an analysis of variance).

The CLASSIFICATION option specifies the classifying factors for the summaries, and the DATA parameter provides variates or factors to be summarized. The STATISTIC parameter specifies the type of numerical summary: counts, totals, numbers of non-missing values, means, medians, minima, maxima, variances, quantiles, standard deviations, skewness and kurtosis coefficients and (within-cell) standard errors of means, skewness and kurtosis. The statistic sums is a synonym of totals. The statistic carry, which only applies to factors, can be used to create summary factors with levels that occur in each group, e.g., in a field trial with repeated measurements in plots, we would like to carry across the factors that give the replicate and treatments for each plot. If the carried factors vary within the classification groups, a warning will be given if WARNING=carry, but this can be suppressed with WARNING=*. In the case of varying levels within groups, the CMETHOD option controls how the levels for these groups are chosen, taking either the median, minimum or maximum level present within the group for the summary level. When STATISTIC=quantiles, the PERCENTILE parameter specifies the quantile to be calculated, as a percentage between 0 and 100.

The NEWDATA parameter saves the summary statistics and the NEWCLASSIFICATION option saves new factors that gives levels of the classifying factors for the summaries. These parameters do not need to set if you set REDEFINE=yes. The DATA and the CLASSIFICATION structures

are then redefined to be the summary statistics and factors respectively.

The PRINT option allows you to print the summaries. By default, nothing is printed.

Options: PRINT, CLASSIFICATION, NEWCLASSIFICATION, REDEFINE, CMETHOD, MVINCLUDE, WARNING.
Parameters: DATA, STATISTIC, PERCENTILE, NEWDATA.

Method

VSUMMARY uses TABULATE to form tables for each statistic, and then VTABLE to extract the new summary factors and variates.

Action with RESTRICT

VSUMMARY takes account of any restrictions on the classifying factors or the DATA variates.

See also

Directives: TABULATE.
Procedures: MTABULATE, SVTABULATE, VTABLE.
Genstat Reference Manual 1 Summary sections on: Basic and nonparametric statistics, Survey analysis.

VSURFACE

Fits a 2-dimensional spline surface using REML, and estimates its extreme point (D.B. Baird).

Options

PRINT = *string tokens*	What to print (description, model, components, effects, vcovariance, deviance, waldtests, extreme, confidence, monitoring); default desc, mode, comp, wald, extr
PLOT = *string tokens*	What to plot (contour, surface); default * i.e. nothing
BASIS = *string token*	Spline basis to use (thinplate, pspline, penalizedspline); default thin
KNOTS = *scalar*, *variate* or *pointer*	Knots to be fitted in spline model, if a scalar, this is the total number of knots to be fitted; if a variate of length 2, this is the number of knots in the X1 and X2 directions; and if a pointer to 2 variates, these are the values for knots in the X1 and X2 directions; default 16
PENALTYMETHOD = *string token*	Which tensor spline penalty to use (isotropic, semiconstrained, unconstrained); default unco
DEGREE = *scalars*	Degree of polynomial used to form the underlying spline; default 1 for METHOD=penalizedspline and 3 for METHOD=pspline
DIFFORDER = *scalars*	Differencing order for p-spline penalty; default 2
EXTREME = *scalars*	Saves the estimated value of y at the extreme point
SEEXTREME = *scalars*	Saves the standard error of the estimated value of y at the extreme point
TYPEEXTREME = *string token*	Type of extreme to be identified (minimum, maximum); default maxi
PREDICTIONS = *matrix* or *pointer*	Saves predictions
PMETHOD = *string tokens*	Method of returning predictions (grid, list); default grid
NBOOT = *scalars*	The number of bootstrap samples to estimate standard errors and confidence limits; default 100
NRETRIES = *scalars*	Number of times to retry bootstrap sampling when the REML fit fails; default is the same value as NBOOT
SEED = *scalars*	The seed used to initialize the randomization in the bootstrap sampling; default 0 continues an existing sequence or, if none, selects a seed automatically
CIPROBABILITY = *scalar*	Probability level for confidence intervals for parameter estimates; default 0.95
COLOURS = *text* or *variate*	Colours for the plots

Parameters

Y = *variates*	Y-variate to which the spline surface will be fitted
X1 = *variates*	The first X-variate which defines the spline surface
X2 = *variates*	The second X-variate which defines the spline surface
ESTIMATE = *variates*	Estimated value of each x-variate at the extreme point
SE = *variates*	Standard error of the estimated value of each x-variate at the extreme point
LEVELS = *scalars*, *variates* or *pointers*	Number of values or values at which to evaluate each X

	for plots and predictions
TITLE = *texts*	Title to use for graphs; default automatically made from the variate identifiers used for Y, X1 and X2.
WINDOW = *scalars*	Window number for the graphs; default 3
SCREEN = *string tokens*	Whether to clear the screen before plotting or to continue plotting on the old screen (clear, keep); default clea
EXIT = *scalars*	Exit code from the REML fit and location of extreme point

Description

VSURFACE fits a spline surface defined by the X1 and X2 parameters to the Y variate, and estimates the extreme point within the region bounded by the values of x-variates. Parameters ESTIMATE and SE can save the estimated value of each x-variate, and their standard errors, at the extreme point. The y-value at the extreme point, and its standard error, can be saved by the EXTREME and SEEXTREME options. The TYPEEXTREME option specifies whether the extreme is a minimum or a maximum.

The BASIS option specifies whether to use thin-plate (the default), p-splines or penalized splines to construct the basis: p-splines or penalized splines are jointly known as tensor splines. Thin-plate splines are 2-dimensional cubic smoothing splines, and are formed using the THINPLATE procedure.

The positions of the knots used in the basis functions are specified by the KNOTS parameter. This can be if a scalar, specifying the total number of knots to be fitted; the procedure will then use equi-spaced knots divided proportionally to the number of distinct points in the two directions. Alternatively, it can be a variate of length 2 specifying the number of equi-spaced of knots in the X1 and X2 directions. Finally, it can be a pointer to 2 variates whose values are used for knots in the X1 and X2 directions.

The degree of polynomial used to form the underlying tensor spline basis functions is specified by the DEGREE option. This has a default of 3 for p-spline models, and 1 for penalized spline models. The DIFFORDER option specifies the differencing order to be used with p-spline models. This determines the strength of the penalty (for a given smoothness parameter). The default is to use second-order differencing. For a p-spline model, the underlying fixed polynomial in each dimension has degree d equal to DIFFORDER−1. For a penalized spline model, the underlying fixed polynomial in each dimension has degree d equal to the value specified by the DEGREE option. The tensor-spline basis is constructed via interactions of the one-dimensional spline bases, as detailed in the TENSORSPLINE procedure.

The PENALTYMETHOD option controls the interaction between the one-dimensional spline bases. An unconstrained penalty (the default) allows a separate smoothing parameter for each term. In this case, the basis pointer has $2d+3$ matrices, one for each term. With the semiconstrained penalty, the same smoothing parameter is imposed across the interaction of polynomials in the first dimension with random terms in the second, and for the interaction of random terms in the first dimension with polynomials in the second dimension. An isotropic penalty uses a single common penalty, and the terms are combined into a single matrix.

The PRINT option selects the output to be displayed:

description	description of the data and spline basis to be fitted,
model	description of model fitted,
components	estimates of variance components and estimated parameters of covariance models,
effects	estimates of the fixed and random effects,
vcovariance	variance-covariance matrix of the estimated components,

`deviance`	deviance of the fitted model (−2 × log-likelihood *RL*),
`waldtests`	Wald tests for fixed terms,
`extreme`	*y* and *x*-values of the extreme fitted value, with
`confidence`	estimated confidence limits of the extreme *y* and *x*-values obtained from the bootstrap analysis, and
`monitoring`	monitoring information at each iteration in the `REML` fitting and for each sample of the bootstrap analysis

The `EXIT` parameter saves a scalar containing the exit code from `REML` if the fit failed (−2, −1 or 1...8), or 9 if the extreme is on the boundary of the `X1`, `X2` region (so the optimum may be outside the region), or 10 if the bootstrapping has not found `NBOOT` successful fits before `NRETRIES` failures (see below). `EXIT` will be 0 if an interior optimum has been found and any bootstrapping has been successful.

If standard errors or confidence limits are required, these are formed by bootstrapping the observations. The `NBOOT` option controls the number of bootstrap samples that are taken. If the `REML` fit for a sample fails, an extra sample will be taken until a total of `NRETRIES` samples have failed, in which case the procedure exits with parameter `EXIT` set to 10. The `SEED` option controls the randomization seed used for the bootstrapping, and the `CIPROBABILITY` controls the probability levels of the confidence limits. The value of `NBOOT` must be large enough that at least one sample falls outside the confidence limits on either side (i.e. `NBOOT` >= 2/(1 - `CIPROBABILITY`)).

The `PREDICTIONS` option can save predictions and fitted values from the fitted spline model. If option `PMETHOD=list` it saves both of these, while if `PMETHOD=grid` it saves just the grid of predictions. The `LEVELS` parameter specifies the values at which to form predictions. This can be a scalar giving the number of equi-spaced grid points between the minimum and maximum of each x-variate, or a variate of length 2 which contains the number of equi-spaced grid points in the `X1` and `X2` direction, or a pointer to two variates containing the grid points to be used for `X1` and `X2`. The predictions are stored either in a matrix (the default if the structure type is not set) or in a pointer.

The `PLOT` option specifies which plots to display, with settings:

`contour`	for a contour plot, and
`surface`	for surface plot.

By default nothing is plotted. The `COLOURS` option specifies a text or variate to define the colours to use. (This is used as the setting of the `PENFILL` parameter of `DCONTOUR` and `DSURFACE`.) The default is a text containing the values `'darkgreen'` and `'yellow'`. The `TITLE`, `WINDOW` and `SCREEN` parameters control the title, window and whether a new plot is started in similar manner to those used in `DCONTOUR` and `DSURFACE`. Note that if both surface and contour plots are produced, then `SCREEN=keep` will cause these to over-plot each other in the same window.

Options: `PRINT`, `PLOT`, `BASIS`, `KNOTS`, `PENALTYMETHOD`, `DEGREE`, `DIFFORDER`, `EXTREME`, `SEEXTREME`, `TYPEEXTREME`, `PREDICTIONS`, `PMETHOD`, `NBOOT`, `NRETRIES`, `SEED`, `CIPROBABILITY`, `COLOURS`.

Parameters: `Y`, `X1`, `X2`, `ESTIMATE`, `SE`, `LEVELS`, `TITLE`, `WINDOW`, `SCREEN`, `EXIT`.

Method

`VSURFACE` forms the spline basis functions using the `THINPLATE` or `TENSORSPLINE` procedures, and fits using `REML`. The extreme value from the fitted surface (over observations and grid points), is then found. Standard errors and confidence limits are formed by bootstrap resampling of the observations.

Action with `RESTRICT`
As in `REML`, either the y-variate or x-variates can be restricted to analyse a subset of the data. If more than one of `Y`, `X1` or `X2` are restricted, the restrictions must be consistent.

See also
Directive: `REML`.
Procedures: `RQUADRATIC`, `THINPLATE`, `TENSORSPLINE`.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models..

VTABLE

Forms a variate and set of classifying factors from a table (P.W. Goedhart).

No options

Parameters

TABLE = *tables*	Tables to be copied
VARIATE = *variates*	New variate to contain the body of each table
CLASSIFICATION = *pointers*	Pointer containing the factors by which each new variate is classified
LABELS = *texts*	Labels for the new variate, indicating the values of the classifying factors corresponding to each of its units

Description

This procedure can be used to store the body of a table in a variate and obtain a set of factors to represent the way in which the data are arranged in the table. These factors then classify the newly formed variate in the same way as in the table. You can also form a text containing labels formed from the values of the classifying factors. Margins of the table are ignored.

The table to be copied is specified by the TABLE parameter, the variate must be specified by the VARIATE parameter, the set of classifying factors can be obtained by setting CLASSIFICATION to a pointer, and the labels by setting the LABELS parameter to a text. If the CLASSIFICATION pointer has not been declared, the names of the classifying factors of the table are used as suffix names. The newly formed factors have the same attributes as the old classifying factors, excluding the setting of EXTRA. Note that the order in which the factors are obtained can be unexpected for implicitly declared tables as explained in the *Guide to Genstat*, Part 1, Section 4.1.5.

Options: none. Parameters: TABLE, VARIATE, CLASSIFICATION, LABELS.

Method

Margins of the table are deleted by the directive MARGIN. The classifying factors of the table are obtained with GETATTRIBUTE. The initial declarations of the new factors are done by DUPLICATE to transfer any relevant attributes. Factor values are then produced by GENERATE.

See also

Procedure: FBETWEENGROUPVECTORS, VMATRIX.
Genstat Reference Manual 1 Summary section on: Calculations and manipulation.

VTCOMPARISONS

Calculates comparison contrasts within a multi-way table of predicted means from a REML analysis (R.W. Payne).

Options

PRINT = *string tokens*	Controls printed output (contrasts, Waldtests); default cont
MODEL = *formula*	Indicates which model terms (fixed and/or random) are to be used in forming the predictions; default * includes all the fixed terms and relevant random terms
OMITTERMS = *formula*	Specifies terms to be excluded from the MODEL; default * i.e. none
FACTORIAL = *scalar*	Limit on the number of factors or variates in each term in the models specified by MODEL or OMITTERMS; default 3
PRESENTCOMBINATIONS = *identifiers*	Lists factors for which averages should be taken across combinations that are present
WEIGHTS = *tables*	One-way tables of weights classified by factors in the model; default *
GROUPS = *factors*	Groups for which to estimate each contrast
DFMETHOD = *string token*	Specifies which degrees of freedom to use for the comparisons (fddf, given, tryfddf, none); default fddf
DFGIVEN = *scalar*	Specifies the number of degrees of freedom to use for the comparisons when DFMETHOD=given, or if d.d.f. are unavailable when DFMETHOD=tryfddf
FMETHOD = *string token*	Controls how to calculate denominator degrees of freedom for the F-statistics, if these are not already available in the REML save structure (automatic, algebraic, numerical); default auto
SAVE = *identifier*	REML save structure for the analysis from which the comparisons are to be calculated

Parameters

CONTRAST = *tables*	Defines the comparisons to be estimated
ESTIMATE = *scalars* or *variates*	Saves the estimated contrasts
SE = *scalars* or *variates*	Saves standard errors of the contrasts
VCOVARIANCE = *symmetric matrices*	Save the variance-covariance matrices of contrasts estimated for GROUPS
STATISTIC = *scalars* or *variates*	Saves saves the test statistic (t or Wald)
DF = *scalars* or *variates*	Saves estimated numbers of residual degrees of freedom of the contrasts
PROBABILITY = *scalars* or *variates*	Saves the probabilities of the contrasts
WALD = *scalars*	Wald statistic for each comparison, combining the tests within groups
FSTATISTIC = *scalars*	F statistics for each comparison, if available, combining the tests within groups
NDF = *scalars*	Numerator d.f. for FSTATISTIC

`DDF` = *scalars* Denominator d.f. for `FSTATISTIC`

Description

`VTCOMPARISON` makes comparisons within multi-way tables of predicted means from a `REML` analysis. The data should previously have been analysed by the `REML` directive in the usual way. The `SAVE` option can be used to specify the save structure from the analysis for which the comparisons are to be calculated (see the `SAVE` option of `REML`). If `SAVE` is not specified, the comparisons are calculated from the most recent `REML` analysis.

Each comparison is specified in a table supplied by the `CONTRAST` parameter. `VTCOMPARISON` calculates the means using the `VPREDICT` directive. The calculations consist of two steps. The first step is to calculate a table of fitted values. The `MODEL`, `OMITTERMS` and `FACTORIAL` options specify the model to use for this. The formula specified by `MODEL` is expanded into a list of model terms, deleting any that contain more variates or factors than the limit specified by the `FACTORIAL` option. Then, any terms in the formula specified by `OMITTERMS` are removed. The second step averages the fitted values over the classifications that are not in the list that was supplied by the `CLASSIFY` parameter. The `WEIGHTS` option can supply one-way tables classified by any of the factors in the model. These are used to calculate the weight to be used for each fitted value when calculating the averages. Equal weights are assumed for any factor for which no table of weights has been supplied. In the averaging all the fitted values are generally used. However, if you define a list of factors using the `PRESENTCOMBINATIONS` option, any combination of levels of these factors that does not occur in the data will be omitted from the averaging. Where a prediction is found to be inestimable, i.e. not invariant to the model parameterization, a missing value is given.

The `GROUPS` option is useful if you want to calculate the same comparisons for several groups, defined by the combinations of levels of one or more factors in the `REML` analysis. You can then use the `CONTRAST` parameter to define the comparison-definition tables ignoring the groups, and the `GROUPS` option to specify the factors defining the groups.

The `DFMETHOD` option specifies how to obtain the numbers of residual degrees of freedom for the comparisons. The default is to use the numbers of denominator degrees of freedom printed by `REML` in the `d.d.f.` column in the table of tests for fixed tests (produced by setting option `PRINT=wald`). These degrees of freedom are relevant for assessing the fixed term as a whole, and may differ over the various comparisons amongst its means, or for predictions produced with different models or weightings from those used in `REML` and `VDISPLAY`. So the t-probabilities should be used with caution. If you want a more exact probability for a comparison, you should set up a covariate to fit this explicitly in the analysis. The `FMETHOD` option controls how the denominator degrees of freedom should be calculated, if they are not already available in the `REML` save structure (e.g. because they were printed in the original analysis). The settings are the same as in the `REML` and `VKEEP` directives, except that there is no `none` setting. (You would set this option only if you really do want to calculate them.)

In some of the more complicated analyses, `REML` may be unable to calculate the denominator degrees of freedom. You might then want to supply the number of degrees of freedom yourself, using the `DFGIVEN` option, rather than having no probabilities at all. For example, you could use the number of denominator degrees of freedom from the analysis of an earlier similar design. However, the results will only be as good as the degrees of freedom that you have supplied, and thus should be used with caution! You can set option `DFMETHOD=tryfddf` to use the denominator degrees of freedom, if these can be calculated, or those specified by `DFGIVEN` otherwise. The setting `DFMETHOD=given` always uses the degrees of freedom specified by `DFGIVEN`.

If no d.d.f. are available, `VTCOMPARISONS` forms Wald statistics instead of t-statistics, and calculates their probabilities using the fact that, asymptotically, they have chi-square distributions with one degree of freedom. The Wald probabilities tend to be biased (giving too

many significant results), and should thus be used with caution. You can set DFMETHOD=none to enforce the use of Wald statistics.

The PRINT option controls printed output, with settings:

contrasts	prints the contrasts (default); and
Waldtests	when GROUPS is set this prints Wald tests combining the tests of each contrast in the various groups, F tests are also given provided REML has been able to estimate the d.d.f.

The ESTIMATE parameter allows you to save the estimates for the comparisons. If the GROUPS option is not set, each comparison will have a single estimate which will be saved in a scalar. Alternatively, if there are groups, there will be an estimate for each group, and these will be saved in a variate defined with unit labels that identify the groups. Similarly, the SE parameter can save the standard errors of the comparisons, the DF parameter can save their estimated number of residual degrees for freedom, the STATISTIC parameter can save their test statistics (t or Wald), and the PROBABILITY parameter can save their probabilities.

When there are groups, the variances and covariances of the estimates for each contrast can be saved in a symmetric matrix, using the VCOVARIANCE parameter. The WALD, FSTATISTIC, NDF and DDF parameters can save the results of the tests combining the tests for each contrast in the various groups.

Options: PRINT, MODEL, OMITTERMS, FACTORIAL, PRESENTCOMBINATIONS, WEIGHTS, GROUPS, DFMETHOD, DFGIVEN, FMETHOD, SAVE.

Parameters: CONTRAST, ESTIMATE, SE, VCOVARIANCE, STATISTIC, DF, PROBABILITY, WALD, FSTATISTIC, NDF, DDF.

See also

Directive: VPREDICT.

Procedures: FCONTRASTS, RTCOMPARISONS.

Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

VUVCOVARIANCE

Forms the unit-by-unit variance-covariance matrix for specified variance components in a REML model (R.W. Payne).

Options

FIXED = *formula*	Fixed model terms; default *
CONSTANT = *string token*	How to treat the constant term (estimate, omit); default esti
FACTORIAL = *scalar*	Limit on the number of factors or covariates in each fixed term; default 3
SEED = *scalar*	Seed for the random numbers used to generate a dummy y-variate; default 12345

Parameters

RANDOM = *formula structures*	Random model terms
COMPONENTS = *variates*	Values for the variance components and residual variance
UVCOVARIANCE = *symmetric matrices*	Saves the unit-by-unit variance-covariance matrices

Description

Procedure VBOOTSTRAP assesses the significance of for fixed terms in a REML analysis, by using a parametric bootstrap. The bootstrap samples are generated from a multivariate Normal distribution with dimension equal to the number of units in the analysis, and this requires a unit-by-unit variance-covariance matrix.

If you want to take the variance-covariance matrix from a previous analysis, this can be done by using the UVCOVARIANCE option of VKEEP. VUVCOVARIANCE provides a solution for the situation where there is no suitable previous analysis, but you can make reasonable assumptions about the likely sizes of the variance components. Note, however, that it cannot handle covariance models.

The RANDOM parameter specifies a formula structure, defining the random model. This may also contain the residual term, but that is not essential. If it is not present, the residual is added as the final model term. The COMPONENTS parameter specifies the variance components for the random terms, including the residual variance (at the end, if VUVCOVARIANCE needs to add this to the RANDOM formula). The UVCOVARIANCE parameter saves the unit-by-unit variance-covariance matrix.

The FIXED, CONSTANT and FACTORIAL options define the fixed model in the usual way. (See VCOMPONENTS.) The SEED option provides a seed for random numbers (default 12345). These are used to generate a dummy y-variate, that is used in a REML analysis inside the procedure to calculate UVCOVARIANCE.

Options: FIXED, CONSTANT, FACTORIAL, SEED.
Parameters: RANDOM, COMPONENTS, UVCOVARIANCE.

Method

UVCOVARIANCE obtains the unit-by-unit variance-covariance matrix by using the COMPONENTS as initial values for the random terms in VCOMPONENTS, and performing a REML analysis with the number of iterations set to zero. The y-variate for the analysis contains Normally distributed random numbers, generated with the specified SEED.

See also
Directive: REML, VCOMPONENTS.
Procedure: VBOOTSTRAP.
Genstat Reference Manual 1 Summary section on: REML analysis of linear mixed models.

WADLEY

Fits models for Wadley's problem, allowing alternative links and errors (D.M. Smith).

Options

PRINT = *string tokens*	Controls printed output (deviance, estimates, correlations, monitoring); default devi, esti
DISTRIBUTION = *string token*	Distribution of the response variate (poisson, negativebinomial, qlnegativebinomial, qlscaledpoisson); default pois
LINK = *string token*	Link transformation (logit, probit, complementaryloglog, cauchit); default logi
TERMS = *formula*	Model to be fitted
CONTROL = *factor*	Factor to distinguish the control, or zero, dose (level 1) from the other treatments (level 2)
MAXIMAL = *factor*	Factor to define the maximal model i.e. with a level for every combination of values of the variates and factors in TERMS
RMETHOD = *string token*	Type of residuals to be formed (deviance, Pearson); default devi

Parameters

Y = *variates*	Response variate for each fit
RESIDUALS = *variates*	Variate to save the residuals from each fit
FITTEDVALUES = *variates*	Variate to save the fitted values from each fit

Description

WADLEY uses the generalized linear models methodology of composite link functions to fit a range of models for the situation known as Wadley's problem. This arises in bioassay where it is possible to count only the number of subjects that have not responded to a particular dose of a drug or stimulus. For example, with eggs of insects fumigated in grain, it is generally possible to count only those that survive and hatch.

By default, the analysis assumes that the numbers of subjects that are treated in each observation follow a Poisson distribution with a common mean parameter; other distributions can be specified using the DISTRIBUTION option or, for user-defined distributions, by providing subsidiary procedure WADDISTRIBUTION (see details of the procedures called by WADLEY).

The analysis estimates the mean of the distribution, and then fits the dose response curve as in an ordinary probit analysis. The LINK option defines the transformation (logit, probit, cauchit, or complementary log-log) required to make the model additive. User-defined transformations can also be specified, by leaving LINK unset and providing subsidiary procedure WADLINK to calculate the necessary fitted values and derivatives, and WADINITIAL to calculate initial values for the linear predictor (see details of the procedures called by WADLEY). The model to be fitted is defined by the TERMS option.

To assist the estimation of the expected total number of subjects, there must be some control observations – for example with zero doses of fumigant. These must be identified by a factor, specified by the CONTROL option, with level 1 for untreated and level 2 for treated. The comparison between the treated and untreated levels of CONTROL must not be aliased with any of the variates and factors in TERMS. (Thus if, for example, TERMS contained a factor representing different types of drug, this must not have a separate level for the untreated observations.)

Often with these sort of data, it is found that the variability exceeds that which would be expected from the distribution assumed for the data. To estimate the amount of overdispersion,

the MAXIMAL option must be set to a factor with a different level for every combination of values of the factors and variates in the TERMS model.

Options: PRINT, DISTRIBUTION, LINK, TERMS, CONTROL, MAXIMAL, RMETHOD.
Parameters: Y, RESIDUALS, FITTEDVALUES.

Method

In essence WADLEY is a specific application of the use of composite link functions in generalized linear models. The actual methods used are those in the Genstat procedure GLM (Lane 1989) and the GLIM macros of Smith & Morgan (1989). The procedure is very similar in spirit to these GLIM macros, and it is recommended that this reference be consulted for further information. However, there are some extensions. The capability to handle user-defined links and distributions has been added. Also, the range of distributions has been extended to include two forms of quasi-likelihood, namely that where the weighting is of negative binomial form (weight=1/(1+*hf*×fittedvalues)), and that where the weighting is of scaled Poisson form (weight=1/*hf*), where *hf* is the heterogeneity factor. If the estimated heterogeneity factor is less than zero in the negative binomial cases, or if it is less than one in the scaled Poisson case, it is set to zero or one respectively.

WADLEY has two subsidiary procedures, WADCODI and WADFIT, to assist with the analysis; neither of these need be modified by the user:

WADCODI prints the results of the iterative processes;

WADFIT performs the iterative model fits.

There are also three other procedures, which can be rewritten or replaced, to cater for further user-defined distributions and links:

WADDISTRIBUTION calculates the variance function and deviance for a user-defined distribution;

WADINITIAL calculates initial estimates of the linear predictor for a user-defined link;

WADLINK calculates the fitted values and derivatives for a user-defined link.

If the DISTRIBUTION option is unset, the procedure will call WADDISTRIBUTION instead of using one of the various standard distributions. For a Poisson error distribution WADDISTRIBUTION should be defined like this.

```
PROCEDURE 'WADDISTRIBUTION'
          "Calculation of variance function and deviance"
PARAMETER 'Y',        "Input: variate; response variate"\
          'FITTED',   "Input: variate; fitted values"\
          'VARIANCE',"Output: variate; variance"\
          'LL',       "Output: variate; log likelihood variate"\
          'DEVIANCE';"Output: scalar; total deviance"\
          MODE=p
          SCALAR two; VALUE=2
          CALCULATE VARIANCE = FITTED
          &         LL       = Y*LOG(Y/FITTED)-Y+FITTED
          &         DEVIANCE = two*SUM(LL)
ENDPROC
```

For other error distributions only the three CALCULATE statements need to be changed.

Similarly, for option LINK unset, WADINITIAL and WADLINK will be called. For a logit link WADINITIAL would be defined as follows.

```
PROCEDURE 'WADINITIAL'
          "Calculation of initial estimates of linear predictor"
PARAMETER 'Y',    "Input: variate; response variate"\
          'LP',   "Output: variate; linear predictor"\
          'IND',  "Input: variate; marker variate with value 1
                   for a control observation, 0 otherwise"\
          'MAXY'; "Inout: scalar; estimate of asymptote"\
```

```
            MODE=p
            SCALAR half,one; VALUE=0.5,1
            CALCULATE LP = IND*LOG(MAXY/(Y+half)-one)
ENDPROC
```

For other links only the CALCULATE statement need be changed so, for example, a probit link would require the statement

```
    CALCULATE LP = IND*NED(one-(Y+one)/MAXY)
```

For a logit link WADLINK would be

```
PROCEDURE 'WADLINK'
          "Calculation of fitted values and derivatives
          of the link function given the linear predictor"
PARAMETER 'LP',   "Input: variate; linear predictor"\
          'IND',  "Input: variate; marker variate with value 1
                   for a control observation, 0 otherwise"\
          'TA',   "Output: variate; estimate of fitted values"\
          'TB',   "Output: variate; estimate of derivatives"\
          'MAXY'; "Input: scalar; estimate of asymptote"\
          MODE=p
   SCALAR half,one; VALUE=0.5,1
          CALCULATE TA = (.NOT.IND)+IND/(one+EXP(LP))
          &         TB = MAXY*EXP(LP)*TA*TA
ENDPROC
```

For other links only the CALCULATE statements need to be changed so, for example, a probit link would require

```
    CALCULATE TA = (.NOT.IND)+IND/(one-NORMAL(LP))
    &         TB = MAXY*EXP(-half*LP*LP)/ROOT2PI
```

where ROOT2PI is a scalar with the value of the square root of 2π. The marker variate IND identifies which is the control and non control data, so TA should always be of the form

```
    TA = (.NOT.IND)+IND*function
```

where function is the link function for the non-control part of the data. The variate TB should always be of the form

```
    TB = MAXY*deriv_fn
```

where deriv_fn is the derivative of the link function with respect to the linear predictor (LP).

If LINK or DISTRIBUTION are unset, but no user routines are given for WADINITIAL, WADLINK and WADDISTRIBUTION, then those given here (for logit link and Poisson error distribution) will be used.

A debt is owned to Dr J. Parrott of Pfizer Central Research, Sandwich, UK for his support and encouragement of this work.

Action with RESTRICT

If the Y-variate is restricted, only the specified subset of the units will be included in the analysis.

References

Lane, P.W. (1989). Procedure GLM. In: *Genstat Procedure Library Release 1.3[2]* (ed. R.W.Payne & G.M.Arnold), 80-82.

Smith, D.M. & Morgan, B.J.T. (1989). Extended models for Wadley's Problem. *Glim Newsletter*, **18**, 21-28.

See also

Procedure: PROBITANALYSIS.

Genstat Reference Manual 1 Summary section on: Regression analysis.

WILCOXON

Performs a Wilcoxon Matched-Pairs (Signed-Rank) test (S.J. Welham, N.M. Maclaren & H.R. Simpson).

Option

PRINT = *string tokens*	Output required (test, ranks): test gives the relevant test statistics, ranks prints out the signed ranks for the vector of differences; default test

Parameters

DATA = *variates*	Variates holding the differences between each pair of samples
RANKS = *variates*	Saves the signed ranks
STATISTIC = *scalars*	Saves each test statistic
PROBABILITY = *scalars*	Saves the probability for each test statistic
SIGN = *scalars*	Scalar to indicate the sign of the total sum of each set of signed ranks: 1 if the sum is positive, 0 otherwise

Description

WILCOXON performs a Wilcoxon Matched-Pairs test on a variate holding differences between two paired samples. This is specified using the DATA parameter. The test statistic can be saved using the STATISTIC parameter. The probability can be saved using the PROBABILITY parameter; this is for a two-sided test i.e. no assumption is made about whether the differences should be positive or negative. The SIGN parameter can save an indicator of whether the total sum of signed ranks is positive (SIGN=1) or negative (SIGN=0), and the RANKS parameter can save a variate of the signed ranks of the differences (i.e. of DATA).

Output from the procedure is controlled by the PRINT option: test produces the relevant test statistics, and ranks prints the vector of signed ranks for the data.

Option: PRINT. Parameters: DATA, RANKS, STATISTIC, PROBABILITY, SIGN.

Method

The Wilcoxon Matched-Pairs test (often also called the Wilcoxon Signed-Ranks test) is a nonparametric test of location in the case of two related samples (e.g. a before-and-after study). The null hypothesis is that two samples arise from exactly the same distribution, with the alternative that the two underlying distributions differ only in location.

The test statistic *WS* is formed from the signed ranks of the differences between each pair of observations and is the smaller in absolute value out of:

1) the sum of positive signed-ranks of the sample, and
2) the sum of the negative signed-ranks.

In this procedure the method used for calculating the test statistic is:

$WS = N \times (N+1)/4$ – modulus(total sum of signed ranks)/2

where N is the number of observations. The probability is calculated using the PRWILCOXON procedure.

For further information, see Siegel (1956) pages 75-83.

Action with RESTRICT

If the DATA variate is restricted, the test is calculated only using the units not excluded by the restriction.

Reference

Siegel, S. (1956). *Nonparametric Statistics for the Behavioural Sciences*. McGraw-Hill, New York.

See also

Procedure: PRWILCOXON, MANNWHITNEY, SIGNTEST, TTEST.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

WINDROSE

Plots rose diagrams of circular data like wind speeds (P.W. Goedhart & R.W. Payne).

Options

PRINT = *string token*	What to print (table); default * i.e. nothing
SEGMENT = *scalar*	Width of sectors (in degrees) into which to group an ANGLES variates before plotting; default 45
MSEGMENT = *scalar*	Defines the centre (in degrees) of the sectors; default 0
INTERVALS = *scalar* or *variate*	Scalar to define the intervals at which to summarize the data values, or a variate defining the boundaries between the intervals; default * i.e. determined automatically
%INTERVAL = *scalar*	Interval (on the percent scale) between the circles drawn to provide a scale on the diagram; default * i.e. determined automatically
COLOURS = *text* or *variate*	Colours to shade the triangles segment for each interval; default * sets suitable colours automatically
SCREEN = *string token*	Whether to clear screen before displaying the graphs (keep, clear); default clea

Parameters

DATA = *variates*	Data values
ANGLES = *factors* or *variates*	Directions of the data values
TITLE = *text*	Title for the graph; default * i.e. identifier of the DATA variate
WINDOW = *scalar*	Window for the graph; default 3

Description

WINDROSE plots data, like wind speeds, that are observed at angles around a circle. The data values are supplied in a variate by the DATA parameter. The angles at which the data values were observed are specified by ANGLES parameter. If this is set to a variate, WINDROSE groups the observations into sectors of width specified (in degrees) by the SEGMENT option, with centres defined by the MSEGMENT option. The sectors are centred at MSEGMENT, MSEGMENT+SEGMENT, MSEGMENT+2*SEGMENT, and so on. The default values for SEGMENT and MSEGMENT are 45 and 0 respectively. Alternatively, ANGLES can be set to a factor; its levels then define the midpoints of the sectors (and these must be in clockwise order).

WINDROSE categorizes the data values by determining the number of observations within a set of intervals specified by the INTERVALS option. The option can supply a variate specifying the lower boundaries of the intervals, or a scalar defining boundaries at multiples of the value that it contains. The diagram has a circular segment for each direction, with radius equal to the percentage of the total observations that are in that direction. To indicate the distribution of the data values in that direction, the segment is subdivided into a section for each interval. The sections are shaded in colours, which can be specified by the COLOURS option; ; by default, the standard colours are used in the same order as for pens 2, 3... (see PEN). Zero, negative or missing values of the DATA variate are assumed to represent "calm" values. These are represented by an empty circle at the centre of the diagram. Circles are drawn at intervals around this inner circle to provide a scale. The intervals between these circles are specified by the %INTERVAL option. You can also print the information, as a two-way table (directions × intervals) by setting option PRINT=table.

The parameters allow several rose diagrams to be plotted at once. The SCREEN option controls whether the existing screen is kept or cleared before plotting begins; by default SCREEN=clear. The WINDOW parameter specifies the graphics window in which each diagram is plotted. If these

are not specified, the FFRAME procedure is used to set up a rectangular array with a window for each diagram. The TITLE parameter can be used to supply a title for each plot; if this is not specified, the identifier of the DATA variate is used.

Options: PRINT, SEGMENT, MSEGMENT, INTERVALS, %INTERVAL, COLOURS, SCREEN.
Parameters: DATA, ANGLES, TITLE, WINDOW.

Method
WINDROSE uses Genstat's standard graphics and calculation commands.

Action with RESTRICT
If DATA or ANGLES are restricted, only the unrestricted units are used.

See also
Procedures: CASSOCIATION, CCOMPARE, CDESCRIBE, DCIRCULAR, DYPOLAR, GRIBIMPORT, RCIRCULAR.
Genstat Reference Manual 1 Summary sections on: Graphics, Basic and nonparametric statistics.

WSTATISTIC

Calculates the Shapiro-Wilk test for Normality (R.W. Payne).

Option

PRINT = *string tokens*	What to print (test); default test

Parameters

DATA = *variates*	Samples of data to be tested for Normality
W = *scalars*	Saves the Shapiro-Wilk W statistic for each sample
PROBABILITY = *scalars*	Saves the probability for W under the assumption that the data are Normal

Description

WSTATISTIC uses the Shapiro-Wilk test to assess whether a sample of data comes from a Normal distribution. The data values must be supplied, in a variate, using the DATA parameter. By default WSTATISTIC prints the statistic, W, with its probability value under the assumption that the data are Normal. (So a low probability indicates that the data are unlikely to be from a Normal distribution.) The printed output can be supressed by setting option PRINT=*. The test statistic can be saved, in a scalar, using the W parameter, and its probability can similarly be saved using the PROBABILITY parameter.

Option: PRINT.
Parameters: DATA, W, PROBABILITY.

Method

WSTATISTIC calculates the statistic and its probability using the methods of Royston (1993, 1995).

Action with RESTRICT

The DATA variate can be restricted to assess a subset of the data.

References

Royston, P. (1993). A toolkit for testing for non-normality in complete and censored samples. *The Statistician*, **42**, 37-43.

Royston, P. (1995). A remark on Algorithm AS 181: the W-test for Normality. *Applied Statistics*, **44**, 547-551.

See also

Directive: DISTRIBUTION.
Procedures: EDFTEST, NORMTEST.
Genstat Reference Manual 1 Summary section on: Basic and nonparametric statistics.

XOCATEGORIES

Performs analyses of categorical data from cross-over trials (D.M. Smith & M.G.Kenward).

Options

PRINT = *string token*	What to print at each fit (model, summary, accumulated, estimates, correlations, fittedvalues, monitoring); default *
PDATA = *string token*	Whether or not a display of category combination by sequence is required (yes, no); default no
METHOD = *string token*	Type of analysis for which factors are required (subject, loglinear, ownsubject, ownloglinear); default subj
CARRYOVER = *string token*	Whether or not models with carryover effects in are to be produced (yes, no); default no

Parameters

SEQUENCE = *factors*	The identifier of the sequence of treatments
RESULTS = *pointers*	Pointer containing factors (one for each period) giving the category scores observed
NUMBER = *variates*	Numbers recorded in the sequence/category combinations
SAVE = *pointers*	Saves the factors constructed to do the analysis
REUSE = *pointers*	To reuse factors saved earlier using SAVE
MODEL = *formula*	Additional terms to be fitted to model if OWNSUBJECT or OWNLOGLINEAR options used; default *

Description

XOCATEGORIES calculates factors, variates and performs various analyses of categorical cross-over data. All analyses conform to one of two different types both utilising a log-linear structure, although only one is derived from an orthodox log-linear model. The first type is based on a latent variable or subject effects model and is described by Kenward & Jones (1991). The subject effects are eliminated through the use of a conditional likelihood and the resultant conditional analysis can be formulated in terms of a conventional log-linear model. In the process of conditioning all between-subject information is lost. This has little consequence for the majority of well-designed cross-over trials in which nearly all information on important comparisons lies in the within-subject stratum. An exception to this is the two-period two-treatment design for which information on the carry-over effect lies in the between-subject stratum. The second type, which uses a multivariate log-linear model, allows between-subject information to be recovered, which in the binary case leads to the Hills-Armitage test for carry-over effect. Details can be found in Jones & Kenward (1989, Section 3.3). If such a test, and other allied tests for the two-period two-treatment design, are required then the log-linear option of the procedure can be used. However, the estimates from this multivariate log-linear model do have the disadvantage of an awkward interpretation. For this reason the latent variable model is to be preferred for higher-order designs and for the two-period two-treatment design when the carry-over test is not required. In the latter case, with binary data, the test for direct treatments reduces to the Mainland-Gart test.

In the latent variable model, effects are defined in terms of generalized logits, reducing to ordinary logits in the binary case. This is not ideal for ordered categorical data because the ordering is ignored. Some account can be taken of the ordering of categories by regressing on category scores in a generalization of Armitage's trend test. This can be done by using the parameter SAVE to obtain the treatment and carryover factors, which are in pointers

SAVE[3...(NTRT+2)] and SAVE[(NTRT+3)...(2+2*NTRT)] respectively, NTRT being the number of treatments. From these treatment and carryover factors (NCAT−1) variates corresponding to linear (−1, 0, 1), quadratic (1, −2, 1), etc., contrasts amongst the NCAT categories can be calculated. For example, using the example data where the number of treatments (NTRT) is 3 and the number of categories (NCAT) is 3, the following statements will produce linear and quadratic variates for treatments.

```
XOCATEGORIES SEQUENCE=Seqid; RESULTS=Res; NUMBER=Number; \
  SAVE=Fsave
CALCULATE TLN[1...3]=(Fsave[3...6].eq.3)-(Fsave[3...6].eq.1)
 & TQU[1...3]=(Fsave[3...6].eq.3)-2*(Fsave[3...6].eq.2)\
              +(Fsave[3...6].eq.1)
```

The OWNSUBJECT or OWNLOGLINEAR options together with the REUSE and MODEL parameters can then be used to fit models involving these variates and the deviances produced used to compare the models. For example, for the above variates.

```
XOCATEGORIES [METHOD=OWNSUBJECT] REUSE=Fsave; \
 MODEL=TLN[1]+TLN[2]++TLN[3]+TQU[1]+TQU[2]+TQU[3]
```

The data for the procedure are specified by parameters SEQUENCES, RESULTS and NUMBERS. SEQUENCES supplies a factor with labels indicating the treatment received at each time period. The treatments are labelled by capital letters A, B &c, so (with three periods) BCA indicates treatment B in period 1, C in 2 and A in 1. RESULTS is a pointer containing a factor for each time period, to indicate the corresponding scores recorded in each period. NUMBER then indicates the number of subjects involved. It is not necessary to input data for category combinations in which no subjects were recorded.

XOCATEGORIES processes the data to form the necessary factors to do the analysis using the Genstat facilities for generalized linear models. This information can be saved using the SAVE parameter (see Method) and input again, to save time in later analyses, using the REUSE parameter.

Output of the procedure comprises significance tests of treatment and/or carryover and first order period interactions; together with estimates of log odds ratios and their standard errors.

Options: PRINT, PDATA, METHOD, CARRYOVER.
Parameters: SEQUENCE, RESULTS, NUMBER, SAVE, REUSE, MODEL.

Method

The methods of analysis follow Kenward & Jones (1991) for SUBJECT and OWNSUBJECT, and Jones & Kenward (1989, pages 124-129) for LOGLINEAR and OWNLOGLINEAR. The actual model fitting is performed using Genstat directives FIT, ADD, DROP and SWITCH, with the PRINT options being those of these directives.

The data structure SAVE has the following form, all factors as Kenward & Jones (1991).

SAVE[1]	= The factor G (sequence).
SAVE[2]	= The factor S (outcome).
SAVE[3...(NTRT+2)]	= The factors T[1...NTRT] (treatment).
SAVE[(NTRT+3)...(2+2*NTRT)]	= The factors C[1...NTRT] (carryover).
SAVE[(3+2*NTRT)...(NPER+2*NTRT)]	= The factors P[1...NPER] (period).
SAVE[NPER+2*NTRT+1]	= The category labels if they exist.

We wish to thank Dr Byron Jones of the University of Kent, Canterbury UK, for his assistance.

Action with RESTRICT

Input structures must not be restricted, and any existing restrictions will be cancelled.

References

Jones, B. & Kenward, M.G. (1989). *Design and Analysis of Crossover Trials*. Chapman & Hall, London.

Kenward, M.G. & Jones, B. (1991). The analysis of categorical data from cross-over trials using a latent variable model. *Statistics in Medicine*, **10**, 1607-1619.

See also

Procedures: AFCARRYOVER, AGCROSSOVERLATIN, XOEFFICIENCY, XOPOWER.
Genstat Reference Manual 1 Summary section on: Regression analysis.

XOEFFICIENCY

Calculates efficiency of estimating effects in cross-over designs (B. Jones & P.W. Lane).

Options

PRINT = *string tokens*	What reports to produce (summary, efficiency, variance, carryover, contrasts, dummyanalysis, incidence); default summ, effi, cont
NPERIODS = *scalar*	Number of periods in the design; no default
CARRYOVER = *string token*	Whether to included effects of carryover (yes, no); default no
CONTRASTTYPE = *string token*	Type of treatment contrasts if POLYNOMIAL and OWN parameters are unset (pairwise, control); default pair
INCIDENCE = *pointer*	Saves incidence matrices; default *

Parameters

SEQUENCES = *formula*	Text, variate or factor with sequence of levels of a single treatment; no default
POLYNOMIAL = *scalars*	Order of polynomials to represent each term in the SEQUENCES parameter; default *, i.e. represent effects according to OWN parameter or CONTRASTTYPE option
OWN = *matrices*	Specific contrasts for each term in the sequences parameter; default *, i.e. represent effects according to POLYNOMIAL parameter or CONTRASTTYPE option
EFFICIENCY = *symmetric matrices*, *variates* or *diagonal matrices*	Saves efficiencies; default *
VARIANCE = *symmetric matrices*, *variates* or *diagonal matrices*	Saves variances; default *

Description

The simplest use of procedure XOEFFICIENCY is for a cross-over design with a single treatment factor. The SEQUENCES parameter should then be set to a factor indicating the treatment level to be applied at each period for each patient: the ordering must be such that the sequence of levels for the first patient come first, then the sequence for the second patient, and so on. The NPERIODS option must be set to specify the number of periods. If preferred, the sequences can be input just as a text or variate structure containing the textual or numeric codes of the treatment, leaving the procedure to form the factor internally.

The procedure calculates the efficiency of estimating the treatment effects. By default, it reports the efficiency of the design for each estimated pairwise difference of treatment levels, together with the mean of these differences. Alternatively, the CONTRAST option can be set to control to request efficiencies of the differences of each level with the reference level of the treatment factor (the first level, by default). Another possibility is to set the POLYNOMIAL parameter to the order of polynomial effects to be estimated for the treatment levels; orthogonal polynomials will be used, based on the marginal replication of the treatment levels. Finally, the OWN parameter can be set to a matrix that specifies comparisons between the treatment levels: the matrix must have one column for each treatment level and one row for each desired contrast. If either of POLYNOMIAL or OWN is set, the CONTRASTTYPE option is ignored.

The PRINT option controls which reports are displayed. By default, a summary of the design is given, and then a symmetric matrix of the efficiencies of each difference between pairs of treatment levels together with the mean pairwise efficiency. In addition, the chosen contrasts are displayed, unless the default pairwise contrasts are required. The variance setting displays the

variance of each contrast. The `incidence` setting displays two tables of the numbers of observations in the design: the first is classified by Subject and Treatment, and the second by Treatment and Period. (There is no point displaying the classification by Subject and Period, since this always consists of a 1 in each cell for the designs dealt with by this procedure.) The `aov` setting produces a skeleton analysis of variance of the specified design, if the design is generally balanced.

By default, carry-over effects are ignored. If the `CARRYOVER` option is set to `yes`, first-order carry-over effects are included in the model, and efficiencies for treatments will be adjusted accordingly. If the `carryover` setting is included in the `PRINT` option, the efficiencies and variances of the carry-over contrasts are displayed in the same way as for the treatment contrasts (that is, with regard to the setting of the `CONTRASTTYPE` option, `POLYNOMIAL` and `OWN` parameters, and the `efficiency` and `variance` settings of the `PRINT` option). If the `incidence` option is included, a further three incidence tables will be displayed: Treatment by Carry-over, Subject by Carry-over, and Carry-over by Period.

The `INCIDENCE` option allows the incidence information, as printed by `PRINT=incidence`, to be stored. It should be set to the identifier of a pointer, which will be set up by the procedure with elements labelled to identify the matrices concerned. If there is no carry-over, the pointer will point to two matrices, ordered as for the `PRINT` option; if there is carry-over, there will be five matrices.

The `EFFICIENCY` and `VARIANCE` parameters allow the variances and efficiencies of the treatment effects to be stored in symmetric matrices (for pairwise differences), variates (for differences with control), or diagonal matrices (for polynomial or own contrasts). If the option `CARRYOVER` is set to `yes`, the stored results will be for the carry-over effects; to get the results for the treatment effects, the procedure must be invoked again with the `CARRYOVER` option set to `no`.

Options: `PRINT`, `NPERIODS`, `CARRYOVER`, `CONTRASTTYPE`, `INCIDENCE`.
Parameters: `SEQUENCES`, `POLYNOMIAL`, `OWN`, `EFFICIENCY`, `VARIANCE`.

Method
The efficiency of a contrast is calculated as the ratio of its theoretically optimal variance to its variance in the supplied design, expressed as a percentage. The optimal variance may not actually be attainable. It is calculated as the variance for the contrast in a design with the same marginal replication of treatment levels, but where the treatment factor is orthogonal to all other factors in the design. For example, the optimal variance for a contrast between two treatment levels (omitting any estimate of dispersion) is calculated as $(1/n_1 + 1/n_2)$, where n_1 and n_2 are the replications of the two levels. The actual variance of the supplied design is calculated by fitting a linear model by linear regression, including terms as specified in the options. The inverse matrix then provides the variance, omitting the estimate of dispersion which would cancel out in the ratio anyway.

Action with `RESTRICT`
No structures should be restricted.

See also
Procedures: `AFCARRYOVER`, `AGCROSSOVERLATIN`, `XOCATEGORIES`, `XOPOWER`.
Genstat Reference Manual 1 Summary section on: Design of experiments.

XOPOWER

Estimates the power of contrasts in cross-over designs (P.W. Lane & B. Jones).

Options

PRINT = *string tokens*	What reports to produce (summary, contrasts, nonequality, equivalence, noninferiority, superiority); default summ, none
NPERIODS = *scalar*	Number of periods in the design; default 2
NREPEATS = *scalar*	Number of repeats of supplied sequences, or variate or a series of numbers to get power for multiples of a design; default 1
CARRYOVER = *string token*	Whether to include the carry-over term (yes, no); default no
CONTRASTTYPE = *string token*	Type of treatment contrasts if POLYNOMIAL and OWN parameters are unset (pairwise, control); default pair
ALPHALEVEL = *scalar*	Significance level at which to test each contrast, adjusted if necessary for multiplicity; default 0.05
DELTA = *scalar*	Tolerance for equivalence & non-inferiority tests; default 0.2231 i.e. log(1.25)
VARWITHIN = *scalar*	Variance of response within subjects; default 1
VARBETWEEN = *scalar*	Variance of response between subjects; default 1
NSIMULATIONS = *scalar*	Number of simulations; default 1000
SEED = *scalar*	Seed for random-number generator; default 0 i.e. continue from previous or use system clock
MONITOR = *string token*	What summary of power values to report every 50 simulations for each report chosen in PRINT option (minimum, mean, median, maximum); default * i.e. no monitoring

Parameters

SEQUENCES = *texts*, *variates* or *factors*	Sequence of levels of a single treatment factor; no default
POLYNOMIAL = *scalars*	Order of polynomials to represent the treatment factor; default * i.e. represent effects according to OWN parameter or CONTRASTTYPE option
OWN = *matrices*	Specific contrasts for the treatment factor; default * i.e. represent effects according to POLYNOMIAL parameter or CONTRASTTYPE option
MEANS = *variates*	Pattern of means for each treatment level for which to establish power; default * i.e. all zero
NONEQUALITY = *symmetric matrices* or *matrices*	Structure to save calculated power values for nonequality; default *
EQUIVALENCE = *symmetric matrices* or *matrices*	Structure to save calculated power values for equivalence; default *
NONINFERIORITY = *symmetric matrices* or *matrices*	Structure to save calculated power values for noninferiority; default *

SUPERIORITY = *symmetric matrices* or *matrices*
: Structure to save calculated power values for superiority; default *

Description

XOPOWER estimates by simulation the power of four types of statistical test commonly carried out on the results of cross-over trials. The most familiar is the test of non-equality; that is, testing whether there is evidence that two treatments are different. This is done by comparing against zero an estimate of a contrast between the levels of the treatment factor: the contrast here is simply the effect of one level minus the effect of the other, with no contribution from any other levels.

To establish the power of such a test you need to:

- set the NPERIODS option to the number of periods;
- set the VARWITHIN option to the within-subject variance;
- if the design is unbalanced, set the VARBETWEEN option to the between-subject variance;
- set the SEQUENCES parameter to the treatment factor defining the sequence of treatments received by each subject in turn (see below);
- set the NREPEATS option to the number of times the sequences are repeated on each subject if this is greater than one;
- set the MEANS parameter to the pattern of treatment means for which the power is to be estimated.

The procedure estimates the power of several contrasts of the treatment effects: by default, for each pairwise difference of treatment levels. Alternatively, the CONTRASTTYPE option can be set to control to request the power of the differences of each level with the reference level of the treatment factor. (The reference level is the first level by default, but other levels can be selected by using the REFERENCELEVEL option of the FACTOR directive.) Another possibility is to set the POLYNOMIAL parameter to the order of polynomial effects to be estimated for the treatment levels; orthogonal polynomials will be used, based on the marginal replication of the treatment levels. Finally, the OWN parameter can be set to a matrix that specifies comparisons between the treatment levels: the matrix must have one column for each treatment level and one row for each desired contrast. If either of POLYNOMIAL or OWN is set, the CONTRASTTYPE option is ignored.

The other three types of test are referred to as *equivalence*, *non-inferiority* and *superiority*. Equivalence requires the confidence limits for a contrast to lie within pre-assigned limits around zero; the limit is set with option DELTA, for which the default is 0.2231, corresponding to the natural logarithm of 1.25, a limit often used in analysis of log-Normal data from pharmacokinetic studies. Non-inferiority is a one-sided version of this, requiring the lower limit of the contrast to be greater than delta. The term *superiority* is sometimes used to refer to the test of non-equality (since interest lies only in establishing superiority of one treatment over another, though regulations insist on a two-sided test). However, here *superiority* is taken to require that the lower confidence limit of the contrast is greater than zero. The superiority test is therefore a one-sided version of the non-equality test. These three types of test are most likely to require pairwise or control contrasts; polynomial contrasts are probably not relevant, but own contrasts can be specified if required.

The SEQUENCES parameter should be set to a factor indicating the treatment level to be applied at each period for each patient: the ordering must be such that the sequence of levels for the first patient come first, then the sequence for the second patient, and so on. The NPERIODS option must be set to specify the number of periods. If preferred, the sequences can be input just as a text or variate structure containing the textual or numeric codes of the treatment, leaving the procedure to form the factor internally. If a series of sequences of factor levels is to be repeated entirely in the design, this can be specified by setting the NREPEATS option rather than having

to supply the series in full. Furthermore, you can supply a series of numbers for the NREPEATS option, to investigate the power of a series of multiples of a design.

XOPOWER can deal with only one treatment factor (though see below for carry-over). If a cross-over study has a more complex design, it may be possible to handle it by combining the treatment combinations into a single factor, and using the OWN parameter to specify the required constrasts (such as main effects and interactions).

The PRINT option controls which reports are displayed. The default settings, summary and nonequality, provide a summary of the design with a record of the option settings, and a report of the power for non-equality tests of each
contrast. You can also set contrasts to request a reminder of what contrasts have been specified. You can set any combination of nonequality, equivalence, noninferiority and superiority to have reports for these tests simultaneously; but these will all use the same settings of the MEANS parameter, so in practice it is unlikely that all four settings would be sensible.

By default, carry-over effects are ignored. However, if you set option CARRYOVER=yes, first-order carry-over effects are included in the model, and the power for the treatment contrasts will be adjusted accordingly. (The default setting is omit.) XOPOWER does not estimate the power for the carry-over effects themselves.

For a test of non-equality or superiority, the MEANS parameter should be set to a pattern of means for each level of the treatment factor. The location of these means does not affect power in the linear model assumed here, so for a two-level factor you can specify either two non-zero values representing the actual means, or a zero and a value representing the difference. In a clinical trial, this difference is usually referred to as the *clinically important difference*. Equivalence and non-inferiority tests are usually carried out under the assumption of equal effect, so the MEANS parameter can be left at its default setting (all zero). But power is sometimes calculated under the assumption of a small difference (compared to delta).

The ALPHALEVEL option allows you to set the Type I error for the tests that are being simulated. With a test of non-equality, this is the error for a two-sided test, but with superiority or non-inferiority it is one-sided. With an equivalence test, the error is that used in each of the two one-sided tests used to establish equivalence.

The VARWITHIN option must be set to an estimate of the within-subject variance expected in the trial. This is a critical part of any exercise evaluating power or calculating sample size before starting a study, and is often the most difficult unless good information is available on variability from previous trials. If the design is balanced in the sense that the treatment effect is orthogonal to the subject effect (as when each treatment is given once to each subject), then the within-subject variance is the only estimate of variance required. But if the design is unbalanced, you also need to provide an estimate of the between-subject variability, using the VARBETWEEN option.

You can control and monitor the simulation process to a limited extent. The NSIMULATIONS option sets the number of simulations to be carried out for each number of repeats specified by the NREPEATS option. The default is 1000, but experience shows that 500 simulations are usually adequate. The SEED option sets a seed for the random number generator used in the simulation, allowing you to ensure repeatability of results if you need this for documentation. The option MONITOR produces a display of progress every 50 simulations, to help you see when enough simulations have been carried out. You can set this to request the display to show the minimum, mean, median or maximum of the contrasts that are being calculated according. This display will show progress only for those tests whose display is requested by the PRINT option.

The NONEQUALITY, EQUIVALENCE, NONINFERIORITY and SUPERIORITY parameters allow the estimates of power to be saved in suitable structures. For pairwise contrasts, a symmetric matrix is used; for control, polynomial or own contrasts, a one-column matrix is used.

Warning: the procedure can generate warning messages when running the simulation, as a

result of problems within the REML process of fitting a model to the simulated data. If there are few messages, the estimated powers should hardly be affected; but if there are many, this is an indication that the specified design is not reliably analysed using the REML command.

Options: PRINT, NPERIODS, NREPEATS, CARRYOVER, CONTRASTTYPE, ALPHALEVEL, DELTA, VARWITHIN, VARBETWEEN, NSIMULATIONS, SEED, MONITOR.
Parameters: SEQUENCES, POLYNOMIAL, OWN, MEANS, NONEQUALITY, EQUIVALENCE, NONINFERIORITY, SUPERIORITY.

Method
An internal factor is set up to represent the treatment sequences, and taking account of requested repeats. For each iteration in a loop, a response is generated using the supplied setting of the MEANS parameter and including random Normal values to represent within- and between-subject variation. A mixed-effects model is fitted using REML including the following fixed effects:

- period,
- treatment,
- carry-over, if requested.

The random effect of Subjects is included, where the division of units between subjects is as implied by the setting of the SEQUENCES parameter and the PERIOD option.

If the NREPEATS option has more than one setting, the whole process is repeated in an outer loop, but ensuring that as much work as possible is done in the outer loop rather than in the inner loop over the simulations.

The results of each REML fit are extracted within the inner loop, and the results corresponding to each type of test (whether or not display or saving has been requested) are acumulated.

Action with RESTRICT
No structures should be restricted.

See also
Procedures: AFCARRYOVER, AGCROSSOVERLATIN, XOCATEGORIES, XOEFFICIENCY.
Genstat Reference Manual 1 Summary section on: Design of experiments.

YTRANSFORM

Estimates the parameter lambda of a single parameter transformation (D.M. Smith).

Options

TRANSFORM = *string token*	Type of transformation (power, modulus, foldedpower, GuerreroJohnson, Aranda1, Aranda2, powerlogit); default powe
METHOD = *string tokens*	Method of evaluating transformation parameter lambda (Atkinson, Andrews, BoxCox, Robust); default boxc
K = *scalar*	Cut-off value for robust method; default *
LOWER = *scalar*	Lower limit of range of lambda; default *
UPPER = *scalar*	Upper limit of range of lambda; default *
STEPLENGTH = *scalar*	Increment of lambda; default (UPPER - LOWER)/20
LAMBDA = *scalar*	Single value of lambda; default *
FVBOUND = *string token*	Replace illegal fitted values by the corresponding boundary values (no, yes); default no
GRAPHICS = *string token*	What sort of graphics to use (lineprinter, highresolution); default high
TERMS = *formula*	Terms of model

Parameters

Y = *variates*	Response variate
NBINOMIAL = *variates*	Denominator for a binomial variate
SAVE = *pointers*	Structures to save the output

Description

This procedure is for evaluating the "best" value of the transformation parameter (lambda) for a range of single parameter transformations. It offers four methods of evaluation and seven families of transformations. If a range of values of lambda is input (using the LOWER and UPPER options), plots are produced of either an F statistic or a log likelihood on the Y axis against lambda on the X axis. For the Atkinson and Andrews methods it is an F statistic, whereas for the Box-Cox and robust methods it is a log likelihood. The interval (of lambda) at which the plotted functions are evaluated can be controlled by the STEPLENGTH option. A list of methods is allowed and the plots have been arranged so that they are all produced on the same screen in order to make comparison easy. By default these are in high-resolution. Setting option GRAPHICS=lineprinter generates line-printer style (character) plots (one per page), and setting GRAPHICS=* suppresses the plots altogether. If a single value of lambda is input (using the LAMBDA option) no graphical display is produced.

The Y parameter must be set to specify the response variate i.e. the variate being considered for transformation. For a binomial distribution the NBINOMIAL parameter must also be set. The terms in the fitted model are specified by the TERMS option, which may be set to a formula or left unset to fit a model involving only a constant term. For reasons of scale invariance, as described in Schlesselman (1971), a constant term must be included in the model. The TRANSFORM option specifies which family of transformations is desired. It can take one of seven values. The setting power represents the power transformation family (Box & Cox 1964); modulus represents the modulus transformation family (John & Draper 1980); foldedpower the folded-power transformation family (Atkinson 1985); guerrerojohnson the Guerrero-Johnson (1982) transformation family; aranda1 and aranda2 the two Aranda-Ordaz (1981) transformation families; and powerlogit the power-logit (otherwise known as skewed logit) transformation family (Stukel 1988). The METHOD option details which methods of evaluating the transformation parameter (lambda) are required. It can be a list of from one to four values.

Four methods of evaluation are incorporated. These are the added variable method of Atkinson (1982), the added variable method of Andrews (1971), the maximum likelihood method of Box & Cox (1964), and a robust method due to Carroll (1980). For this latter method a scalar `K` is required. This value is the standard normal deviate value (*z*) at which the distribution changes from a standard normal to an exponential.

One problem with transforming data and then fitting models is that the fitted values (of the transformed data) can go out of the legal range. If the data are binomial, proportions of zero or one are replaced inside the procedure by 0.5/`NBINOMIAL` and 1 – 0.5/`NBINOMIAL` respectively. Conversely, when proportions are input directly in the `Y` variate, units with values less than or equal to zero or greater than or equal to one are ignored in the calculations. Option `FVBOUND` controls what happens in other circumstances when a fitted value goes outside the allowed range of the transformation. By default, no action is taken but, if `FVBOUND=yes`, illegal fitted values are replaced by the corresponding limiting values of the transformation.

The values of the F statistics or log likelihoods can be saved, with the associated values of lambda, using the `SAVE` parameter. This returns a pointer containing four elements. The first three of these are texts specifying, respectively, the transformation family (`SAVE[1]`, one value), the value of `FVBOUND` (`SAVE[2]`, one value) and the methods used (`SAVE[3]`, one to four values). The fourth element (`SAVE[4]`) is a matrix of results with dimensions (number of values of lambda evaluated × number of methods plus one). Column 1 of this matrix contains the evaluated values of lambda, column 2 has the values (F statistics or log likelihoods) for the first method requested, and so on for the other methods. If the option `LAMBDA` is used, this matrix has only one row.

Full details of the methodology implemented are given by Smith (2002).

Options: `TRANSFORM`, `METHOD`, `K`, `LOWER`, `UPPER`, `STEPLENGTH`, `LAMBDA`, `FVBOUND`, `GRAPHICS`, `TERMS`.
Parameters: `Y`, `NBINOMIAL`, `SAVE`.

Method

Much of the methodology implemented is based on that described and reviewed in Atkinson (1985), and Cook & Weisberg (1982). The four methods of evaluation are the added variable method of Atkinson (1982), the added variable method of Andrews (1971), the maximum likelihood method of Box & Cox (1964), and a robust method (based on maximum likelihood) due to Carroll (1980). The seven transformations are the power transformation of Box & Cox (1964), the modulus transformation of John & Draper (1980), the folded-power transformation (as expounded in Atkinson 1985), the Guerrero-Johnson (1982) transformation, the two transformations of Aranda-Ordaz (1981), and the power-logit (otherwise known as skewed logit) transformation of Stukel (1988). The log-likelihood produced for the Box & Cox method differs from that given by Box & Cox (1964), as they omit the constant term $N/2$. `YTRANSFORM` includes this for compatibility with Carroll's robust method, which collapses to Box & Cox's method as K becomes infinite.

Action with `RESTRICT`

If the `Y` variate is restricted, the analysis will use only the units not excluded by the restriction.

References

Andrews, D.F. (1971). A note on the selection of data transformations. *Biometrika*, **58**, 249-54.

Aranda-Ordaz, F.J. (1981). On two families of transformation to additivity for binary response data. *Biometrika*, **68**, 357-63.

Atkinson, A.C. (1982). Regression diagnostics, transformations and constructed variables (with discussion). *Journal of the Royal Statistical Society, Series B*, **44**, 1-36.

Atkinson, A.C. (1985). *Plots, Transformations and Regression*. Oxford University Press, Oxford.
Box, G.E.P. & Cox, D.R. (1964), An analysis of transformations (with discussion). *Journal of the Royal Statistical Society, Series B*, **26**, 211-46.
Carroll, R.J. (1980). A robust method for testing transformations to achieve approximate normality. *Journal of the Royal Statistical Society, Series B*, **42**, 71-78.
Cook, R.D. & Weisberg, S. (1982). *Residuals and Influence in Regression*. Chapman & Hall, New York.
Guerrero, V.M. & Johnson, R.A. (1982). Use of Box-Cox transformation with binary response models. *Biometrika*, **69**, 309-14.
John, J.A. & Draper, N.R. (1980). An alternative family of transformations. *Applied Statistics*, **29**, 190-97.
Schlesselman, J. (1971). Power families: a note on the Box and Cox transformation. *Journal of the Royal Statistical Society, Series B*, **33**, 307-311.
Smith, D.M. (2002). Computing single parameter transformations. *Communications in Statistics – Simulation and Computation*, **32**, 605-618.
Stukel, T.A. (1988). Generalized logistic models. *Journal of the American Statistical Association*, **83**, 426-31.

See also

Directive: CALCULATE.
Procedure: ABOXCOX.
Genstat Reference Manual 1 Summary sections on: Calculations and manipulation, Regression analysis.

Addresses of authors

A.E. Ainsley,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

C.J. Alexander,
Biomathematics & Statistics Scotland,
Scottish Crop Research Institute,
Invergowrie,
Dundee,
DD2 5DA, UK.

G.M. Arnold,
Biometrics Section,
IACR - Long Ashton Research Station,
Department of Agricultural Sciences,
University of Bristol,
Long Ashton,
Bristol BS18 9AF, UK.

D.B. Baird,
VSN (NZ) Limited,
8 Mariposa Crescent,
Aidanfield,
Christchurch 8025,
New Zealand

W. van den Berg,
Applied Plant Research,
Wageningen UR,
P.O. Box 430,
8200 AK Lelystad,
The Netherlands.

K.E. Bicknell,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

J.B. van Biezen,
Research Institute for Nature Management,
P.O. Box 9102,
6800 HB Arnhem,
The Netherlands.

M.P. Boer,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

C.J.F. ter Braak,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

P. Brain,
Pfizer Central Research
Sandwich,
Kent CT13 9NJ, UK.

N. Bratchell,
Pfizer Central Research
Sandwich,
Kent CT13 9NJ, UK.

J. de Bree,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

C.J. Brien,
University of South Australia,
GPO Box 2471,
Adelaide SA 5001,
Australia.

J.K.M. Brown,
Cereals Research Department,
John Innes Centre,
Norwich Research Park,
Norwich, Norfolk NR4 7UH, UK.

K. Brown,
Biomathematics & Statistics Scotland,
The King's Buildings,
Edinburgh EH9 3JZ, UK.

R.C. Butler,
Plant & Food Research,
Private Bag 4704,
Christchurch, 8140,
New Zealand.

V.M. Cave,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

S.J. Clark,
Rothamsted Research,
Harpenden, Herts AL5 2JQ, UK.

R. Cunningham,
Statistical Consulting Unit,
The Graduate School,
c/- MSI John Dedman Building,
Australian National University,
Canberra ACT 0200, Australia.

M.F. D'Antuono
Biometrics Unit,
Department of Agriculture,
South Perth,
Australia.

M.S. Dhanoa,
Biomathematics Group,
Institute of Grassland and Environmental Research, Aberystwyth Research Centre,
Plas Gogerddan,
Aberystwyth,
Dyfed SY23 3EB, UK.

the late P.G.N. Digby,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

P.J. Diggle,
Department of Mathematics and Statistics,
Fylde College,
Lancaster University,
Lancaster LA1 4YF, UK.

C. Donnelly,
Statistical Consulting Unit,
The Graduate School,
c/- MSI John Dedman Building,
Australian National University,
Canberra ACT 0200,
Australia.

E.I. Duff,
Biomathematics & Statistics Scotland,
Kings Buildings (JCMB),
Mayfield Road,
Edinburgh EH9 3JZ, UK.

P.H.C. Eilers,
Department of Biostatistics,
Erasmus Medical Center,
3015 GE Rotterdam,
The Netherlands.

D.A. Elston,
Biomathematics & Statistics Scotland,
Kings Buildings (JCMB),
Mayfield Road,
Edinburgh EH9 3JZ, UK.

M.F. Franklin,
Scottish Agricultural Statistics Service,
Rowett Research Institute,
Bucksburn,
Aberdeen AB2 9SB, UK.

S.A. Gezan,
Forest Resources and Conservation,
University of Florida,
Gainesville,
Florida 32611, USA.

A.I. Glaser,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

P.W. Goedhart,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

A.W. Gordon,
Biometrics Division,
Department of Agriculture and Rural Development (Northern Ireland),
Agriculture and Food Science Centre,
Newforge Lane,
Belfast BT9 5PX.

M.C. Hannah,
Agriculture Victoria,
Dairy Research Institute,
RMB 2460,
Ellinbank,
VIC 3820, Australia.

R.M. Harbord,
Department of Social Medicine,
University of Bristol,
Canynge Hall, Whiteladies Road,
Bristol BS8 2PR.

S.A. Harding,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

S.K. Haywood,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

D.I. Hedderley,
Plant and Food Research,
Private Bag 11600
Palmerston North, 4442,
New Zealand

C.F. Johnston,
Leukaemia Research Fund Centre for Clinical Epidemiology,
University of Leeds, UK.

P.D. Johnstone,
AgResearch,
Invermay Agricultural Centre,
Private Bag, Mosgiel,
New Zealand.

B. Jones,
Research Statistics Unit,
GlaxoSmithKline,
New Frontiers Science Park (South),
Third Avenue, Harlow CM19 5AW, UK.

A.F. Kane,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

Z. Karaman,
Limagrain Genetics Biometrics Unit,
Domaine de Mons, B.P. 115,
63203 Riom Cedex, France.

E.A.A. Kaul,
TNO Dept of Applied Statistics,
PO Box 155,
2600 AD Delft,
The Netherlands.

the late A. Keen,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

M.G. Kenward,
London School of Hygiene and Tropical Medicine, Keppel Street,
London WC1E 7HT, UK.

J.H. Klotz,
Department of Statistics,
University of Wisconsin-Madison,
1210 West Dayton Street,
Madison, Wisconsin 53703, USA.

P.W. Lane,
Statistical Research Unit,
GlaxoSmithKline,
New Frontiers Science Park (South),
Third Avenue, Harlow, CM19 5AW, UK.

P.J. Laycock,
School of Mathematics,
The University of Manchester,
Manchester M13 9PL, UK.

S.D. Langton,
Defra Environmental Observatory,
1-2 Peasholme Green,
York YO1 7PX, UK.

Y. Lee,
Statistics Department,
Seoul National University,
Sinromdong Kwanakgu,
Seoul City 151-742,
Korea.

P.K. Leech,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

the late R.P. Littlejohn,
AgResearch, Invermay Agricultural Centre,
Private Bag, Mosgiel,
New Zealand.

N.M. Maclaren,
University of Cambridge Computer Laboratory, Corn Exchange St,
Cambridge CB2 3QG, UK.

J.H. Maindonald,
College of Physical and Mathematical Sciences,
Australian National University,
Canberra ACT 0200,
Australia.

M. Malosetti,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

A.R.G. McLachlan,
Plant & Food Research,
Private Bag 11600,
Palmerston North, 4442,
New Zealand.

J.W. McNicol,
Biomathematics & Statistics Scotland,
Kings Buildings (JCMB),
Mayfield Road,
Edinburgh EH9 3JZ, UK.

K.L. Moore,
STATWOOD Partnership,
Invision House,
Wilbury Way,
Hitchen,
Herts SG4 0XE, UK.

A.W.A. Murray,
Central Science Laboratory,
Sand Hutton,
York
YO41 1LZ, UK.

D.A. Murray,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

the late J.A. Nelder
Department of Mathematics,
Imperial College,
180 Queen's Gate,
London SW7 2BZ, UK.

M. Noh,
Division of Mathematical Sciences,
Pukyong National University,
Busan 608-737,
Korea.

J. Ollerton,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

M.E. O'Neill,
Statistical Advisory & Training Service Pty Ltd
PO Box 939,
Tumut,
NSW 2720
Australia.

B.M. Parker,
University of Southampton,
Highfield Campus
Southampton
SO17 1BJ, UK

M.W. Patefield,
Department of Mathematics and Statistics,
Whiteknights,
PO Box 220,
Reading RG6 6AX, UK.

R.W. Payne,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

K. Phelps,
Biometrics Department,
Horticulture Research International,
Wellesbourne,
Warwick CV35 9EF, UK.

J.F. Potter,
AgResearch,
Grasslands Research Centre,
Tennent Drive,
Private Bag 11008,
Palmerston North, 4442,
New Zealand.

R.F.A. Poultney,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

K. Punyawaew,
Biosci Co., Ltd.
Thailand

R.J. Reader,
Biometrics Department,
Horticulture Research International,
Wellesbourne,
Warwick CV35 9EF, UK.

M.S. Ridout,
Mathematics Institute,
Cornwallis Building,
University of Kent at Canterbury,
Canterbury,
Kent
CT2 7NF, UK.

D.M. Roberts,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

A.J. Rook,
Biomathematics Group,
Institute for Grassland and Environmental Research,
North Wyke,
Okehampton,
Devon EX20 2SB, UK.

P.J. Rowley,
School of Mathematics,
The University of Manchester,
Manchester M13 9PL, UK.

B.S. Rowlingson,
Department of Mathematics and Statistics,
Fylde College,
Lancaster University,
Lancaster LA1 4YF, UK.

K. Ryder,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

L.H. Schmitt,
School of Anatomy & Human Biology M309,
The University of Western Australia,
35 Stirling Highway,
Crawley, WA 6009,
Australia.

Eric D. Schoen,
TNO Dept of Applied Statistics,
PO Box 155,
2600 AD Delft,
The Netherlands.

S. Senn
Competence Center for
Methodology and Statistics,
1AB Rue Thomas Edison,
L-1445 Strassen. Luxembourg.

the late H.R. Simpson,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

D.M. Smith,
Biostatistics, Novartis Pharmaceuticals,
One Health Plaza, Room 419/2113
East Hanover, NJ 07936, USA.

M.F. Smith,
Agricultural Research Council,
Biometry Unit, Private Bag X519,
Silverton 0127, RSA.

E. Stephens
Biometrics Section,
IACR - Long Ashton Research Station,
Department of Agricultural Sciences,
University of Bristol, Long Ashton,
Bristol BS18 9AF, UK.

R.D. Stern,
Statistical Services Centre,
The University of Reading,
Harry Pitt Building,
Whiteknights Road,
Reading RG6 6FN, UK.

R.A. Sutherland,
Biometrics Department,
Horticulture Research International,
Wellesbourne,
Warwick CV35 9EF, UK.

M. Talbot,
Biomathematics & Statistics Scotland,
The King's Buildings,
Edinburgh, EH9 3JZ, UK

D. Tandy,
Biomathematics Group,
Institute of Grassland and Environmental
Research, Aberystwyth Research Centre,
Plas Gogerddan, Aberystwyth,
Dyfed SY23 3EB, UK.

J.T.N.M. Thissen,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

A.D. Todd,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

G. Tunnicliffe Wilson
Mathematics Department,
Lancaster University,
Lancaster,
Lancs LA1 4YL, UK.

H. Turner,
Biometrics Section,
IACR - Long Ashton Research Station,
Department of Agricultural Sciences,
University of Bristol, Long Ashton,
Bristol BS18 9AF, UK.

F.A. van Eeuwijk,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

H. van der Voet,
Biometris, Wageningen UR,
P.O. Box 100,
6700 AC Wageningen,
The Netherlands.

I. Wakeling,
Institute of Food Research,
Earley Gate, Whiteknights Road,
Reading, Berkshire RG6 6BZ, UK.

R. Webster,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

S.J. Welham,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

D.C. van der Werf,
DLO-Institute for Forestry and Nature
Research, P.O. Box 23,
6700 AA Wageningen,
The Netherlands.

R.P. White,
Rothamsted Research,
Harpenden,
Herts AL5 2JQ, UK.

J.T. Wood,
Statistical Consulting Unit,
Australian National University,
Canberra, ACT 0200,
Australia.

Z, Zhang,
VSN International,
2 Amberside,
Wood Lane,
Hemel Hempstead,
Herts HP2 4TP, UK.

Index