

Issue No. 14

October 1984

The
GENSTAT
Newsletter

NAG
NUMERICAL
ALGORITHMS
GROUP



Editors

R W Payne
Rothamsted Experimental Station
Harpenden
Hertfordshire
AL5 2JQ

M G Richardson
NAG Central Office
Mayfield House
Oxford
OX2 7DE

Printed and produced by the Numerical Algorithms Group

©NAG Limited 1985

All rights reserved. No part of this newsletter may be reproduced, transcribed, stored in a retrieval system, translated into any language or computer language or transmitted in any form or by any means, electronic, mechanical, photocopied recording or otherwise, without the prior permission of the copyright owner.

NAG is a trademark of the Numerical Algorithms Group.

The views expressed in contributed articles are not necessarily those of the publishers.

Rw Payne

GENSTAT NEWSLETTER
Issue No. 14

© Numerical Algorithms Group Ltd

NP1032

October 1984

Contents

	Page
1. Editorial	3
2. Status Report	3
3. Genstat - The Future	9
4. Genstat - An Easier Way To Learn	10
5. Genstat: Origins and Prospects	12
6. Dendrograms and Ziggurats	14
7. Drawing Pretty Dendrograms	18
8. Using Genstat to Fit Complex Association Models to Contingency Tables	27
9. Manual and Notice Board Amendments	32
10. Notices	
10a. 4th Genstat Conference: September 1985	36
10b. A Genstat Primer	37

Enclosures

Genstat Newsletter Order Form
Genstat Primer Order Form
Genstat Reference Summary Order Form
4th Genstat Conference Registration and Payments Form
Genstat 4.04 Manual Amendment List

Published Twice Yearly by
Rothamsted Experimental Station Statistics Department
and the Numerical Algorithms Group Limited

Printed June 1985

Editorial

This is the first issue of the Newsletter published since John Nelder's retirement from Rothamsted and we are sure that all Genstat users will want to join us in wishing him a very long, enjoyable and active retirement.

Dr. Nelder's recollections of some of the history and development of Genstat appear in this issue, as well as a prognosis by John Gower, the new head of the Rothamsted Biomathematics Division.

A free copy of the new Genstat Reference Summary is enclosed with this Newsletter. The editors have already found this a useful aide-memoir and hope that it will also prove so to other Genstat users. Additional copies may be ordered from NAG Central Office. (See enclosed order form).

The content of this issue of the Newsletter was mainly provided from Rothamsted. To guarantee a lively and relevant publication, we urgently require submissions from Genstat users, so once again we stress our urgent need for articles on any aspect of Genstat.

For some years, NAG has produced the Genstat Newsletter at a loss: following a general review of the pricing of all NAG Newsletters, an increase in the basic subscription price was agreed, to £8 per year, rather than the £6 which would have covered inflation since the previous increase. In addition, for customers requiring an invoice, an invoicing charge of £3 per transaction will be made. To enable cash customers to order the Newsletter by airmail, a fixed airmail rate of £10 per year will replace the previous 'postage charged at cost' arrangement.

Finally, may we remind readers that the next Genstat Conference will take place, at York, from 23 to 26 September 1985. Information and a registration form are included in this issue.

Status Report

*M.G. Richardson
NAG Central Office
Mayfield House
256 Banbury Road
Oxford OX2 7DE
United Kingdom*

*H.R. Simpson
Rothamsted Experimental Station
Harpenden
Hertfordshire AL5 2JQ
United Kingdom*

A number of errors were found in Release 4.04A and corrections for these were generated, to form 4.04B, which was released to AFRC users in March 1984 and has been sent out to 82 other VAX sites since.

The Prime version of 4.04B was implemented at Rothamsted, shortly before we lost our 550, and distributed to the Prime sites. Responsibility for this version has now been assumed by J.K. Wight of the Poultry Research Centre at Roslin, near Edinburgh.

A version of 4.04A is available for GEC 4100 series computers and 4.04B has been implemented for the following systems and distributed to users:

Honeywell computers under both GCOS and Multics, IBM (under OS and related systems), ICL 2980, Telefunken and Univac. The CDC and DEC-10 versions are expected soon.

The number of Genstat sites continues to grow steadily, with 65 new sites over the last twelve months. Although a few sites inevitably terminate their Genstat licences, the total number of sites has grown at a rate of more than one per week. At the time of writing (27 February 1985) the total number of sites using Genstat, or with it on order, was 299.

The number of sites with Genstat on several computers is also growing: there are now 22 such sites with the package mounted on up to 8 processors.

A full list of Genstat sites, classified by country and machine range, is given below. Anyone wishing to contact another site should first approach the Genstat Co-ordinator at NAG Central Office.

AUSTRALIA

Burroughs

Public Service Board of New South Wales
University of Tasmania Computing Centre

CDC

CSIRO Division of Human Nutrition
CSIRONET
CSIRONET
South Australian Department of Agriculture
University of Sydney

DEC Systems 10 & 20

James Cook University of North Queensland
University of New England
University of Queensland
WARCC

DEC Vax-11

Anglo-Australian Observatory
Applied Computing
Australian Graduate School of Management
Australian National University
CSIRO (DMS)
CSIRO Cotton Research Unit
CSIRO Division of Plant Industry
CSIRO Marine Laboratories
CSR Ltd
Department of Agriculture Victoria
Department of Primary Industry, Qld
Macquarie University
Roseworthy Agricultural College
University of Adelaide
University of Melbourne
Woods and Forest Department Adelaide

IBM & similar

Agrisearch Services Pty Ltd
Australian Commonwealth Department of Health
SEQEB
Uncle Ben's of Australia

PRIME

Australian National Parks & Wildlife Svs
Department of Agriculture - Western Australia
Flinders University of South Australia
Institute of Family Studies
New South Wales Department of Agriculture

Sperry 1100

Australian National University

Tasmanian Treasury

Other machines

University of Adelaide

AUSTRIA

DEC Vax-11

Universitaet in Salzburg

BELGIUM

DEC Vax-11

Solvay AB

BRAZIL

IBM & similar

DMQ/EMBRAPA

CANADA

DEC Vax-11

Agriculture Canada
Ontario Institute for Studies in Education

IBM & similar

McGill University
University of Toronto

COLOMBIA

IBM & similar

CIAT

DENMARK

CDC

Aarhus University
Rigshospitalet, Data - Dept
Universitetet i Aalborg

DEC Vax-11

Danmarks Geologiske Undersoegelse

IBM & similar

Technical University of Denmark

ICL 2900

NOVO INDUSTRIA/S

PRIME

Cancerregisteret
Handelshoejskolen i Koebenhavn

Sperry 1100

Danish Meteorological Institute
Datacenteret vd Odense Universitetscenter
ELSAM
RECKU

EIRE

DEC Vax-11

Agricultural Institute
IIRS
NIHE Limerick
University College Cork

IBM & similar

University College Dublin

ICL 2900

Guinness Ireland Limited

FRANCE

CDC

SEP/CNES (Toulouse)

DEC Vax-11

Centre International Recherches Dermatologiques
UTAC

HB-CII Iris

I.N.S.E.R.M.
IFREMER

Honeywell Series 60 & similar

C.I.C.T.
Centre National de Recherches Zootechniques

IBM & similar

Aerospatiale
Centre 'Air'
Universite de Paris Sud

PRIME

CESA

Sperry 1100

ATOCHEM
CELAR
SEP.T (Rungis)
Universite de Paris Sud

West GERMANY

Burroughs

Universitaet Duisburg

CDC

Freie Universitaet Berlin

ICL 2900

Universitaet Hohenheim

Modcomp

Institut fuer Numerische Statistik

Siemens BS2000

Bundessortenamt
Universitaet Duesseldorf

Telefunken TR440

Universitaet Osnabrueck

GREECE

CDC

National Technical University of Athens

HONG KONG

Sperry 1100

Universities and Polytechnic Computer Centre

ICELAND

DEC Vax-11

University of Iceland

INDIA

DEC Vax-11

ICRISAT

ITALY

Burroughs

Universita di Genova

DEC Vax-11

I.A.T.A.-C.N.R. (Firenze)
Universita di Pavia

HB-CII Iris

Universita degli Studi Firenze

IBM & similar

Istituto per le Applicazioni del Calcolo
Universita degli Studi di Padova

Sperry 1100

Universita degli Studi di Napoli
Universita di Roma

JAPAN

IBM & similar

Fuyo Data Processing & Systems Development Ltd
Institute of Statistical Mathematics
University of Tokyo Hospital

KUWAIT

Sperry 1100

Kuwait University

MALAYSIA

Sperry 1100

Universiti Pertanian Malaysia

MEXICO

Burroughs

IIMAS - UNAM

DEC Vax-11

CIMMYT

The NETHERLANDS

Burroughs

Gist Brocades NV

CDC

Centrum voor Data-Analyse, RU Utrecht
IWIS-TNO
Landinrichtingsdienst

DEC Systems 10 & 20

Agricultural University of Wageningen

DEC Vax-11

Agricultural University of Wageningen
CIVO Institutes TNO
ESTEC
KIWA N.V.
Ri Vg Mh
TFDL
TNO Institute for Applied Geoscience

PRIME

Provinciaal Bestuur van Drenthe

Other machines

Old Dutch Grain Casualty & Indemnity Ltd

NEW ZEALAND

Burroughs

University of Otago

DEC Vax-11

DSIR
Lincoln College
New Zealand Forest Service
University of Waikato
Wool Research Organisation of New Zealand (Inc)

IBM & similar

Ministry of Works and Development
University of Auckland
Victoria University of Wellington

ICL 2900

Trentham Computer Centre

PRIME

Massey University
Ministry of Agriculture and Fisheries

NIGERIA

CDC

Ahmadu Bello University

DEC Vax-11

IITA

NORWAY

CDC

University of Tromsøe

DEC Vax-11

Norsk Regnesentral
Papirindustriens Forskningsinstitutt

PAPUA NEW GUINEA

ICL 2900

National Computer Centre

PORTUGAL

Other machines

LNETI

SENEGAL

IBM & similar

CRODT

SOUTH AFRICA

Burroughs

Division of Biometric and Datametric Services
University of South Africa

IBM & similar

Premier Group Limited

PRIME

University of Fort Hare

Sperry 1100

University of Cape Town
University of Natal Computer Services

SWEDEN

DEC Vax-11

Foersvarets Forskningsanstalt

IBM & similar

Stockholm Data Center QZ
Uppsala Universitets Datacentral

PRIME

Svenska Geologiska AB

Sperry 1100

Universitetet i Lund

SYRIA

DEC Vax-11

CERS Syria

TAIWAN R.O.C.

IBM & similar

Pig Research Institute

TURKEY

IBM & similar

Ege Universitesi

UNITED KINGDOM

Burroughs

University of Warwick

CDC

British-American Tobacco Co Ltd
Imperial College
Transport and Road Research Laboratory
UMRCC

University of Bradford

DEC Systems 10 & 20

Clinical Research Centre
Dundee College of Technology
Hatfield Polytechnic
Liverpool Polytechnic
National Institute for Medical Research
Polytechnic of Central London
Polytechnic of South Bank
Trent Polytechnic
University College of North Wales
University of Essex
University of York

DEC Vax-11

A. U. W. E. (S)
BP International Ltd (Research Centre)
Beecham Pharmaceuticals
Brighton Polytechnic
Cambridgeshire C A T
Campden Food Preservation Research Association
City of London Polytechnic
DOE Building Research Establishment
Edinburgh University Linguistics Department
FBC Ltd
Forestry Commission (Edinburgh)
G D Searle & Co Ltd
Hunting Engineering Ltd
Janssen Pharmaceutical Ltd
Johnson Matthey Research Centre
Kingston Polytechnic
MRC TB & Chest Disease Unit
Millward Brown Market Research
NERC Computing Service
National Radiological Protection Board
Oxford University Computing Service
Pfizer Central Research
Queen's University of Belfast (Agricultural Biom)
Queen's University of Belfast
Rank Xerox Information Systems
Research Services Ltd
Rothamsted Experimental Station
Shell Sittingbourne Research Centre
Silsoe College
Smith Kline & French Research Ltd
Unilever Research
University of Bristol
University of Edinburgh (CAST)
University of Lancaster
University of St Andrews
University of Stirling
University of Sussex
Water Research Centre
West Midlands Gas
West Sussex County Council

GEC 4000/4100

MRC Labs, Carshalton

Gould SEL

CDE Porton Down

Harris

Coventry (Lanchester) Polytechnic

Unilever Research

University of Aston in Birmingham

Honeywell Series 60 & similar

Aberdeen University
Associated Octel
Brunel University
Loughborough University of Technology
NERC Computing Service
University College, Cardiff
University of Birmingham

Honeywell Sigma 6

H M Government Communications Centre
Ministry of Defence (PE)

IBM & similar

AERE Harwell
AWRE
BUPA Services
British Gas Corporation
British Marine Technology Limited
Cadbury Schweppes plc
Huntingdon Research Centre
Imperial Tobacco Limited
Meteorological Office
National Foundation for Educational Research
Perkins Engines
Rutherford & Appleton Laboratories
Scottish Office
Shell Research Ltd Thornton
Shell Sittingbourne Research Centre
Smith, Kline & French Laboratories Ltd
University of Cambridge
University of Leeds
University of Liverpool
University of London Computer Centre
University of Newcastle upon Tyne
University of Reading
Wellcome Foundation Ltd

ICL 1900 & similar

MAFF(ADAS)
Meat and Livestock Commission
Northern Regional Health Authority
Thames Water Authority
University of Hull
Welsh Water Authority
Wessex Regional Health Authority
Yorkshire Regional Health Authority

ICL 2900

Business Statistics Office
Central Electricity Generating Board
Eastern Gas
Huddersfield Polytechnic
Mersey Regional Health Authority
North East Gas Board
North West Water
Oxford University Computing Service
Portsmouth Polytechnic
Queen Mary College
S W U R C C
Severn Trent Water Authority
University of Edinburgh (CAST)

University of Lancaster
University of Nottingham
University of Southampton
West Midlands County Council
West Midlands RHA
West Midlands RHA (North Staffordshire Hospitals)

PRIME

Bristol Polytechnic
DOE Building Research Establishment
Forestry Commission (Farnham)
Glaxo Group Research Ltd
Gloucestershire CAT
Institute of Occupational Medicine
Leatherhead Food Research Association
Ministry of Agriculture Fisheries and Food
Napier College
Nature Conservancy Council
North East London Polytechnic
Plymouth Polytechnic
Poultry Research Centre
RSPB
Royal Botanic Gardens
Royal Institute of British Architects
Thames Polytechnic
University of Sheffield
University of Surrey
Welding Institute

Perkin Elmer

DHSS
Life Science Research
Motor Industry Research Association

Sperry 1100

BP International Ltd Information Systems Services
Shell UK

Other machines

Dr G W Winn
MicroAPL Ltd
Oxford University Department of Forestry
Worcester College of Higher Education

UNITED STATES

CDC

College of Business, Florida State University
Colorado State University
Michigan State University

DEC Vax-11

New York State Health Department
Union College
University of California, Riverside

Honeywell Series 60 & similar

Honeywell PMSD

Modcomp

United States Department of Agriculture (ARS)

PRIME

Drexel University
New York State Agricultural Experiment Station

Sperry 1100

State University of New York at Albany

YUGOSLAVIA

Sperry 1100

University of Zagreb

ZIMBABWE

Perkin Elmer

Zimbabwe Biometrics Bureau
Zimbabwe Scientific Computing Centre

Genstat – The Future

*J.C. Gower
Statistics Department
Rothamsted Experimental Station
Harpenden
Hertfordshire AL5 2JQ
United Kingdom*

John Nelder retired from Rothamsted on 31 October 1984. This has led some to contemplate the future of Genstat. They need have no fear that it will be abandoned, will attract no further development or no support. The Statistics Department at Rothamsted has been much involved with statistical computing since 1954 when Frank Yates, its then Head, obtained an early prototype computer – the Elliott/NRDC 401. Computer programs were rapidly developed for assisting with the Department's main burden of computing, arising from statistical consulting and associated methodological research problems. The early computers would support only small programs, necessarily of a specialised nature. However, it was soon recognised that the way forward was to write increasingly general programs, which incidentally encouraged the unification of statistical methodology. By the mid-sixties Rothamsted had developed a general survey program, a small suite of programs to handle the main classes of experimental designs, an early version of the Maximum Likelihood Program and CLASP for cluster analysis: we were beginning to explore how these could be accommodated within a single system and had recognised that a basic set of data-structures was the essential unifying feature. Genstat may be viewed as the next step in this development.

Over the past 16 years, Genstat itself has changed enormously, by the addition of new statistical features, developments in the language, the provision of basic graphics etc. Many of these extensions have been of an ad hoc nature, some opportunistic. The upshot is a language with many inconsistencies which make it difficult to learn, teach and remember – hence the Genstat 5 revision, whose specification is complete and whose implementation is in progress. The PROCEDURE facility in particular, with its similarity to the specification of directives, will make the new Genstat even easier to extend than is the current version and, hence, much increase the power of the language in providing new statistical facilities. In association with this revision, we are taking the opportunity to catch up with advances in computing. A properly interactive version of Genstat is being prepared and integrated, flexible, portable graphics are being investigated. The next stage will be a menu-driven version.

The justification for all this is that, at the end of it all, we will be able to offer a better version to Rothamsted scientists and to other scientists in the Agricultural and Food Research Council. However, all users of Genstat, backed up by the support of NAG, will reap the benefits. I hope this will reassure those that need it, that Genstat has a sound future. Looking further ahead, we can contemplate integrating artificial intelligence features into the system. Perhaps we shall then re-establish our links with John Nelder, for it is in just this area that John is engaged, in his very active 'retirement'.

Genstat – An Easier Way To Learn

*T.J. Dixon
Statistics Department
Rothamsted Experimental Station
Harpenden
Hertfordshire AL5 2JQ
United Kingdom*

[Based on an article appearing in the Professional Statistician (August/September 1984)]

There are numerous comparisons of statistical packages in the statistical computing literature. When Genstat is mentioned, it invariably gets excellent marks for the range and flexibility of the analyses it can perform. From a special issue of the Professional Statistician devoted to statistical computing (June/July 1983) the following comments were extracted:

'Genstat is not just a statistical package but is, in effect, a programming language.'

'Genstat was provided to try to mop up those analyses which could not be handled through the interactive packages.'

'This [Genstat] certainly permits the statistician to do a great deal with a single software item.'

'Genstat is a statistical programming language of considerable power.'

These highlight the fact that Genstat not only provides high quality algorithms, to deal with standard statistical problems, but also has a programming language in which the user can develop his own analysis or try out new methods published in the statistical literature.

At the same time, there are often references to the difficulty of learning to use Genstat.

'Genstat has quite a steep learning curve to get even a quite trivial problem running.'

'[A Genstat user requires] a substantial investment in time both in learning the programming language and in actually writing programs to analyse his data.'

'Graduate statistics students learn to master it [Genstat] but it is not for the naive statistical user.'

The comparison between Genstat and many other statistical programs is similar to that between Fortran (or Pascal) and Basic. Genstat and Fortran provide the ability to do virtually any task efficiently, once a familiarity with the language is established. Other packages and Basic are easier to learn and facilitate routine tasks but are stretched to do anything out of the ordinary.

The adverse comments about learning Genstat often stem from the fact that, until recently, the main way of learning to use the program was by reading the manual. To learn from a manual two inches thick, which covers all aspects of a comprehensive program, is ridiculously complicated – exactly like writing Fortran programs using only the syntactic definition.

In the past, having read the manual, the user would then punch Genstat statements and submit a batch job. He would then get his output some time later and, like any beginner in any language, find it would contain errors. Gradually, he would iron out the mistakes and the program would work. If he used Genstat frequently he would develop enough expertise to get most programs to run first time. Asking advice from an experienced user of Genstat, if available, could speed up the process but, nevertheless, many users became discouraged.

The alternative to starting directly with the manual was an introductory Genstat course. These consisted of a long lecture and a single practical in each of six half-days. Lectures provided a much easier introduction than the manual but the problem of slow batch turnaround meant that there was, often, insufficient time in the three days to get many programs working and learn by the experience, even with the help of the Genstat experts giving the course.

Nowadays, with the modern computers available, particularly those with paged operating systems, there is little need to differentiate between small and large programs. As a consequence, most machines can turn round large batch jobs quickly. They can also provide interactive facilities for large programs, such as Genstat, and the recent releases of Genstat have been adapted for interactive running. Along with new introductory material for learning Genstat, this means there are now several much simpler ways of becoming familiar with Genstat.

One starting point is the series of video-tapes 'Genstat: A Preview'. The first tape introduces the language by using Genstat to analyse a simple glasshouse experiment with analysis of variance. The second tape deals with regression analysis of some real data, while the third presents the views of statisticians from all over the world on the ways in which Genstat is useful in their work. The tapes are not for teaching how to write Genstat programs but give a flavour of the way a program is constructed.

A follow up to the video tapes, or an alternative starting point, is the book 'An Introduction to Genstat', which provides an elementary description, introducing ideas only as they are needed. After chapters dealing with the handling of data and graphical facilities, the reader soon encounters regression analysis and goes on to cover analysis of variance and the use of tables. The final chapters encourage the reader to explore some of the other areas of Genstat not covered in detail. The use of well chosen examples and explanation of the statistical concepts involved provide all the information required to analyse straightforward data. Problems are given, for the reader to try out his new-found expertise, and solutions are given at the end of the book.

For a more in-depth introduction to Genstat, the three day course has been redesigned to take advantage of fast, modern computing facilities by interleaving short lectures and practical classes. The student can then master each idea as soon as it is introduced, thus seeing his mistakes straightaway, while his train of thought is still on the subject. By the end of the course the majority of students are quite confidently writing programs. The only thing that may hold them back is their own knowledge of the statistical techniques they now have at their fingertips. Over the three days the student is taught to handle data within the program, produce graphs and histograms, tabulate data, analyse designed experiments using Anova and perform regression analyses on continuous and grouped data. Courses can be provided for new Genstat sites, whilst established sites are encouraged to give their own courses, using material supplied by Rothamsted and tailoring courses to suit local requirements.

To summarise, Genstat is now easier to learn and more convenient to use. The benefits of a flexible statistical language are great and new facilities are being added all the time. Some statisticians have always dismissed Genstat as being too complex to be worth the effort, but this is no longer true (if it ever was). An increasing number of statisticians are finding Genstat invaluable in their work. The Genstat Newsletter and Genstat conference give numerous accounts of how Genstat has enabled them to find appropriate solutions to their problems.

References

- Alvey, N., Galwey, N., and Lane, P. (1982) *An Introduction to Genstat*. Academic Press, London.
- Alvey, N., Lane, P. and Nelder, J. (1982) *Genstat: A Preview*. University of London Audio Visual Centre, London.
- The Professional Statistician (1983) (Eds. Huson and Tarling) Institute of Statisticians, Bury St Edmunds.

Genstat: Origins and Prospects

J.A. Nelder
Department of Mathematics
Imperial College
London SW7 2BZ
United Kingdom

Once upon a time there was a first-generation computer called the Elliott 401, the programming of which seemed of such complexity that I was happy to hand over my problems for others to put on the machine. Then came Orion and with it (Extended) Mercury Autocode, a first-generation high-level language, and we began to write programs at Wellesbourne for our statistical analyses. Space was limited, speed was slow, and the operating system not very sophisticated. Nonetheless it would, a bit grudgingly perhaps, allow you to join separate programs together in a sequence and pass the output from one to the next. So we joined together input routines, a program for calculating derived variates, and another for the analysis of variance to produce a primitive system. So far as I know nobody ever used our modules except our statistical group at The National Vegetable Research Station. Getting a job into the machine was not an easy business; paper-tape readers were more robust than card readers, but paper tape had to be edited by splicing and individual hole-punching. Thus you chose between this and the ease of replacing cards, only to have them chewed up by the card reader if there was the least blemish on their edges.

In 1965 came the opportunity to spend a year in Adelaide, divided between the CSIRO Division of Mathematical Statistics, then housed in the University, and the Waite Institute, which, although part of the University, was some 7 miles away to the south. The University had acquired a CDC3300 computer which used Fortran, so I set off on the *Oriana* to Perth, armed with the Fortran manual to study en route. After a journey by rail across the Nullarbor, during which it rained unceasingly, we arrived in Adelaide on a cold winter's afternoon to be met by Graham Wilkinson and taken to the house we were to live in.

First acquaintance with the CDC3300 showed that here was something altogether more powerful than anything I had met before. The card reader was unbelievably robust – you could crumple a card in your hand, smooth it out again, and the reader would accept it happily and read it accurately. The Fortran had encode/decode plus buffer in/out, which meant one could encode formats dynamically and then use these to encode the output. What a time it took for this to become standard!

It was in this environment that I began, with Graham Wilkinson's enthusiastic support, to write the first versions of Genstat. We used the data matrix as the main data structure and we included various operations on data matrices as well as vector arithmetic, regression, and, as a *piece de resistance*, Graham's algorithmic version of my ideas for the analysis of generally balanced designs. We made all kinds of errors of design and implementation, which were gradually removed; for instance, in writing the scatter-plot routine it never occurred to me that anyone would try to plot a variable *all* of whose values were missing. Yet the third user did just that, and the scaling routine crashed. We had no debugging tools, except hand-inserted write statements, and only a crude overlaying system for segmenting the program. However, people gradually began to use the program and Graham's anova algorithm scored a spectacular success when some ingenious experimenter brought along data from a Graeco-Latin square design with a balanced incomplete block design superimposed on the plots. I am sure that some statisticians could not have worked out the form of the analysis-of-variance table correctly and it was fascinating to see an instance where an interaction term came out in a higher stratum than one of its corresponding main effects. Work went on through the summer, when the computer room was the only cool place in the building, for

which the architects had designed a system that ensured that every room had a built-in supply of hot air from outside. On a day with a howling north wind from the desert, when the temperature reached 104° F, I remember going home early.

On my return to England the environment changed abruptly. The Orion had just acquired a Fortran compiler but, when demonstrated by its originators, it failed to work and it was quite clear that large-scale program development in Fortran was not going to work. Furthermore, as the result of contact with the Rothamsted statisticians (I was still at NVRS) it became clear to me that systems based on the data matrix were simply not flexible enough for a good statistical package, which must support multi-stage analyses: it was not enough, for example, to form tables, you had to be able to manipulate them. Thus, when I moved to Rothamsted in 1968, Genstat underwent a complete re-design, to emerge as a system close to its present form. The data structures were extended to include three kinds of matrix, as well as tables, and vectors of pointers also appeared. The (fairly!) systematic use of lists as parameters in directives and expressions gave a potentially powerful interpretive language. The Wilkinson algorithm for the analysis of generally balanced designs was refined and remained a cornerstone of the system; in the 1970s following Wedderburn's and my development of generalized linear models, the regression section was updated to include GLMs: the Australian connection was resumed and Ron Baxter played a major role here.

For years the table manipulation facilities of Genstat were largely unused but, about ten years ago, I began to hear people at meetings talking about operations on tables, a splendid new idea! I still find it surprising that so many for so long had been content with packages which, although they allow formation of tables, allow nothing to be done with them afterwards except for display. GLMs have also taken time to become established but now seem to be increasingly used. A major, more recent, extension of facilities has been for time-series analysis, the result of a fruitful collaboration between Granville Tunnicliffe Wilson and Peter Lane.

Because Genstat 4 is now well established and distributed, it must, therefore, be out of date. So much has happened to hardware, operating systems and languages in the last ten years that a major update has become inevitable. I am leaving Rothamsted just at the time when the new Genstat 5 is beginning to jell, and the prospects look exciting. Much tidying-up of syntax, names, etc. is possible, modern control structures can be introduced and the data structures expanded to embrace the compilation phase as well as the execution phase; as much as possible of the tried and tested algorithmic code is being retained but the image presented to the user will be simpler, more consistent and potentially more powerful. I wish the team all success in these future developments; what they are doing would have been literally impossible in the environment of twenty years ago when we took the first steps.

Dendrograms and Ziggurats

*P.G.N. Digby
Statistics Department
Rothamsted Experimental Station
Harpenden
Hertfordshire AL5 2JQ
United Kingdom*

When n units are hierarchically classified into groups, the results are often displayed as a dendrogram (see the previous article). The order of the units in the dendrogram is obviously constrained by the hierarchical grouping; what many users of such methods may not realise is that there are 2^{n+1} different admissible orderings with which the dendrogram can be drawn. This is because, for each of the $n-1$ merges, either group may be placed above the other in the order. The order produced by the HIERARCHY directive, which may be saved by the user, is chosen in a way that may not be particularly sensible. A variate giving an alternative order of units, defined in terms of ziggurats (Critchley, 1983) may be obtained from the macro `z1gord(1)` given below. This has the same identifier, `c1ord`, as was used in the previous program, and so could be used there.

Ziggurats are defined recursively: a ziggurat of degree $k+1$ is an initial join of two ziggurats of degree k to form one group, followed by a series of joins of ziggurats of degree k to this group. Initially, all the units are ziggurats of degree zero; a degree-one ziggurat consists of an initial merge of two units (degree-zero ziggurats) followed by any repeated joining of individual units to this group. Thus, the first stage is to form all the ziggurats of degree one; this need not involve all the units, in which case any units not part of degree-one ziggurats are now defined themselves to be degree-one ziggurats. Degree-two ziggurats are formed in a similar manner, from the degree-one ziggurats and so on. The order of units is determined as follows. The highest degree (say $p+1$) ziggurat is expanded and the initial two degree- p ziggurats are placed at the beginning of the order (in arbitrary position relative to each other); thereafter, each degree- p ziggurat which joins this group is placed at the end of the existing order. After this, the degree- p ziggurats are expanded, and so on.

One particular advantage of the ziggurat order is that it enables any 'chaining' to be seen very easily. Chaining, which most usually occurs with single linkage cluster analysis, is the repeated joining of individual units or small groups to one larger group. When this occurs, the ziggurat order will place all the individual units and small groups after the larger group. The figure below shows a dendrogram in ziggurat order, from a single linkage cluster analysis: here the chaining of 11 single units and one group of 2 units to the single group of 15 units is displayed very clearly; there is also some indication of chaining in the formation of the group of 8 units at the left of the dendrogram.

It is important to note that the input to the macro is not the same as the merging clusters (printed) output from the HIERARCHY directive. It is fairly easy to change that output to the form required by the macro `z1gord(1)`: the program in the previous article contains the statements needed to do this. The relevant section of that program is the FOR-REPEAT loop after the comment 'loop through nodes. alter merge no.s. and accumulate weights'; there is no need to supply weights to the macro `z1gord(1)`, so the section of the loop which starts with GOTO and ends at the LABEL is superfluous.

The macro `z1gord(1)` is probably best used at run-time. It has 29 named local structures, and generates about the same number of unnamed structures, although these will be cleared by the DEVALUE statement at the end of the macro.

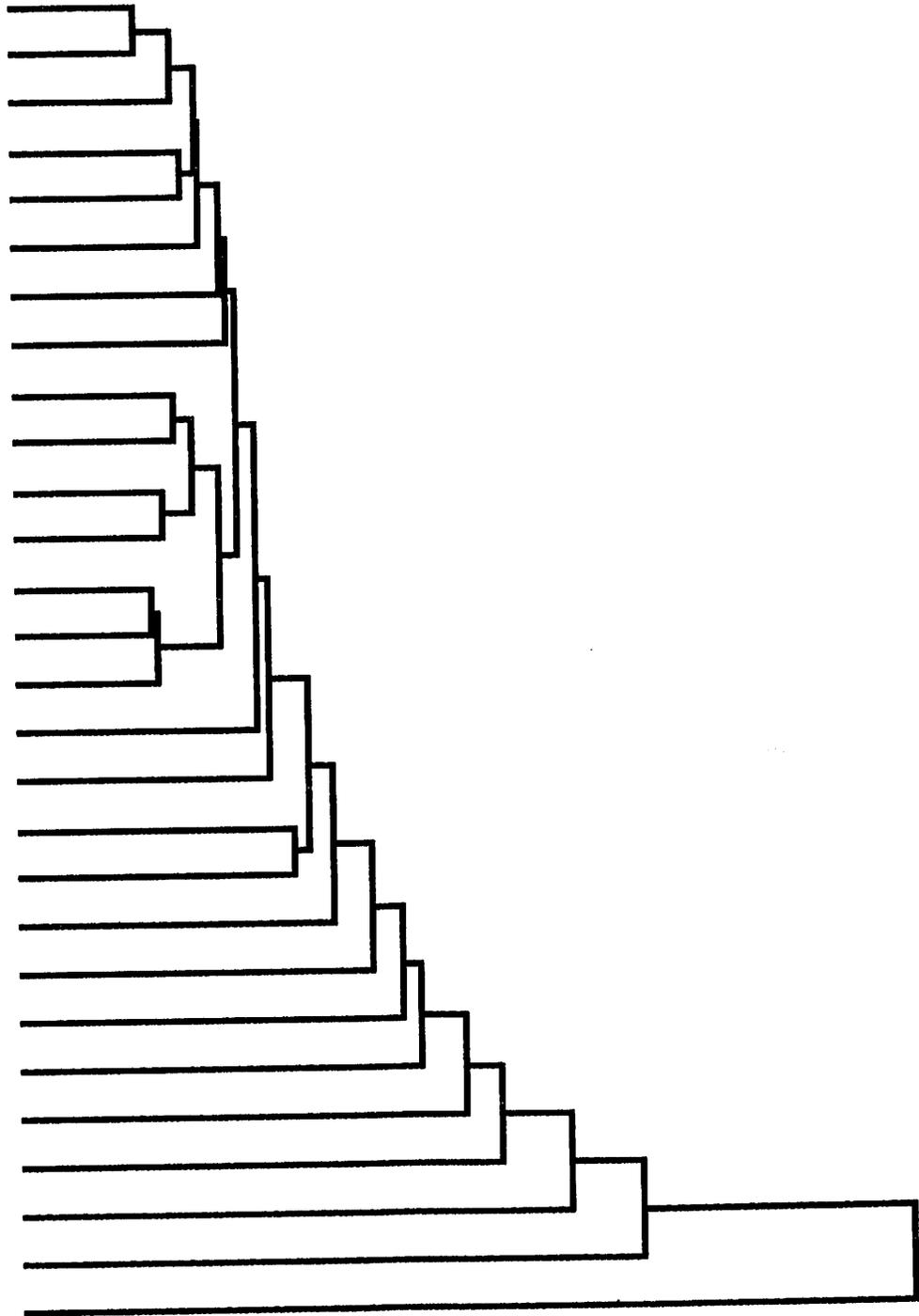


Figure 1

The Macro

```
'macro'  zigord(1) $
..
macro to calculate ziggurat clustering order according to
scheme to Frank Critchley (1983) ziggurats and dendrograms.
Report no. 43, Department of Statistics, University of Warwick
..
..
input data structures:
  merge(1,2) variates of length (n-1) specifying left and right
              nodes for the hierarchical merges of n units.
              N.B. these must be such that each merge defines a
              new node no. (n+1,...2n-1) and subsequent merges
              refer to these new node nos.; this is NOT as the
              hierarchy directive prints them; for further details
              see Digby (1985) Drawing pretty dendrograms.
              Genstat newsletter no. 14.
output data structure:
  clord      a variate of length n defining the ziggurat order.
..
..
Algorithmic details. The algorithm is complex in Genstat, although
fairly simple to do by hand. The procedure is in 2 stages: find the
ziggurat-degree of each node (N.B. here we use degrees one larger than
in Critchley's article); use this information to roll out the order.
The variate d holds the degrees of all nodes: nodes that are
intermediate in the definition of a ziggurat of degree k+1 are coded
as degree k+1/2. The formation of degrees of each node is done in
increasing degree; the rolling out is done in decreasing degree.
..
'local'  n,nm1,n2,deg,mv,mzd,pmzd,pmzdp1,nmzd,sn(1,2),zdsn(1,2),tsn(1,2),i,
        tzdsn(1,2),d,allnode,f(0...2),node,pos,zd,repeat,norepeat,temp(1,2)
'start'
  .. get sizes ..
'scalar' n,nm1,n2,deg,mv,mzd,pmzd,pmzdp1,nmzd,
        sn(1,2),zdsn(1,2),tsn(1,2),tzdsn(1,2) = *
'calc'  n2 = (n = (nm1 = nval(merge(1))) + 1) + n
'r'
'vari'  d = (1)n,(mv)n
        : allnode = 1...n2
        : f(0...2),node,pos = 1...nm1
        : clord,zd,temp(1,2) = (0)n
'equate' deg = 1
'calc'  node = node+n
'label' repeat
  .. get f s.t. f=1 for all deg-deg and deg-deg 1/2 joins ..
```

```

'calc'   f = (intpt(elem(d;merge(1))).eq.deg) .and.
           (intpt(elem(d;merge(1))+elem(d;merge(2))).eq.2*deg)
  '' check for all ziggurats of degree k+1 found completely ''
'goto'   norepeat * (sum(f.eq.f(1)).eq.nm1)
'calc'   f(1) = f
  '' put deg+1/2 into d for nodes found above ''
'calc'   f = f*node
  :     f = f+n2*(f.eq.0)
  :     elem(d;f) = deg+1/2
  '' repeat search for deg-deg+1/2 joins ''
'goto'   repeat
'label'  norepeat
  '' got all deg-deg and deg-deg+1/2 joins;
  now find all unmatched deg and deg+1/2 nodes,
  and deg+1/2-deg+1/2 matches ''
'calc'   f(1,2) = ((intpt(elem(d;merge(1,2))).eq.deg) .and.
                 (elem9d;merge(2,1)).eq.mv))
                 .or. (elem(d;merge(1,2))+elem(d;merge(2,1)).eq.2*deg+1)
  :     f(1,2) = f(1,2)*merge(1,2)+n2*(f(1,2).eq.0)
  '' put de+1 into nodes found above, increase deg, and repeat ''
  :     deg = deg+1
  :     elem(d;f(1)) = deg
  :     elem(d;f(2)) = deg
'goto'   repeat * (nmv(d).ne.0)
  '' end of stage one ''
  '' reset z-deg of all end-nodes to zero (to avoid expansion);
  initialise for stage 2 ''
'calc'   d = d*(allnode.gt.n)
  :     elem(clord;1) = 2*n-1
  :     elem(zd;1) = deg
  '' loop for stage 2; find position and node with max z-deg in order ''
'for'    i = (1)nm1
'calc'   mzd = max(zd)
'posi'   pmzd = mzd $ d
'calc'   nmzd = elem(clord,pmzd)-n
  '' get sub-nodes and their z-degs ''
  :     sn(1,2) = elem(merge(1,2);nmzd)
  :     zdsn(1,2) = elem(d;sn(1,2))
  '' make a hole in the order to expand node ''
'equate' pos = 1
'calc'   elem(pos;pmzd) = 2
  :     pos = cum(pos)
'copy'   temp(1,2) $ pos = clord,zd
  '' order sub-nodes and their z-degs; put them into order ''

```

```
'calc'   tsn(1,2),tzdsn(1,2) = sn(1,2)*(zdsn(1).ge.zdsn(2)) +
:                                               sn(2,1),zdsn(2,1)*(zdsn(1).lt.zdsn(2))
:   sn(1,2),zdsn(1,2) = tsn(1,2),tzdsn(1,2)
:   pmzdp1 = pmzd+1
'copy'   clord $ pmzd,pmzdp1 = sn(1,2)
:   zd $ pmzd,pmzdp1 = zdsn(1,2)
'repeat'
  '' end of stage 2; tidy-up and exit ''
'deval'  d,allnode,f(0...2),node,pos,zd
'endmac'
'input'  1
'r'
```

Reference

Critchley, F. (1983) Ziggurats and dendograms. Report No. 43, Department of Statistics, University of Warwick.

Drawing pretty Dendrograms

*P.G.N. Digby
Statistics Department
Rothamsted Experimental Station
Harpenden
Hertfordshire AL5 2JQ
United Kingdom*

To present a conference paper recently, I wished to produce 35 mm transparencies to display the results of hierarchical cluster analyses. Various Genstat programs were written to construct files containing the graphical control instructions for the required types of display. These were then plotted, in white on a blue background, by the Sigma Colour graphics terminal at RES and photographed onto colour transparency film.

Various types of Genstat output were used. For example, the abbreviated similarity matrix output from the directive SMPRINT was used to construct shade diagrams with blocks of differing intensities of blue. Other programs included the manipulation of minimum spanning trees, and ordinations. The most complicated process was for drawing dendrograms involving five stages, described below.

First a Genstat program was used to obtain details of the clustering, produced by the HIERARCHY directive, in a secondary output file. In the second stage this output file was edited to form the secondary input file for the third stage; this involved the deletion of most of the HIERARCHY output, to leave only the 'merging clusters' information, and the insertion of extra data, for example the number of units and their required order in the dendrogram. The third stage converted the details of the clustering into graphical control codes for the Sigma; it is this step which is described in detail below. The fourth stage again used the editor, now to convert the graphical control code output to a form which the Sigma would recognise; this mainly involved the deletion of spaces. Lastly, the file of control codes was sent to the terminal and the dendrogram was drawn.

I assume that the dendrogram is to be drawn with the units on the left and with similarity decreasing as one moves across the display from left to right. The program shown draws the dendrogram in this fashion (although it will also draw the dendrogram as a mirror image). A modified version of the program is needed to draw a 'hanging' dendrogram, i.e. with the units across the top of the display (or one with the units across the bottom).

The basis of the program is to draw separately the portion of the dendrogram which corresponds to each 'merge'. Each of these 'shapes' consists of two horizontal lines with a vertical line joining their right hand ends. These shapes are drawn in the order of merging which Genstat produces from the HIERARCHY directive. There is no attempt to minimize the total distance travelled by the pen; this does not matter on a visual display terminal but drawing large dendrograms on a plotter may be somewhat inefficient.

The position of each shape can be given by (x,y) co-ordinates of four points. For example, the first merge is of the units labelled 11 and 13 on the dendrogram shown in the figure: the four points are the two left-hand ends and the two right-hand ends of the short horizontal lines to the right of the labels 11 and 13; if we follow the shape around in a clockwise direction, let us label these points $A, B, C,$ and D . Clearly the x -coordinates of A and D are the same and their y -coordinates are easily determined from the required order of the units. These y -coordinates are also those of B and C , while the x -coordinates of B and C are a linear function of the similarity at which units 11 and 13 merge and so can easily be calculated. Now consider the next merge involving these units, which is the joining of unit 18. Let us label the four points of the shape for this merge as E, F, G and H , again traversing the shape in a clockwise direction. The x - and y -coordinates of H , and thus the y -coordinate of G , are determined in the same way as those for A and D . The x -coordinate for both F and G is again a function of the similarity at which unit 18 joins units 11 and 13. The x -coordinate of E is the same as that of B and C and the y -coordinate of E , and thus also of F , lies somewhere between those of B and C . Therefore, when the positions of B and C are calculated, we can store their x -coordinate, to use for E , and calculate and store an appropriate y -coordinate, to use for E and F . Obviously this will only work if we deal with the merges in the order in which they occur; fortunately this is how the HIERARCHY directive prints them.

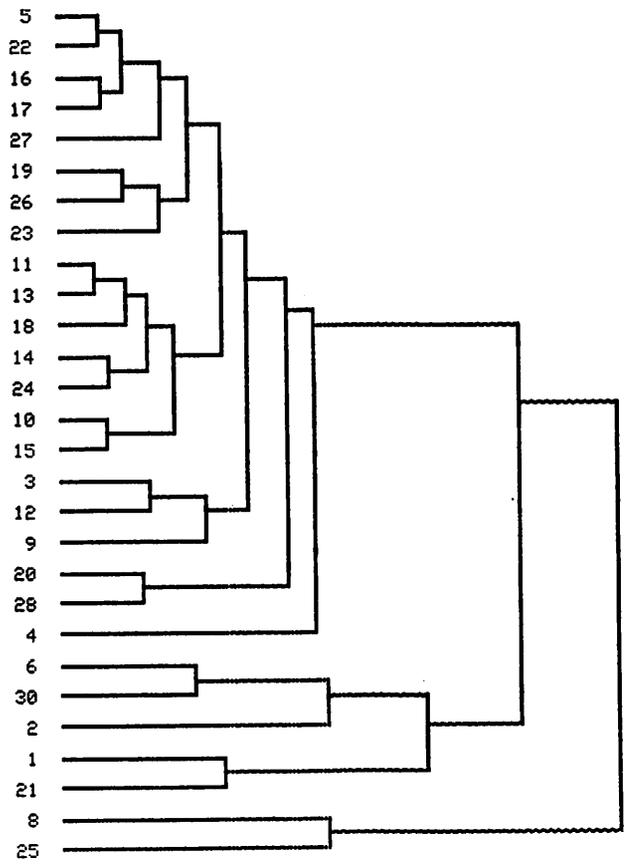


Figure 1

The above explanation can be formalised by considering the dendrogram to be a set of $(2n-1)$ nodes (assuming that there are n units to cluster). The first n nodes are simply the left-hand end positions for the units, (e.g. the points A , D and H); nodes $(n+1)$ to $(2n-1)$ represent the result of each merge, (e.g. point E) in order of merging. The position of each of these $(n-1)$ nodes can be determined from the position of two lower-numbered nodes, each new node being placed somewhere on the vertical line which is part of the 'shape' drawn for the respective merge. The program shown allows three options: the new node is at the bottom of the vertical line (this is how the printed dendrograms from Genstat appear); the new node may be half-way along the vertical line (this may be considered to be an unweighted average of the two contributing, old, nodes); or the new node may be placed at the central position of the group it represents (this is a weighted average of the two old nodes, the weights being the sizes of the two sub-groups).

We can now look at the Genstat program, presented here as Genstat output so that the lines of program and data, from the secondary input channel, are numbered according to the usual Genstat scheme. The first block of instructions reads the number of units and calculates a couple of other useful scalars. The second block of the program declares various data structures and reads in the main body of data. This consists largely of the 'merging clusters' output from the HIERARCHY directive (data lines 2 to 28) and a clustering order, i.e. the order of units for the dendrogram, which may have been saved from the HIERARCHY directive or may have been chosen separately. The other items are relevant to the graphical device and will be described later, apart from the value l (data line 42) stored in the scalar `style`, which specifies the position of the new node defined by each merge.

```

1  refe dendrograms on the sigma
2  scal n.nm1.n2m1
3  inpu 2
4  read/d n
5  inpu 1
6  calc n2m1 = n+(nm1 = n-1)
7  r

1  28

8  head headinst.tailinst
9  vari ends.endpos(1.2).clord $ n
10      merge(1.2).sim.node.sel.wt(1.2).firstold(1.2).foboth = (1)nm1
11      pos(1.2) $ n2m1
12  scal l.old.new.x0.y0.dely.delsim.thick.labfmt.wnew.style = 0
13  valu ends = 1 n
14  calc node = cum(sel)+n
15  inpu 2
16  read/d merge(1.2).sim
17  read/s,d clord.headinst.tailinst.x0.y0.dely.delsim.thick.labfmt style
18  inpu 1
19  r

2      10      12      98.4
3      5       21      98.2
4      15      16      98.1
5      9       14      97.9

```

6	13	23	97.8
7	5	15	97.2
8	18	25	97.2
9	10	17	97.1
10	19	27	96.4
11	10	13	96.2
12	3	11	96.1
13	18	22	95.6
14	5	26	95.5
15	9	10	95.0
16	5	18	94.3
17	6	28	94.2
18	3	8	93.7
19	5	9	92.9
20	1	20	92.9
21	3	5	91.8
22	3	19	90.1
23	3	4	88.9
24	2	6	88.4
25	7	24	88.4
26	1	2	84.2
27	1	3	80.1
28	1	7	75.8
29	'eod'		

```

30 5 21 15 16 26 18 25 22 10 12 17 13 23 9 14 3 11 8 19 27 4
31 6 28 2 1 20 7 24 'eod'
32 '+-*/HHGAAB
33 +-*/JG000FO
34 +-*/JG1FFF0'
35
36 476
37 540
38 -16
39 12
40 2
41 2
42 1

```

In the procedure described above, each merge defines a new node; however, the HIERARCHY directive uses one of the old node numbers to represent new nodes. For example, the first merge, of units 10 and 12, needs to be subsequently referred to as node 29 (=n+1), but HIERARCHY refers to this as node 10 (see data line 9, also data lines 11 and 15). (N.B. units 7 to 27 on the input are labelled 8 to 28 on the figure and unit 28 is labelled 30.) So the third block of instructions (program lines 20 to 34) is used to convert the HIERARCHY output to the form which we need. Line 25 locates all the relevant references to the new node and line 26 converts these to the new node numbers. Lines 28 to 30 of this block also accumulate the group sizes in the variates wt(1,2), if the centroid style of drawing has been requested by giving the scalar style the value 1 (line 27). The result of these transformations can be seen by comparing the printed table, after line 34, with the input data.

```

20  '' loop through nodes, alter merge no.s, accumulate weights ''
21  'for' dum = (i)nm1
22  'calc' i = i+1
23  'copy' sel $ i = 0
24      :   old,new = merge(1),node $ i
25  'calc' firstold(1,2) = sel*(merge(1,2).eq.old)
26      :   merge(1,2) = merge(1,2)*(1-firstold(1,2))+new*firstold(1,2)
27  'goto' overwt * (style.ne.1)
28  'calc' wnew = elem(wt(1);1)+elem(wt(2);1)
29      :   firstold(1,2) = firstold(1,2) *(cum(vsum(firstold(1,2)))) .eq. 1)
30      :   wt(1,2) = wt(1,2)*(1-firstold(1,2))+wnew*firstold(1,2)
31  'labe' overwt
32  'repe'
33  'prin/p' node,merge(1,2),sim,wt(1,2) $ (10)3,10.1
34  'r'

```

node	merge(1)	merge(2)	sim	wt(1)	wt(2)
29	10	12	98.4	1	1
30	5	21	98.2	1	1
31	15	16	98.1	1	1
32	9	14	97.9	1	1
33	13	23	97.8	1	1
34	30	31	97.2	2	2
35	18	25	97.2	1	1
36	29	17	97.1	2	1
37	19	27	96.4	1	1
38	36	33	96.2	3	2
39	3	11	96.1	1	1
40	35	22	95.6	2	1
41	34	26	95.5	4	1
42	32	38	95.0	2	5
43	41	40	94.3	5	3
44	6	28	94.2	1	1
45	39	8	93.7	2	1
46	43	42	92.9	8	7
47	1	20	92.9	1	1
48	45	46	91.8	3	15
49	48	37	90.1	18	2
50	49	4	88.9	20	1
51	2	44	88.4	1	2
52	7	24	88.4	1	1
53	47	51	84.2	2	3
54	53	50	80.1	5	21
55	54	52	75.8	26	2

The Sigma screen is addressed in absolute pixel (x,y) coordinates, and the bottom left-hand corner of the screen has coordinates (100,100), so that all locations have a pair of three-digit coordinates (as with Ordnance Survey grid references). The scalars x0 and y0 define the position of the first unit in the dendrogram and the scalar del y specifies the change in the y-coordinate from one unit to the next. Lines 35 to 39 of the program obtain these end positions for the units and put them (reordered

into the cluster order) into the first n values of the variates `pos(1,2)`, which will finally hold the positions of all $(2n-1)$ nodes. The positions of the last $(n-1)$ nodes are calculated in the next block (lines 40 to 54): if the scalar `style` has value -1 , each new node is placed at the bottom of the vertical part of its 'shape', otherwise the variates `wt(1,2)` are used to calculate the y -coordinate of the new node. (N.B. if the scalar `style` equals 0 an unweighted position will result.) The x -coordinate of each new node is a linear function of the similarity of the merge and the scalar `delsim` controls this. Once the position of each new node has been determined, the coordinates of four points to describe the 'shape' are stored in the variates `out(1...8)`.

```

35 '' get screen locations for end positions ''
36 'calc' endpos(1,2) = x0,dely
37   :   endpos(2) = cum(endpos(2))-dely+y0
38 'copy' pos(1,2) $ clord = endpos(1,2) $ ends
39 'r'

40 '' get screen positions for new nodes ''
41 'vari' out(1...8) $ nm1
42 'scal' oldx1,oldx2,newx,oldy1,oldy2,newy,old1,old2,new,simval,wt1,wt2
43 'calc' i = 0
44 'for'  dum = (1)nm1
45 'calc' i = i+1
46 'copy' old1,old2,new,simval,wt1,wt2 = merge(1,2),node,sim,wt(1,2) $ i
47   :   oldx1,oldy1,oldx2,oldy2 = pos(1,2) $ 2(old1,old2)
48 'calc' newx = x0+intpt((100-simval)*delsim+0.5)
49   :   newy = intpt((wt1*oldy1+wt2*oldy2)/(wt1+wt2)+.5)*(style.ne.-1)
50           +vmin(oldy1,oldy2)*(style.eq.-1)
51 'copy' pos(1,2) $ new = newx,newy
52 'copy' out(1...8) $ i = oldx1,oldy1,newx,oldy1,newx,oldy2,oldx2,oldy2
53 'repe'
54 'r'

```

Each 'shape' is drawn by one move-with-the-pen-up, to the start of the shape, and three moves-with-the-pen-down around the shape. Each output line draws one shape; the movement codes are G1 (pen up) and GJ (pen down), and each line of instructions for the Sigma must be prefixed by `+-*/`. These control codes are stored in the name structures `leader`, `goup` and `godown`; they can easily be changed (lines 56 to 58) to suit other devices (although their formats for printing on line 69 may also need to be changed). The headings `headinst` and `tailinst`, which were read earlier in the program, are to allow for initialisation of the Sigma; here the former heading clears the screen and defines two pen colours to be blue (for the background) and white (for the dendrogram); there are no extra instructions to perform after drawing the dendrogram, so the latter heading is blank. Ordinarily I would direct all the graphical instructions to output channel 3 (line 64) but this has been commented out here so that we can see the resulting output. Note the PRINT option `SQUASH=Y` to suppress blank lines. Dendrograms drawn one pixel wide on the Sigma appear rather thin; the scalar `thick` allows me to specify the thickness in pixels. This is achieved by repeatedly writing out the control codes for all the shapes, shrinking the shapes by one pixel each time (lines 72 to 74).

```

55 '' sigma graphics control codes ''
56 'name' leadern = +-*/
57   :   goupn = G1
58   :   godownn = GJ
59   :   leader,goup,godown $ nm1

```

```

60 'equa' leader = leadern
61      :   goup = goupn
62      :   godown = godownn
-63  ''
-64 'outp' 3
65  ''
66 'prin/squa=y' headinst
67 'labe' redo
68 'prin/p.labc=1,squa=y' leader,goup,out(1,2),godown,out(3,4),godown,
69      out(5,6),godown,out(7,8) $ 4.(2,(4)2)4
70 'calc' thick = thick-1
71 'goto' done * (thick.le.0)
72 'calc' out(2,4) = out(2,4)-2*(out(2).gt.out(8))+1
73      :   out(3,5) = out(3,5)-1
74      :   out(6,8) = out(6,8)-2*(out(8).gt.out(2))+1
75 'goto' redo
76 'labe' done
77 'prin/squa=y' tailinst
78 'outp' 1
79 'r'

```

```

+--*/HHGAAB
+--*/JG000FO
+--*/JG1FFFFO
+--*/GI 476 412GJ 495 412GJ 495 396GJ 476 396
+--*/GI 476 540GJ 498 540GJ 498 524GJ 476 524
+--*/GI 476 508GJ 499 508GJ 499 492GJ 476 492
+--*/GI 476 332GJ 501 332GJ 501 316GJ 476 316
+--*/GI 476 364GJ 502 364GJ 502 348GJ 476 348
+--*/GI 498 532GJ 510 532GJ 510 500GJ 499 500
+--*/GI 476 460GJ 510 460GJ 510 444GJ 476 444
+--*/GI 495 404GJ 511 404GJ 511 380GJ 476 380
+--*/GI 476 252GJ 519 252GJ 519 236GJ 476 236
+--*/GI 511 396GJ 522 396GJ 522 356GJ 502 356
+--*/GI 476 300GJ 523 300GJ 523 284GJ 476 284
+--*/GI 510 452GJ 529 452GJ 529 428GJ 476 428
+--*/GI 510 516GJ 530 516GJ 530 476GJ 476 476
+--*/GI 501 324GJ 536 324GJ 536 380GJ 522 380
+--*/GI 530 508GJ 544 508GJ 544 444GJ 529 444
+--*/GI 476 204GJ 546 204GJ 546 188GJ 476 188
+--*/GI 523 292GJ 552 292GJ 552 268GJ 476 268
+--*/GI 544 484GJ 561 484GJ 561 364GJ 536 364
+--*/GI 476 156GJ 561 156GJ 561 140GJ 476 140
+--*/GI 552 284GJ 574 284GJ 574 428GJ 561 428
+--*/GI 574 404GJ 595 404GJ 595 244GJ 519 244
+--*/GI 595 388GJ 609 388GJ 609 220GJ 476 220
+--*/GI 476 172GJ 615 172GJ 615 196GJ 546 196
+--*/GI 476 124GJ 615 124GJ 615 108GJ 476 108
+--*/GI 561 148GJ 666 148GJ 666 188GJ 615 188

```

```

+*/GI 666 172GJ 715 172GJ 715 380GJ 609 380
+*/GI 715 340GJ 766 340GJ 766 116GJ 615 116
+*/GI 476 411GJ 494 411GJ 494 397GJ 476 397
+*/GI 476 539GJ 497 539GJ 497 525GJ 476 525
+*/GI 476 507GJ 498 507GJ 498 493GJ 476 493
+*/GI 476 331GJ 500 331GJ 500 317GJ 476 317
+*/GI 476 363GJ 501 363GJ 501 349GJ 476 349
+*/GI 498 531GJ 509 531GJ 509 501GJ 499 501
+*/GI 476 459GJ 509 459GJ 509 445GJ 476 445
+*/GI 495 403GJ 510 403GJ 510 381GJ 476 381
+*/GI 476 251GJ 518 251GJ 518 237GJ 476 237
+*/GI 511 395GJ 521 395GJ 521 357GJ 502 357
+*/GI 476 299GJ 522 299GJ 522 285GJ 476 285
+*/GI 510 451GJ 528 451GJ 528 429GJ 476 429
+*/GI 510 515GJ 529 515GJ 529 477GJ 476 477
+*/GI 501 325GJ 535 325GJ 535 379GJ 522 379
+*/GI 530 507GJ 543 507GJ 543 445GJ 529 445
+*/GI 476 203GJ 545 203GJ 545 189GJ 476 189
+*/GI 523 291GJ 551 291GJ 551 269GJ 476 269
+*/GI 544 483GJ 560 483GJ 560 365GJ 536 365
+*/GI 476 155GJ 560 155GJ 560 141GJ 476 141
+*/GI 552 285GJ 573 285GJ 573 427GJ 561 427
+*/GI 574 403GJ 594 403GJ 594 245GJ 519 245
+*/GI 595 387GJ 608 387GJ 608 221GJ 476 221
+*/GI 476 173GJ 614 173GJ 614 195GJ 546 195
+*/GI 476 123GJ 614 123GJ 614 109GJ 476 109
+*/GI 561 149GJ 665 149GJ 665 187GJ 615 187
+*/GI 666 173GJ 714 173GJ 714 379GJ 609 379
+*/GI 715 339GJ 765 339GJ 765 117GJ 615 117

```

The final block of instructions reads a set of labels for the units, provided that is required (labfmt not equal to 0). The positions for these are calculated on lines 89 and 90 (this may need to be changed for different graphical devices) and then instructions for drawing the labels are also output (usually to channel 3).

```

80 'goto' exit * (labfmt.le.0)
81 'name' labsn = +*/GI
82   :   textn = BA
83 'name' labsname, textname, labels $ n
84 'equa' labsname = labsn
85   :   textname = textn
86 'inpu' 2
87 'read/,d' labels
88 'inpu' 1
89 'calc' endpos(1) = endpos(1)+10-(20+7*labfmt)*(delsim.ge.0)
90   :   endpos(2) = endpos(2)-3
-91 ..
-92 'outp' 3
93 ..
94 'prin/p,labc=1,squa=y' labsname,endpos(1,2),textname,labels $ 6,(4)2.2,labfmt

```

```
95 'labe' exit
96 'outp' 1
97 'r'
```

```
43 .5 22 16 17 27 19 26 23 11 13 18 14 24 10 15 .3 12 .9 20 28 .4
44 .6 30 .2 .1 21 .8 25 'eod'
+*/GI 452 537BA.5
+*/GI 452 521BA22
+*/GI 452 505BA16
+*/GI 452 489BA17
+*/GI 452 473BA27
+*/GI 452 457BA19
+*/GI 452 441BA26
+*/GI 452 425BA23
+*/GI 452 409BA11
+*/GI 452 393BA13
+*/GI 452 377BA18
+*/GI 452 361BA14
+*/GI 452 345BA24
+*/GI 452 329BA10
+*/GI 452 313BA15
+*/GI 452 297BA.3
+*/GI 452 281BA12
+*/GI 452 265BA.9
+*/GI 452 249BA20
+*/GI 452 233BA28
+*/GI 452 217BA.4
+*/GI 452 201BA.6
+*/GI 452 185BA30
+*/GI 452 169BA.2
+*/GI 452 153BA.1
+*/GI 452 137BA21
+*/GI 452 121BA.8
+*/GI 452 105BA25
```

```
98 'clos'
```

The fourth step, after running this program, is to do two global edits on the file of graphical instructions. The first removes all the spaces, the second replaces all decimal points with spaces (hence the decimal points on the single digit labels when they were input in data lines 43 and 44). Lastly, the file of instructions is sent to the graphics terminal and the dendrogram appears on the screen. This is shown above (in black-on-white).

Obviously, the program given here will only be useful, as it stands, to drive Sigma graphical terminals like the one we have at Rothamsted. However, it could be modified very easily to provide graphical control codes for many other types of device. Most important is that the basic algorithm to generate the instructions for drawing dendrograms is device independent.

References

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.

Using Genstat to Fit Complex Association Models to Contingency Tables

B. Fingleton
 School of Geography
 Cambridgeshire College of Arts and Technology
 East Road
 Cambridge CB1 1PT
 United Kingdom

Introduction

A useful development in modelling contingency table frequencies (entailing a Poisson response variate) has been the advent of the family of association models, as described, for example, by Goodman (1979) and Agresti (1983). Fingleton (1984) gives some simple Genstat code for fitting log-linear models from this family and for fitting log-linear contingency table models in general. The aim of the present paper is to supplement this with further Genstat code (given later) which pertains to a more complex association model which is not log-linear but which has some useful features (model (5) below). The facilities in Genstat enable model (5) to be fitted more easily than methods available hitherto. Although he does not give specific code showing how to fit model (5), Breen (1984) uses GLIM to perform a model fitting exercise similar to the one described here; he gives details of how GLIM fares vis-a-vis the alternative approach described in the literature and the comparisons he makes and advantages he cites apply equally to Genstat.

Some Examples

Prior to setting out some details of association models allied to model (5), it is illuminating to illustrate features of the model using the data in Table 1, which is taken from Goodman (1979). A reason for fitting any model to data is to elicit otherwise latent information from those data, and this is no less the case in this context. In particular, one pertinent question on which model (5) can shed light concerns the 'true' distances between the categories forming the row and column variables of Table 1, given that the 'usual' method of analysis, via model (1), is to assign scores, say 0,1,2,..., to the categories, and to presume that they are tenable.

Mental Health Status	Parents Socio-economic Status					
	A	B	C	D	E	F
Well	64	57	57	72	36	21
Mild symptom formation	94	94	105	141	97	71
Moderate symptom formation	58	54	65	77	54	54
Impaired	46	40	60	94	78	71

Cross-classification of subjects according to their mental health and the parents' socio-economic status (from Goodman (1979)).

Table 1

The first column of Table 2a contains the assigned column scores and the second column contains scores estimated under model (5) via Genstat. For convenience, since the estimates *per se* possess no intrinsic value, scaled estimates are provided in the third column. These are obtained thus:

$$(\text{estimate} + 1.0152) \times (5 / (3.8789 + 1.0152)) = \text{scaled estimate of column score.}$$

Category	Assigned Score	Genstat parameter estimate	Scaled parameter estimate
(a) Columns			
A	0	-1	0.0155
B	1	-1.0152	0
C	2	0.2341	1.2763
D	3	0.8968	1.9534
E	4	2.5339	3.6259
F	5	3.8789	5
(b) Rows			
Well	0	-1	0
Mild symptom formation	1	-0.8463	1.4918
Moderate symptom formation	2	-0.8185	1.7616
Impaired	3	-0.6909	3

Row and Column Scores and Estimates for Table 1

Table 2

Note that the relative intercategory distances, which *are* of intrinsic value, are not affected by this and that comparison with the assigned score is made easier. It is evident from Table 2a that socio-economic status A is quite similar to socio-economic status B, though strictly they should change places, as they are not in order in the original table. Scaling the row estimates in an analogous way produces the scaled estimates of Table 2b. From these it is evident that 'mild symptom formation' and 'moderate symptom formation' are less distinct categories than presumed in using the assigned scores.

Naturally, these deductions rest on certain assumptions. In particular, if linear by linear association holds true, then the estimates in Table 2 provide the relative intercategory distances. Alternatively, they provide the intercategory distance ratios required to best approximate linear by linear association. (The term linear by linear association is used when the deviation from independence between row and column variables *R* and *C* is linear in *R* for fixed *C* and linear in *C* for fixed *R*).

Various sociological applications are described by Clogg (1982b) who illustrates how intercategory distance estimates can be useful. For example, Clogg uses association models to ascertain the position of a 'don't know' response to a question concerning attitudes about court judgements. He finds that one way of wording the question puts 'don't know' between 'about right' and 'not harshly enough', and very close to the latter. He explains this by saying that "perhaps the 'don't know's' for this wording were leaning toward the response, 'not harshly enough', but for whatever reason did not wish to commit themselves to this 'conservative' response". The second form of wording the question places 'don't know' between 'too harsh' and 'about right', close to the latter.

There are certainly other areas in the biological and behavioural sciences where the order of, and relative distance between, categories is of substantive interest. Upton and Fingleton (1985), for example, analyse spatial data using this approach.

Some Relevant Fundamental Aspects of Association Models

We now make a minor digression into theory in order to explicate the model (5) which is of ultimate interest, commencing the discussion with the simplest member of the association family, the so-called uniform association model, which is, for the two-way table,

$$\ln m_{ij} = \mu + \lambda_i^R + \lambda_j^C + v_i v_j \varphi \quad (1)$$

where m_{ij} is the expected frequency in table cell (i,j) , R and C are cross-classifying row and column factors with I and J levels respectively, $v_i = i - 1$ and $v_j = j - 1$ ($i = 1, \dots, I$; $j = 1, \dots, J$). Under this model, the association between variables R and C is represented by the single parameter

$$\varphi = \ln \frac{m_{ij} m_{i+1,j+1}}{m_{i+1,j} m_{i,j+1}}$$

Model (1) is fitted by the Genstat directive

```
'fit' ROWS + COLS + ZZ
```

where ROWS and COLS are factors representing λ_i^R and λ_j^C and ZZ is a variate obtained, for example, by

```
'calc' ZZ = (ROWS-1) * (COLS-1)
```

A generalized expression of the structure imposed by the model on the expected frequencies, to allow for arbitrary scores v_i and v_j , is

$$\varphi(v_{i+1} - v_i)(v_{j+1} - v_j) = \ln \frac{m_{ij} m_{i+1,j+1}}{m_{i+1,j} m_{i,j+1}}$$

The goodness-of-fit statistic for model (1) is checked against $\chi_{f,\alpha}^2$ where $f = (I-1)(J-1) - 1$.

Part of the fitting procedure for model (5) requires the use of the row-effects model (2) and the column-effects model (3), close relatives of the uniform association model which modify the interaction according to the row or column of the table.

$$\ln m_{ij} = \mu + \lambda_i^R + \lambda_j^C + \varphi_i^R v_j \quad (2)$$

$$\ln m_{ij} = \mu + \lambda_i^R + \lambda_j^C + \varphi_j^C v_i \quad (3)$$

Under model (2),

$$(\varphi_{i+1}^R - \varphi_i^R)(v_{j+1} - v_j) = \ln \frac{m_{ij} m_{i+1,j+1}}{m_{i+1,j} m_{i,j+1}}$$

and under model (3),

$$(\varphi_{j+1}^C - \varphi_j^C)(v_{i+1} - v_i) = \ln \frac{m_{ij} m_{i+1,j+1}}{m_{i+1,j} m_{i,j+1}}$$

In model (2), variable C is ordinal with scores v_j and in model (3) variable R is ordinal with scores v_i . In Genstat, model (2), say, can be quickly fitted using

```
'fit' COLS + ROWS * colscores
```

Since in model (2) φ_i^R replaces the single uniform association parameter φ , there are now $(I-1)(J-1) - I + 1 = (I-1)(J-2)$ degrees of freedom for testing the model. Likewise there are $(I-2)(J-1)$ degrees of freedom for testing model (3).

It is opportune to mention a further refinement with two ordinal variables and both row and column effects, when it is appropriate to fit model (4)

$$\ln m_{ij} = \mu + \lambda_i^R + \lambda_j^C + \varphi_i^R \nu_j + \varphi_j^C \nu_i \quad (4)$$

In this case,

$$(\varphi_{i+1}^R - \varphi_i^R)(\nu_{j+1} - \nu_j) + (\varphi_{j+1}^C - \varphi_j^C)(\nu_{i+1} - \nu_i) = \ln \frac{m_{ij} m_{i+1, j+1}}{m_{i+1, j} m_{i, j+1}}$$

and there are $(I-2)(J-2)$ degrees of freedom for the chi-squared goodness-of-fit statistic. We can fit this model in Genstat using

`'fit' ROWS * colscores + COLS * rowscores`

We have suggested that the most interesting of the association models is model (5), since it facilitates the *estimation* of all the relative intercategory distances rather than using *assumed* category scores. The reason why we focus on intercategory distances is because intercategory distance ratios are preserved irrespective of changes in the scale and location of the category values, as was shown in the example (see Table 2). This is important in enabling comparisons to be made between alternative parameterisations. Some authors, for example Goodman, write category values in deviation form and, incidentally, invoke 'centred-constraints' on the λ and φ parameters instead of the Genstat default option of 'cornered constraints'.

Written as

$$\ln m_{ij} = \mu + \lambda_i^R + \lambda_j^C + \varphi \nu_i \nu_j \quad (5)$$

model (5) is reminiscent of the uniform association model (1), although, as was mentioned earlier, it is not loglinear and therefore maximum likelihood estimation demands the slightly more complicated iterative method coded below. Note that to test model (5) we use a chi-square distribution with the same number of degrees of freedom as model (4).

The pattern established by model (5) on the expected frequencies is given by

$$\varphi(\nu_{i+1} - \nu_i)(\nu_{j+1} - \nu_j) = \ln \frac{m_{ij} m_{i+1, j+1}}{m_{i+1, j} m_{i, j+1}}$$

which is comparable to that of model (1) (indeed the φ of the two models are directly comparable if the estimated ν_i and ν_j are placed on the same scale as the fixed scores of model (1) (see Clogg (1982b)).

A major difference between models (1) and (5) is that model (5) is invariant when row and column orders are changed (see Goodman (1979)). Indeed, model (5) can be considered to be a generalization of models (2) and (3). This is apparent from the fact that the estimated expected frequencies of model (5) are exactly equal to those of model (2) when the column scores ν_j are taken as model (5)'s ν_j estimates, in which case the φ_i^R from model (2) and the ν_i estimates from model (5) take the same values. A similar condition holds for estimates of model (3).

Using Genstat to fit Model (5)

Estimation via Genstat proceeds by first assuming values for ν_j in order to fit the row-effects model (2), which provides the estimates ν_i . These are then taken as the fixed row variable scores in the column-effects model (3), in order to estimate ν_j . The iteration then returns to model (2) using these ν_j estimates as known scores etc. One can terminate the iteration when minimal change occurs in successive ν_i and ν_j estimates. An alternative stopping rule is to terminate when the

estimate expected frequencies given by models (2) and (3) are equal. The above forms lines 12 to 22 of the code. To make models (1) and (5) comparable, we invoke line 24,

```
'fit' ROWS + COLS + ZZ
```

having first obtained ZZ via

```
'calc' ZZ = rowscores * colscores
```

where rowscores and colscores are now the final estimates of ν_i and ν_j . Note that the wrong degrees of freedom are printed when line 24 is used to fit model (5) because the fact that ZZ is calculated from estimated parameters is not taken into account.

The Genstat Program

```
'reference'    GOOD79A
'unit'        $24
'variate'     COUNT=64,57,57,72,36,21,94,94,105,141,97,71,
              58,54,65,77,54,54,46,40,60,94,78,71
'factor'     ROWS$ 4 =6(1...4)
              :      COLS$ 6 =(1...6)4
'scalar'     maxdiff,r(2...4)
'variate'     oldc $ 14 : 'calc' oldc=0
'calc'       rowscores,colscores = ROWS,COLS -1 : ZZ=(ROWS-1)*(COLS-1)
'terms'      COUNT + ROWS*colscores + COLS*rowscores + ZZ
'y/error=poisson' COUNT
'label'      startofloop
'fit'        COLS + ROWS*colscores ; coeff=C  ''Model 2''
'equation'   r(2...4)=C $ 9X,3
              :      rowscores=6(0,r(2...4))
'calc'       rowscores=rowscores-1
'fit'        ROWS + COLS*rowscores ; coeff=C  ''Model 3''
'equation'   colscores=0,C $ 1,9X,5
'calc'       colscores=colscores-1
'calc'       maxdiff=abs(max(C-oldc))
              :      oldc=C
'jump'       startofloop * (maxdiff.gt.0.00001)
'calc'       ZZ = rowscores*colscores
'fit/cau'    ROWS + COLS + ZZ  ''Model 5''
'calc'       ZZ = (ROWS-1)*(COLS-1)
'fit/cau'    ROWS + COLS + ZZ  ''Model 1''
'run'
'close'
'stop'
```

Conclusion

There are other tactics for fitting model (5), based on the same overall strategy, which have been advanced by Goodman (1979) and Clogg (1982a). However the Genstat coding given here is comparatively succinct and is capable of easy generalization, to include the extra factors and homogeneity constraints which they suggest.

References

- Agresti, A. (1983) A survey of strategies for modelling cross-classifications having ordinal variables. *Journal of the American Statistical Association*, **78**, 184–198.

- Breen, R. (1984) Fitting nonhierarchical and association loglinear models using GLIM. *Sociological Methods and Research*, **13**, 77–107.
- Clogg, C.C. (1982a) Some models for the analysis of association in multiway cross-classifications having ordered categories. *Journal of the American Statistical Association*, **77**, 803–815.
- Clogg, C.C. (1982b) Using association models in sociological research: some examples. *American Journal of Sociology*, **88**, 114–134.
- Fingleton, B. (1984) *Models of Category Counts*. Cambridge University Press.
- Goodman, L.A. (1979) Simple models for the analysis of association in cross-classifications having ordered categories. *Journal of the American Statistical Association*, **74**, 537–552.
- Upton, G.J.G. and Fingleton, B. (1985) *Spatial Data Analysis by Example. Volume 2*. Wiley.

Manual and Notice Board Amendments

*H.R. Simpson
Statistics Department
Rothamsted Experimental Station
Harpenden
Hertfordshire AL5 2JQ
United Kingdom*

Manual Amendments

An amendment list for the Genstat 4.04 Manual is enclosed with this newsletter. Further copies may be obtained, free of charge, from NAG Central Office.

With the exception of the first two corrections, all amendments should also be applied to any Genstat 4.03 manuals. In addition, the following amendments should be made to 4.03 manuals only:

Part I, Ch. 11 2 p 11 and 2 p 12

The explanations of RE 18 and RE 23 should read:

RE 18 Vector for prediction is not in current regression model

RE 23 Some parameters in the current regression model are aliased.

Part II, Contents p 4 and p 5

Insert

5.2.7.1 Options 5.2 p 5

6.7.1.5 Options 6.7 p 2

Part II, 5.2 p 5

The section number '5.2.3.1' should be '5.2.7.1'

Genstat 4.04B Notice Board

The error notices below should be inserted in the Notice Board in the appropriate (alphabetic) position.

R69	***** Error *****	04/12/84
'ANOVA' & 'REGRESS'	With LIMA=1, a model formula A.B/C will be expanded to be the main effect of C, instead of a null formula. If a formula contains expressions of this sort, expand it explicitly.	
R45	***** Error *****	21/11/84
'CALC'	The symbols < and > should not be used; they are incorrectly compiled as .NE. and .LE. !	
R65	***** Error *****	04/12/84
'CALC'	The functions PDIT and TPDT do not check the number of arguments given, which may lead to errors without comment if only one argument is given.	
R66	***** Error *****	04/12/84
'CALC'	The function LINT does not check the type of the third argument, which should be scalar. This can lead to errors without comment.	
R67	***** Error *****	04/12/84
'CALC'	Use of an ELEM function within an ELEM function can produce unexpected results.	
R68	***** Error *****	04/12/84
'CALC'	RSYMRI will not accept diags as arguments.	
R46	***** Error *****	21/11/84
Cluster	The directives ESMAT, KEY, LIST, RELATE and SMATRIX will fail if more than 200 variates are given for the data matrix.	
R60	***** Error *****	01/08/84
'ESTIMATE'	Diagnostic of 'OP 30' is reported when there are no residual degrees of freedom for estimating the parameters of a time-series model.	
R61	***** Error *****	01/08/84
'ESTIMATE'	If the EXACT and MARGINAL form of the likelihood is used, the deviance is wrong, both when printed by the PRINT option and saved with the DEV keyword. The value actually given is the Residual sum of squares, which should be scaled to form the deviance. (The scaling is not relevant for the LS form of the likelihood.) To get the deviance, set REPORT = -1 in the 'ESTIMATE' statement, and take the last value of the deviance given in the convergence report.	
R63	***** Error *****	31/10/84
'ESTIMATE'	If a model with no parameters is fitted (not even a constant), then will fail when attempting to produce correlations, unless the PRINT option is set without the letter C.	
R64	***** Error *****	31/10/84
'ESTIMATE'	If the integer of time-series model orders has an invalid length, e.g. the period of seasonality is left out, subsequent time-series statements may produce rubbish or fail.	
R57	***** Error *****	25/07/84
'FIT' etc	Variance-covariance matrices are not calculated when there are no residual degrees of freedom even when the variance is specified by the SCALE option of 'Y'	

Notices

4th Genstat Conference : September 1985

*J. Byrne
Computing Service
University of York
York Y01 5DD
United Kingdom*

The 1985 conference, for all those interested in the Genstat statistical package, will take place at York (England) from Monday 23 September to Thursday 26 September 1985. The scientific programme will include:

- * Descriptions of new Genstat facilities.
- * Interesting applications of Genstat.
- * Use of Genstat in the teaching of statistics.
- * Lectures explaining the statistical methodology in Genstat.
- * Descriptions of the use of macros.

Participants will also be able to see demonstrations of the latest Genstat release, exchange views with other users and obtain advice from the developers of Genstat.

Practical Arrangements

The conference will take place at Langwith College and in the Physics Department of the University of York. The University is at Heslington, a short bus ride from the centre of York. Participants may stay on campus at Langwith College or may arrange their own accommodation in York itself. College accommodation is mainly in single rooms but a few twin rooms are available. Evening meals will be available at Langwith College on Sunday, Monday and Tuesday but not Wednesday, when the conference dinner will take place at St. William's College (close to York Minster).

The registration fee for participants is £45 before 1 July, £50 thereafter. This includes tea, coffee and the four mid-day meals. Accompanying persons (wives, husbands, children, etc.) need pay registration only if they require lunch, for which the fee would be £15. Registration will take place at 3 pm to 6 pm on Sunday 22 September before a sherry reception at 6-7 pm. Alternatively, late registration will be available on Monday morning. (The first session will start at 9.30 am on Monday.)

On Tuesday afternoon two excursions will be available: either Fountains Abbey and the 18th Century landscaped Studley Royal Park (or Newby Hall, if wet), or the North Yorkshire Moors Railway (by coach to Pickering, by rail to Grosmont and then back across the moors to York by coach). Places are limited and early booking is strongly advised.

Costs (per person)

	£
Registration: Participants – before 1 July 1985	45.00
on or after 1 July 1985	50.00
Accompanying persons	15.00
Bed and Breakfast (at Langwith College)	12.50 per day
Evening Meal (at Langwith College)	4.50 per day
Conference Dinner (including wine and entertainment)	12.50
Excursions: exact costs are not yet available but are expected to be in the region of	5.00 – 6.00

A Genstat Primer

*A.J. Weekes
Department of Economics and Related Studies
University of York
York Y01 5DD
United Kingdom*

This is the third redraft of notes which have been used in the University of York since October 1981. The interest shown in them outside the University has encouraged me to make them more widely available.

This version runs to 120 pages and includes exercises and data sets, as well as some notes on detecting and correcting mistakes in user programs. The chapter headings are as follows:

- Structures Statements and Summary Statistics**
- Assigning values to structures and making calculations**
- More on structures, data layout and calculations**
- Improving the appearance of output (and other miscellaneous matters)**
- Loops in Genstat programs**
- Graphical displays**
- Factors**
- Some simple uses for factor structures; Table structures**
- Using Genstat to fit regression models**
- An introduction to matrix structures in Genstat**
- Some notes on multivariate statistical analysis in Genstat**
- Some finer points**
- Further reading**
- Some notes on errors in Genstat user programs**

The Primer is available now, price £ 4.20 per copy, including postage. A discount is possible on orders for five or more copies.

