NAG®

GENSTAT

Newsletter

Issue No. 30

Editors

Sue Welham
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
HARPENDEN
Hertfordshire
United Kingdom AL5 2JQ

Anna Kane
NAG Ltd
Wilkinson House
Jordan Hill Road
OXFORD
United Kingdom OX2 8DR

©1994 The Numerical Algorithms Group Limited
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.

Printed and Produced by NAG®
NAG is a registered trademark of:
The Numerical Algorithms Group Ltd
The Numerical Algorithms Group Inc
The Numerical Algorithms Group (Deutschland) GmbH
Genstat is a trademark of the Lawes Agricultural Trust
GLIM is a trademark of the Royal Statistical Society

ISSN 0269-0764

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

Please note that the cover of this Newsletter has been adapted by kind permission of Oxford University Press, from the cover of the Genstat 5 Reference Manual.

Genstat Newsletter

Issue No. 30

NP2584

May 1994

Contents

Published by
The Rothamsted Experimental Station Statistics Department
and The Numerical Algorithms Group Ltd

Editorial

This issue of the newsletter sees a change in editors: Sue Welham from Rothamsted and Anna Kane from NAG are taking over from Peter Lane and Geoff Morgan. We would like to thank Peter and Geoff for all their hard work over the past few years.

There are several Genstat meetings now being planned. There will be an open residential Genstat Introductory Course based on Release 3 held in Cambridge during July 4-6th. As usual, the course will include a large proportion of practical work and is given by Genstat developers. This will be followed by a one-day Workshop on Time Series Analysis in Genstat on July 7th with Dr Granville Tunnicliffe-Wilson from Lancaster University as the main lecturer. Further details of these courses are available from NAG. Later in the year, during 28-30th November, a statistical conference for Genstat users will be held in Wagga Wagga (New South Wales, Australia) around the themes REML/GLMMs, generalized additive models and repeated measures / spatial analysis. A wide range of invited speakers will be present and contributed papers will also be included in the program: abstracts should be submitted by 31st August. A flyer for the conference is included in this issue of the newsletter, and abstracts should be sent to Ross Cunningham, Department of Statistics, ANU, GPO Box 4, Canberra City ACT 2601, Australia (Fax +61 6 249 8007; AARNet: Ross.Cunningham@anu.edu.au). Details have also been posted to the Genstat discussion list.

This issue of the newsletter contains the second helping of 'Genstat Talk', a summary of some of the subjects discussed by those users who have tuned in to the Genstat discussion list. Details of how to subscribe to the ever-growing list are given once again.

Also present in this issue are some of the papers arising from the Eighth International Genstat Conference held last July at the University of Kent at Canterbury, which was attended by about 70 participants mainly from the UK, but with a substantial contingent from the Netherlands and individuals from Denmark, Australia and Africa. The issue begins in earnest, however, with a detailed article describing the new facilities of the eagerly awaited Genstat 5 Release 3, which is now available from NAG.

Then follow several articles on Genstat's GLM capabilities. The first discusses the properties of some GLMM estimators for a simple binomial model based on a simulation study. Several of these estimators are directly available in Genstat by using the GLMM procedure. The second article illustrates how the Genstat GLM facilities can be exploited to estimate the parameters of a finite mixture model, whilst the third describes how the GLM approach can be adopted to fit growth-curve models.

Next comes an article dealing with a topic not often discussed: the use of Genstat in statistical process control. Here, a Genstat Taguchi analysis is illustrated together with a brief description of the method and underlying principles.

The final three articles deal with a variety of subjects. The first introduces a procedure to assess the performance of models with independent data, whilst the second describes the application of Genstat ANOVA to the analysis of Youden rectangles. The third article arises from the Australasian Genstat Conference in Roturua, and describes how to construct a useful interface between Genstat and the Gnu Emacs editor.

From now on, the procedures appearing in this and future Genstat newsletters are available in electronic form in the NAG bulletin board, which is run under the Gopher server. Connection details are:

Gopher: Name=NAG Gopher Server
Type=1
Port=70
Path=1/
Host=www.nag.co.uk

Mosaic: http//www.nag.co.uk:70/

GENSTAT TALK

Extracts from the Genstat electronic discussion list, May to November 1993, summarized and edited by Peter Lane, Rothamsted. To join the discussion, send the message:

SUBSCRIBE Genstat *first-name last-name*

to the address: **LISTSERV@IB.RL.AC.UK**

The opinions expressed here are not necessarily endorsed by either NAG or Rothamsted, and statements may not have been checked for accuracy. However, members of the Genstat development team and of NAG's Statistics Section are contributors to the discussion.

Mixtures of distributions

Query: Does anyone have a Genstat procedure for estimating the parameters of a mixture of two Normal distributions?

Reply: See the article *Finite mixture distributions* by Whitaker in Genstat Newsletter 28.

Genstat under DOS 6

Query: Does the MS-DOS version of Genstat run under MS-DOS version 6? The Installer's note says it needs versions later than 3.3.

Reply: Genstat 5 Release 2.2 works fine under MS-DOS 6 using QEMM386. I haven't tried using the new version of EMM386 that comes with DOS 6, but QEMM386 has always been a better and more reliable memory manager that EMM386.

Unbalanced ANOVA

Query: I'm trying to analyse a large designed experiment with nested treatments. This is usually easy with the **BLOCK** and **ANOVA** directives, but one factor is unbalanced. I have tried **FIT**, but it gives diagnostic RE 16. There are 3889 parameters, so the SSPM needs 7564105 values but there is room for only 9172552. Is this problem just too large?

Reply 1: I suggest using **REML**, but I don't know how the space requirements compare to **FIT**.

Reply 2: I would be amazed if anyone could understand or interpret an analysis with 3889 parameters! My tack with such a problem would be to break it into smaller parts.

Addendum: There is an error in the diagnostic reporting of the trial version of Release 3 that generated the message: the calculations of space available is wrong.

DBOS 2.70

Query: Is there a version of DBOS (the PC memory manager supplied with Genstat) that is more reliable than Version 2.67? I have encountered several anomalies running Genstat under Windows, particularly in conjunction with the Pathworks local area network software, Microsoft's own memory managers, and SMARTDRV. I have tried adding the Shift Interrupts switch to DBOS, but it still does not yield a reliable and predictable environment.

Reply: My experience with DBOS is that the Shift Interrupts switch works only when DBOS is directly handling extended memory. If QEMM386, HIMEM or EMM386 are loaded, then DBOS calls for memory from them. DBOS uses interrupt 5, which often conflicts with other hardware: BUS mice or VGA cards.

Rejoinder: Since my initial query, I have received DBOS 2.70 from NAG, and virtually all of the DBOS reliability problems we were having have vanished. I would recommend anyone else having DBOS problems to get this version from NAG.

Diallel crosses

Query: In Newsletter 10, Robin Thompson described how to analyse diallel crosses with Genstat 4.03. Has anyone translated to Genstat 5?

Reply 1: A procedure has been submitted to the Library to analyse full and half diallels, though not by Robin Thompson's method. It does not deal yet with Griffing or with incomplete tables.

Reply 2: An article is being prepared for this Newsletter about diallel crosses. I have a translation of Robin Thompson's method into Genstat 5, and some code implementing Griffing's Method 4, model 1.

Covariates in REML

Query: I have a covariate in the fixed terms of a REML model. If I subtract a constant from the covariate, the means given for the factors are affected. Is this a sensible thing to happen?

Reply: **REML** presents means for level 0 of the covariate. If you want another level, say the mean of the covariate, or some reference value, you have to subtract it from the covariate.

Demonstrating Genstat

Query: I want to demonstrate some features of Genstat using a PC. For speed, I want Genstat to read pre-prepared commands, echo them and process them on screen, pausing at strategic points. **INPUT** does most of this but it isn't easy to get it to pause in the right places.

Reply 1: One possibility is the GNU emacs interface for Genstat. It gives you multiple windows, even on a terminal or PC without Microsoft Windows, so you can step through commands in one window and see the output in another. Emacs is freely available by FTP.

Reply 2: The usual way we have done this in the past is to intersperse the pre-prepared statements with **INPUT** statements, trying to ensure that not too much output is generated from one **INPUT** to the next. This can sometimes be achieved by suppressing default output and using the ***DISPLAY** directives. The statement

```
SET [PAUSE=n]
```

can be used to break up long sections. With the PC implementation you can always scroll through the output window which has a 1000-line memory.

Ticks and labels

Query: What is the neatest way of suppressing tick marks and labels on high-res graphs? I have a vague recollection that labels can be suppressed by specifying them as strings of space characters, but there must be a better way than that.

Reply: In Release 2 there is no neat solution. But in Release 3, additional parameters **XMPOS**, **XLPOS**, **YMPOS** and **YLPOS** in the **AXES** directive give explicit control over tick marks and labels. Setting **XMPOS=***, for example, will suppress tick marks on the x-axis. Other settings allow marks to be centred, above or below the axis.

Extracting values from a vector

Query: I have a simple-sounding problem which is giving me no end of hassles. I have a vector of parameters (**Beta**) and I wish to extract its individual elements into variables (**b1** and **b2**, say) for use in a calculation. How can I do it?

Reply: You can use **EQUATE**:

```
SCALAR b1,b2
 EQUATE OLD=Beta; NEW=!p(b1,b2)
```

or **CALCULATE**:

```
CALC b1,b2 = Beta$[1,2]
```

This latter method may mean you don't need to extract the values at all, but can use the expression on the right directly in other calculations. But in Release 2 you need to remember that **Beta$[1]** is treated as a variate with one value rather than a scalar, which can cause problems in calculations with variates of a different length.

ANOVA with unequal variances

Query: How can I perform an ANOVA on data with slightly different variances? Bartlett's test gives a value between the 90th and 95th percentile of chi-squared. The best solution so far is a weighted sum of squares approach, suggested by Snedecor (1962); is there anything more recent?

Reply 1: In most circumstances where I have seen this phenomenon, the data are strictly positive, thus implying a possible relationship between the means and variances. If this is true for your problem, you could consider a models assuming a constant coefficient of variation (see McCullagh and Nelder, 1989).

Reply 2: Relying on a frequentist *p*-value can't be the right thing to do as this depends on the size of the data set. With large data sets one will find 'significant' but unimportant differences. If cellwise variances differ by more than about four to five fold there is cause for concern. A formal ANOVA should be used only after one has looked at the data graphically.

Further discussion: This problem stimulated a general discussion on Genstat's capability for user computations, exploratory analysis and graphics. If you are interested, send the message

```
SEND Genstat log9307
```

to **LISTSERV@IB.RL.AC.UK**. This facility allows previous discussions to be retrieved, in monthly collections (here, for July 1993).

Zero-inflated Poisson models

Query: We are interested in modelling count data with excess zero values. Has anyone developed a procedure for this?

Reply 1: I have used a general ML fitting process for this, though not in Genstat. There is a useful paper which provides good starting estimates: Kemp and Kemp (1988) *Statistician* 37, 243–255.

Reply 2: I have implemented Lambert's EM algorithm for these models in Genstat. We plan to submit it to the Newsletter, but can make it available to anyone interested.

Further discussion: The originator let out that the model was to be used to model the occurrence of Leadbeater's Possum, an endangered species; whereupon the discussion degenerated, with reference to Possum distributions and the relative numbers of people/sheep/possums in New Zealand! But it was established that other names for this model are ZIP regression, and Poisson–with–added–zeros.

Pointer labels

Query: I am having trouble getting access to the labels of a pointer. The **GETATTRIBUTE** directive does not provide this information when I set the **ATTRIBUTE** option to **labels**.

Reply: There is a bug in **GETATTRIBUTE** in Release 2: some of the attributes of pointers cannot be accessed. This has been fixed in Release 3.1, but in Release 2 you can get the information using **DUMP**.

Repeated measures ANOVA

Query: How can I do a simple, single test to establish whether there is any interaction in a repeated measure ANOVA? The **ANTORDER** and **ANTTEST** procedures seem to be concerned with main effects rather than interactions.

Reply: We have written a procedure for analysing repeated measures data in the manner of Chapter 5 of Diggle (1990) *Time series: a biostatistical introduction.* It has the facility for doing most of what Diggle suggests, when each unit is sampled at all the same times and there are no missing values. It allows one treatment factor and no higher blocking strata. We can make it available, but there is no library–style documentation yet.

Randomization

Query: I am using **RANDOMIZE** to get several randomizations of a lattice design. What is the best way to update the seed each time?

Reply 1: I usually just add 1 to the seed each time and this seems to generate independent randomizations. I did once write some code to randomize the seed at each pass of a loop:

```
VARIATE [VALUES=10(0...9)] sb
RANDOMIZE [SEED=12345] sb
FOR [NTIMES=1000]
  CALC seed = sb$[1]*10000+\
    sb$[2]*1000+sb$[3]*100+\
    sb$[4]*10+sb$[5]
  ...
  RANDOMIZE [SEED=seed] sb
ENDFOR
```

Reply 2: I have used:

```
SCALAR [VALUE=579462] seed
CALC seed = URAND(seed)
& seed = INT(seed*1000000)+1
FOR ...
  CALC seed = URAND(0)
  & seed = INT(seed*1000000)+1
  RANDOMIZE [SEED=seed] ...
ENDFOR
```

Reply 3: I faced this issue recently, where I wanted the same design at a number of sites, so I just nested the design within the sites, as follows:

```
RANDOM [BLOCK=site/year/ \
  (row*col)/subplot; SEED=389] \
  year,season,cut
```

where **year** is a block factor, **season** is randomized into a Latin square in the **row*col** stratum, and **cut** is a subplot treatment.

Teaching t-tests

Query: I am preparing a course that will teach basic stats using Genstat, going as far as some simple ANOVA. I want to cover the *t*–test to introduce *p*–values. The problem is that the **TTEST** procedure operates with the two samples in different variates and **ANOVA** operates with all data in one variate. Has anyone any suggestions?

Reply 1: You could modify the code of **TTEST** to accept a single variate and a factor.

Reply 2: We have just run our first Introductory Stats course at Rothamsted, and based it entirely on the Menu System – customized for the occasion. In particular the Test Menu will take data either from two variates or from one variate and a factor. It will go out with Release 3.

Reply 3: The **TTEST** procedure has been modified for Release 3 to do precisely what is required. Details are available if anyone wants to apply the changes for Release 2.

A summary of new facilities in Genstat 5 Release 3

Roger Payne
AFRC IACR Rothamsted Experimental Station
Harpenden, Herts AL5 2JQ, UK

This article is based on the information given by procedure **NOTICE**. Full details of Release 3 can be found in the new Genstat 5 Release 3 Reference Manual (Payne *et al* 1993) published by Oxford University Press.

1. New facilities

Release 3 is a major upgrade with many important extensions.

The regression section now caters for generalized additive models. These allow variates to be fitted whose contributions cannot be modelled by any specific function but need to be fitted by non-parametric shapes, such as splines. This is achieved by using a function of the variate in the formulae in **FIT**, **ADD**, **TERMS** etc, instead of the variate itself. For example

```
FIT SSPLINE(X; 4)
```

will fit a smooth spline with approximately 4 degrees of freedom for the effect of **X**. Ordered categorical data can also now be analysed, by specifying a list of *y*-variates in **MODEL** (one for each category) and setting options **DISTRIBUTION=multinomial** and **YRELATION=cumulative**. Polynomial and other contrasts can be fitted using the **REG** and **POL** functions as in **ANOVA**. Finally, among more minor changes, **PREDICT** now requires very much less workspace to form means and standard errors, and the **PRINT** option has thus been changed to show standard errors by default.

In analysis of variance, Genstat now automatically combines information on treatment terms that are estimated in more than one stratum. These can be printed by setting option **PRINT** of **ANOVA** or **ADISPLAY** to **cbmeans** or **cbeffects** for combined means and combined effects respectively. The **AKEEP** directive has also been updated to allow all this information to be saved within Genstat. Sums of squares for covariates are split to show the contribution of each individual covariate in turn, and rigorous checks are now made for partial aliasing in orthogonal designs, which will increase the protection against incorrectly specified models.

The multivariate directives will now define their output structures by default, if necessary, to be structures of the appropriate shape and type, and there is more flexibility in the input structures that are allowed.

The **REML** facilities have been extended to allow testing of fixed effects, either by Wald tests (by setting option **PRINT=Waldtests** in **REML**) or by the more accurate likelihood tests (by setting option **SUBMODEL** and **PRINT=deviance** in **REML**). It is now possible to estimate negative variance components (**CONSTRAINTS=none** in **VCOMPONENTS**) or to impose linear constraints on the components (by the new option **RELATIONSHIP** in **VCOMPONENTS**). Fixed correlations can also be specified between the levels of a random factor, and directive **VKEEP** can now save variance–covariance matrices of fixed effects.

New statistical facilities include the estimation of parameters of statistical distributions, and many new probability and distribution functions for use, e.g., in **CALCULATE**.

In the more general areas, Release 3 should produce clearer fault messages. The key combination **Control-C** can be pressed to interrupt long analyses or long streams of output, with the opportunity then to continue or to abandon the statement concerned. The character **#** can be used in option settings to indicate the default setting:

```
RDISPLAY [PRINT=#,fitted]
```

sets the **PRINT** option to **model,summary,estimates,fitted** (where the first three comprise the default). Range checks can be requested for data structures using the new parameters **MINIMUM** and **MAXIMUM** of **VARIATE**, **MATRIX** etc, and the new directive **DUPLICATE** allows new structures to be defined with attributes identical to those of existing structures.

The facilities for high-resolution graphics have been extended to include three-dimensional histograms (new directive **D3HISTOGRAM**). Tables of data supplied to **DHISTOGRAM** can now specify bars with non-integral and negative heights, and the lines around the bars are now drawn after the shading takes place to avoid the blurring of bar boundaries that could previously occur when plotting interactively. Greater control is allowed over the form and labelling of axes, and more windows, pens and colours are provided. The **FRAME** directive now allows the background colour of the graphical windows to be specified for most interactive colour devices and for PostScript output, and the **PEN** directive has been extended to cater for user–defined symbols and to allow labels to be printed alongside the plotting symbols. A new directive **DDISPLAY** allows the current graphical display to be redrawn on most interactive devices, and the new directive **DKEEP** together with the new parameter **SAVE** of **AXES**, **FRAME** and **PEN** allow details of the current graphics environment to be saved.

There have been many improvements to **READ**. More informative prompts are produced when reading interactively, and it is easier to recover after data errors. If the length of any vector (factor, text or variate) has not already been defined before attempting to read its values, this will now be set automatically by **READ**: to the default length specified by an earlier **UNITS** directive if available, otherwise to the same length as any vectors of known length that are being read in parallel or, failing that, according to the number of units present in the data. The new option **SETLEVELS** allows factors to be defined automatically, and summaries are now produced for factors, texts and scalars as well as for variates. Factor labels and other strings need now be placed within quotes only if they contain separators. Tabs can be used to separate data values except when reading in fixed format.

The printing of tables has been simplified. **PRINT** has new options **DOWN**, **ACROSS** and **WAFER** to replace **NDOWN**, **INTERLEAVE** and **PERMUTE**. The **CLOSE** directive has a new parameter **DELETE** to allow the deletion of temporary files, and it can also now close texts. The new **ENQUIRE** directive allows information to be obtained about the files open on each channel; this will be particularly useful for writers of procedures.

In data manipulation, the new **GROUPS** directive takes over the formation of factors from **SORT**, which now caters for multiple indexes. **GROUPS** is very much more flexible in the definition of levels and labels, and also allows existing variates and texts to be redefined as factors. The **COMBINE** directive now allows slices of tables to be placed into tables of smaller dimensions. Again intended for procedure writers, **FCLASSIFICATION** allows the full list of factors in a formula to be obtained (new option **CLASSIFICATION**) and, with new option **METHOD=preserve**, the **ASSIGN** directive will assign values only to the dummies that are not already set.

Also for procedure writers, the **EXECUTE** directive allows the contents of a text to be executed (as a list of Genstat commands) at the time that the procedure is used. A new option **RESTORE** of the **PROCEDURE** directive itself allows various aspects of the Genstat environment to be restored automatically, and setting the new option **PARAMETERS=pointers** will cause all the settings of each parameter to be available at once within the procedure, in a pointer rather than a dummy. A new option **LIST** of the **OPTION** directive controls whether each option is to expect a single setting or a list (which would then be put into a pointer). The default **LIST=no** should cover most existing situations, but putting **LIST=yes** will greatly simplify the handling of options that expect a list of one or more identifiers; it will also need to be set for options that expected a list of strings, and this is one directive where there may be incompatibilities with Release 2. However, problems that are encountered when writing a procedure may now be found more quickly; by putting **DEBUG [FAULT=yes]** Genstat can be requested to break when it hits the next fault.

Again for the advanced user, new directives **SETOPTION** and **SETPARAMETER** allow the default values of options and parameters of either directives or procedures to be modified; these will be especially useful in start-up files.

With all these changes, there will inevitably be some incompatibilities with existing Genstat programs, as new facilities cause a reassessment of the Release 2 syntax, and the opportunity is taken to smooth out contradictions that have come to light in the Release 2 syntax. However, most programs should continue to run exactly as in Release 2, and where there is an incompatibility it will produce a fault rather than incorrect results! Details of the incompatibilities are listed at the end of this section, advice on changes that may need to be made to general programs and procedures are in Section 2.4, and full details of the new syntax can be obtained using the Genstat **HELP** directive.

1.1 New Directives in Release 3

DDISPLAY	redraws the current graphical display
DISTRIBUTION	estimates the parameters of continuous and discrete distributions
DKEEP	saves information from the last plot on a particular device
DUPLICATE	forms new data structures with attributes taken from an existing structure
D3HISTOGRAM	produces 3-dimensional histograms
ENQUIRE	provides details about files opened by Genstat
EXECUTE	executes the statements contained within a text
GROUPS	forms a factor (or grouping variable) from a variate or text, together with the set of distinct values that occur
SETOPTION	sets or modifies defaults of options of Genstat directives or procedures
SETPARAMETER	sets or modifies defaults of parameters of Genstat directives or procedures

1.2 Incompatibilities with Release 2

ADISPLAY	option **SE** renamed **PSE**; parameters **RESIDUALS** and **FITTEDVALUES** removed (these are now saved by **AKEEP**)
ANOVA	option **SE** renamed **PSE**
AXES	options **YINTEGER** and **XINTEGER** deleted
DEVICE	setting **ENDACTION=unchanged** removed (in all directives concerned with setting the graphics environment, options of parameters that are not set remain as before)
DCONTOUR, DGRAPH, DHISTOGRAM and **DPIE**	the default for the **PEN** parameter will now use pens 1, 2, etc for the successive graphs
FITNONLINEAR	**CALCULATION** option cannot now be set to a pointer, but must be set to a list of expressions
FSIMILARITY	**TEST** parameter now has strings as its settings
FOR	option **COMPILE** removed
GET	option **FAULT** now saves the textual form of the code (e.g., **'AN 1'**)
HLIST	**TEST** parameter now has strings as its settings
HSUMMARIZE	**TEST** parameter now has strings as its settings
OPTION	new parameter **LIST** to control whether or not each option expects a list of settings; the default of **no** covers most situations, but **LIST** will need to be set to **yes** for example for **PRINT** options that allow more than one setting.
PRINT	options **ACROSS**, **DOWN** and **WAFER** replace **PERMUTE**, **INTERLEAVE** and **NDOWN**
READ	settings **both** and **neither** of the **JUSTIFIED** option are deleted (replaced by putting **JUSTIFIED=left,right** and **JUSTIFIED=***, respectively)
RELATE	**TEST** parameter now has strings as its settings
RFUNCTION	**CALCULATION** option cannot now be set to a pointer, but must be set to a list of expressions
SORT	options **GROUPS**, **LIMITS**, **NGROUPS**, **LEVELS** and **LABELS** removed (the definition of factors from variates and texts is taken over by the new directive **GROUPS**)
VDISPLAY	option **CHANNEL** inserted before **PTERMS**

There have also been minor changes in the ordering of options or parameters within the following directives, but these will cause no problems provided options and parameters beyond the first two are named: **AKEEP**, **ANOVA**, **AXES**, **ESTIMATE**, **FCLASSIFICATION**, **MODEL**, **OPEN**, **PEN**, **PRINT**, **PROCEDURE**, **REML**, **RKEEP**, **SET**, **VKEEP**. These arise mainly from the addition of new options/parameters; full details are obtainable using Genstat **HELP** directive.

The new **ANOVA** facilities for combination of information on effects estimated in more than one stratum have required changes to the **DESIGN** structures. As a result, **DESIGN** structures that have been saved by backing store from Release 2 cannot be reused in the **DESIGN** option of **ANOVA** in Release 3. However, the design can still be recovered using the fact that the **DESIGN** structure is a pointer in which element 10 stores the block formula, and element 11 stores the treatment formula. The only aspect that is not stored is the setting of **FACTORIAL** used in the original **ANOVA**, but this should be easy to obtain from the earlier output. For example, assuming that

FACTORIAL was not set when the **DESIGN** structure **R2des** was formed, to form a new Release 3 **DESIGN** structure called **R3des** requires

```
BLOCKSTRUCTURE    #R2des[10]
TREATMENTSTRUCTURE #R2des[11]
ANOVA [DESIGN=R3des]
```

Sums of squares in analyses of covariance may differ from those in previous releases. As Preece (1980) has pointed out, the usual method of adjusting the sums of squares of a treatment term for covariates may involve adjusting the term also for higher-order terms to which it is marginal. Release 3 contains a new algorithm which avoids this deficiency.

2. Changes to the Procedure Library

Release 3[1] of the procedure library contains 22 new procedures.

2.1 New procedures in Library version 3[1]

AGALPHA	forms alpha designs by standard generators for up to 100 treatments
AGCYCLIC	generates cyclic designs from standard generators
AGDESIGN	generates generally balanced designs
AGFRACTION	generates fractional factorial designs
AGHIERARCHICAL	generates orthogonal hierarchical designs
BOXPLOT	draws box-and-whisker diagrams or schematic plots
CHISQUARE	calculates chi-square statistics for one- and two-way tables
DESIGN	helps to select and generate effective experimental designs
DMST	gives a high resolution plot of an ordination with minumum spanning tree
DREPMEAS	plots profiles and differences of profiles for repeated measures data
FACAMEND	permutes the levels and labels of a factor
FDESIGNFILE	forms a backing-store file of information for **AGDESIGN**
FILEREAD	reads data from a file, assumed to be in a rectangular array
GRANDOM	generates pseudo-random numbers from probability distributions
PCOPROC	performs a multiple Procrustes analysis
PPAIR	displays results of *t*-tests for pairwise differences in compact diagrams
PREWHITEN	filters a time series before spectral analysis
PTDESCRIBE	gives summary and second order statistics for a point process
RJOINT	does modified joint regression analysis for variety-by-environment data
RUNTEST	performs a test of randomness of a sequence of observations
VORTHPOL	calculates orthogonal polynomial time-contrasts for repeated measures
XOCATEGORIES	performs analyses of categorical data from crossover trials

2.2 Obsolete Procedures

However, 18 of the earlier procedures are now obsolete, either because of the changes within Genstat 5 Release 3 itself, or because they have been superseded by other procedures in the Library:

BINOMIAL	calculates probabilities from the binomial distribution
GRBETA	generates pseudo–random numbers from a beta distribution
GRCHI	generates pseudo–random numbers from the chi–squared distribution
GRF	generates pseudo–random numbers from the F distribution
GRGAMMA	generates pseudo–random numbers from the gamma distribution
GRLOGNORMAL	generates pseudo–random numbers from the log–Normal distribution
GRNORMAL	generates pseudo–random numbers from the Normal distribution
GRT	generates pseudo–random numbers from Student's *t*–distribution
GRWEIBULL	generates pseudo–random numbers from the Weibull distribution
INVNORMAL	calculates probabilities from the inverse Normal distribution
LOGNORMAL	calculates probabilities from the lognormal distribution

MANCOVA	performs a multivariate analysis of covariance
NPCHECK	checks the validity of input data for nonparametric procedures
ORDINALLOGISTIC	fits McCullagh's ordinal logistic regression model
POISSON	calculates probabilities from the Poisson distribution
STUDENT	calculates probabilities from Student's *t*–distribution
VWALD	prints Wald tests for fixed terms in a REML analysis
WHISKER	produces box–and–whisker diagrams

These will remain available, in the **obsolete** module of Release 3[1] of the Library, but will be deleted after that.

2.3 Changes in Syntax of Procedures

There are also various changes in the syntax of some of the existing procedures, to retain compatibility with the syntax of directives in Release 3 and to improve ease of use. It will no longer be necessary to form pointers to use the non–parametric procedures.

The two–sample procedures **KOLMOG2**, **MANNWHITNEY**, **SIGNTEST** and also **TTEST** now have parameters **Y1** and **Y2** which can be used to specify the samples in two separate variates; alternatively you can put all the data in a single variate (specified by **Y1**) and then identify the members of the samples by a factor, specified by the new **GROUPS** option. **CONCORD**, **KRUSKAL** and **SPEARMAN** are also modified to expect a list of variates from the **DATA** parameter, or just one variate and the **GROUPS** option to be set to a factor to define the samples; the test statistics and associated details are now saved by options.

Procedures **ALIAS**, **ANTORDER**, **ANTTEST**, **MANOVA** and **REPMEAS** also now expect lists instead of pointers, and **SUBSET** now allows only lists.

The setting **highquality** is renamed **highresolution** in procedures **BJESTIMATE**, **BJFORECAST**, **DDENDROGRAM**, **SMOOTHSPECTRUM** and **VPLOT**, and option **SE** is renamed **PSE** in **NLCONTRAST** (as in **ANOVA** and **ADISPLAY**).

Extensions include the ability to transform axes in **DBARCHART**, and further links and distributions in **GLMM**.

As in previous releases, details of the syntax of all the Library procedures can be obtained from within Genstat using procedure **LIBHELP**, as explained on page 239 of the Genstat 5 Release 3 Reference Manual. Writers of procedures for the Library are reminded that Instructions for Authors were published in Genstat Newsletter 20. They also can be obtained from the Secretary of the Genstat Procedure Library Editorial Committee (c/o Statistics Department, Rothamsted Experimental Station, Harpenden, Herts AL5 2JQ, United Kingdom).

2.4 Conversion changes to existing procedures

When converting procedures to Release 3, the directives that are most likely to require changes are (in order): **OPTION**, **SORT**, **FITNONLINEAR**, **EXIT**, **ADISPLAY**, **ANOVA** and the clustering directives **FSIMILARITY**, **HLIST**, **HSUMMARIZE** and **RELATE**.

The **LIST** parameter of **OPTION** must be set to **yes** for any option of mode **t** that allows more than one setting.

The use of the **SORT** directive to define factors from texts is taken over by the new directive **GROUPS**. The **LMETHOD** option of **GROUPS** gives very much more flexibility in the formation of levels and labels than existed in **SORT**. If options **LABELS** and **LEVELS** were not set in the original **SORT** statement, **GROUPS** requires **LMETHOD=*** so that no vectors of levels or labels are formed; any existing settings for the factor are then retained. Otherwise, the default **LMETHOD=median** will form levels/labels from group medians regardless of whether any named structures are supplied to store them by the **LEVELS** and **LABELS** parameters of **GROUPS**. Other settings of **LMETHOD** allow them to be formed from minima, maxima or from the structures supplied by **LEVELS/LABELS**. The other difference is that, following experience with **SORT**, limits by default are taken to supply lower boundary values; upper values require option **BOUNDARIES=upper**. Also, the **REDEFINE** option

allows a text or variate to be redefined as a factor.

Some simple examples:

```
SORT [INDEX=i; GROUPS=f]                    ->   GROUPS [LMETHOD=*] i; FACTOR=f
SORT [INDEX=i; GROUPS=f; LEVELS=v]          ->   GROUPS i; FACTOR=f; LEVELS=v
                                                 (if f and v are needed later)
SORT [INDEX=i; GROUPS=f; NGROUPS=n]         ->   GROUPS [LMETHOD=*; NGROUPS=n] i; \
                                                   FACTOR=f
SORT [INDEX=i; GROUPS=f; LIMITS=l]          ->   GROUPS [LMETHOD=*; BOUNDARIES=upper]\
                                                   i; FACTOR=f; LIMITS=l
```

If several calculations are to be specified for the **EXPRESSION** option of **FITNONLINEAR**, the relevant expression structures need no longer be placed into a pointer, but must be given as a list.

The **FAULT** option of **GET** now supplies the fault code as a string rather than as a number, thus allowing greater legibility of programs and the ability to add new diagnostics without the need to renumber. So, for example,

```
GET [FAULT=Diag]
EXIT [REPEAT=yes] Diag .EQ. 224
```

would become

```
GET [FAULT=Diag]
EXIT [REPEAT=yes] Diag .EQS. 'SP 4'
```

in Release 3, where **Diag** is now a single-valued text.

Residuals and fitted values are no longer saved by **ADISPLAY**, as was the case (illogically) in Release 2, but by options **RESIDUALS** and **FITTEDVALUES** of **AKEEP**. The **SE** option of **ANOVA** and **ADISPLAY** has been renamed **PSE** to avoid the confusion arising from the use of **SE** to save standard errors in **PREDICT**.

In the clustering directives **FSIMILARITY**, **HLIST**, **HSUMMARIZE** and **RELATE** the **TEST** parameter now expects (more meaningful) strings rather than the integer codes that are used in Release 2.

2.5 Improvements in Release 3 to Increase the Efficiency of Procedures

The new **PARAMETER** option of **PROCEDURE** allows a procedure to process the complete list of settings of the parameters in one call, which can greatly improve efficiency. You might also want to use the new **RESTORE** option to save having to **GET** and then **SET** environment or special structures that need to be reset at the end of the procedure.

Also useful is the new **METHOD** option of **ASSIGN**. Genstat **IF** blocks of the form

```
IF UNSET(RESIDUALS)
  ASSIGN Resids; RESIDUALS
ENDIF
```

can be replaced by

```
ASSIGN [METHOD=preserve] Resids; RESIDUALS
```

References

Payne R W, Lane P W, Digby P G N, Harding S A, Leech P K, Morgan G W, Todd A D, Thompson R, Tunnicliffe-Wilson G, Welham S J and White R P (1993) *Genstat 5 Release 3 Reference Manual* Oxford University Press, Oxford.

Preece D A (1980) Covariance analysis, factorial experiments and marginality *The Statistician* **29** 97-122.

Comparisons of some GLMM estimators for a simple binomial model

D Waddington[1], S J Welham[2], A R Gilmour[3] and R Thompson[1]

[1] *AFRC Roslin Institute (Edinburgh), Roslin, Midlothian, EH25 9PS, UK*
[2] *AFRC IACR, Rothamsted Experimental Station, Harpenden, Herts, AL5 2JQ, UK*
[3] *NSW Agriculture, Agricultural Research Institute, Orange, NSW, Australia*

Summary

Several methods have been proposed recently for obtaining estimates of parameters of Generalized Linear Mixed Models (GLMMs). Within these methods, estimation of fixed effects are either calculated conditional on random terms in the linear predictor (Schall 1991, Engel and Keen 1993), or by excluding the random terms from the linear predictor, as in the marginal models of Gilmour *et al* (1985) and Breslow and Clayton (1993). The performance of these models in estimating parameters for a one–way random effects model for binomial data has been examined by simulation. Biases in the estimates of variance components are investigated further for two of the methods.

1. Introduction

Two of the four models may be fitted routinely in Genstat using the procedure **GLMM** (Welham 1993). Model specification and output of results are similar to the **VCOMP**, **REML** and **VDISP** commands. The fitting algorithm for the GLMM models is analogous to that for generalized linear models. For data y with mean μ, the mean is related to the linear predictor by the link function g

$$g(\mu) = \eta = X\beta + Zu$$

where β represents the fixed effects and u is a vector of random effects var(u)=G, a function of the unknown variance components $\sigma_1^2,\sigma_2^2,\ldots$. A working dependent variate

$$z = X\beta + Zu + D(\mu)(y-\mu); \quad D(\mu) = \left(\frac{dg}{d\mu}\right)_\mu = \text{diag}\ \{\ g'(\mu_1),\ g'(\mu_2)\ldots\ \}$$

is created by linearising the link function applied to the data about their mean values. The working variate has three components: fixed effects, random effects and an 'error term' which depends on the distribution of y, and on the link function through $D(\mu)$. The first and third terms are those of a working variate for a standard generalized linear model. The second and third terms give

$$\text{var}(z) = ZGZ^T + DVD$$

where V is the variance of y, conditional on the random effects u, and DVD is a diagonal matrix. Thus the working variate is described by a linear mixed model with fixed effects β, random effects u and weights $W = (DVD)^{-1}$. Given an estimate of μ, the standard mixed model equations can be used to estimate β and u:

$$\begin{pmatrix} X^TWX & X^TWZ \\ Z^TWX & Z^TWZ+G^{-1} \end{pmatrix}\begin{pmatrix} \beta \\ \hat{u} \end{pmatrix} = \begin{pmatrix} X^TWz \\ Z^TWz \end{pmatrix}.$$

The Genstat REML algorithm will solve these equations and produce estimates of the variance components in G. It has been found that restricting the REML algorithm to two iterations to provide an approximate solution, rather than allowing it to converge, does not affect the speed of convergence of the GLMM algorithm and saves computing time. Given new estimates of β and u, further estimates of μ are formed and used to update the working variate and weights (as in generalized linear models), and the estimation is repeated until convergence.

The method of Schall uses an estimate of the conditional mean of y given μ, that is

$$\hat{\mu} = g^{-1}(X\hat{\beta} + Z\hat{u}) .$$

This method has been proposed by several authors previously, for example Gianola and Foulley (1983). It corresponds to the 'subject specific model' of Zeger *et al* (1988), and is the default for the Genstat **GLMM** procedure. The model may also be derived as an approximation to the likelihood, assuming the random effects u to be normally distributed (Breslow and Clayton 1993).

The algorithm for the marginal model of Breslow and Clayton is identical except for the estimation of the mean as $\hat{\mu} = g^{-1}(X\hat{\beta})$. This model may be considered as an approximation to that of Schall when the σ_i^2 are small, and is the 'population–averaged' model of Zeger *et al* (1988). It is fitted by the **GLMM** procedure using the option **FMETHOD=fixed**.

A modification to the Schall model has been proposed by Engel and Keen (1993). They point out that the weights W are functions of the random effects u, and suggest that the variance of z is more correctly obtained by integrating out the u terms to obtain $E_u(W^{-1})$. The weights $E_u(W^{-1})$ then take different forms for combinations of error distributions, links and random models. Estimation is then by quasi–likelihood, which is equivalent to the iterative procedure outline above, but with a modified weight.

The method of Gilmour *et al* (1985), initially proposed for binomial data only, also takes a quasi–likelihood approach. They obtain the mean and an approximation to the variance of the observations y by taking expectations over u of the first two derivatives of the log–likelihood of y conditional on u. The mean is therefore the marginal mean, and the variance of the working variate z has the same contribution from the random terms u as all the other models. The difference from the marginal model of Breslow and Clayton lies in the expectation of the conditional binomial variances V, which are used to form the iterative weights W corresponding to the error term in z. The random effects on the scale of z induce a covariance between the binary elements of an observation y_i, leading to a reduction in the conditional variance of y_i. This will be elaborated in Sections 3 and 5.

2. Simulation details

Data were simulated from a one–way random effects model in which 512 individuals with binary observations (e.g. affected/not affected) were grouped into families of size N = 2, 4, 8 or 16. Using the logit link function, if the probability of an individual being affected is θ (θ = 0.05, 0.1, 0.2, 0.35 or 0.5) then the equivalent threshold on the logistic scale is $\eta = \log(\theta/(1-\theta))$. This threshold is then modified by adding the random effect of family k, $u_k \sim N(0, \sigma^2)$ to give $\eta'_{ik} = \log(\theta/(1-\theta)) + u_k$ for the ith individual in the kth family. This addition of extra noise changes the average probability of an individual being affected to approximately $\exp(\eta'_{ik})/(1 + \exp(\eta'_{ik}))$. Five different variances σ^2 were used, corresponding to a wide range of intra–class correlations ρ:

σ^2	= 0	0.4	1	4	16
ρ	= 0	0.11	0.23	0.44	0.83

where

$$\rho = \frac{\sigma^2}{\pi^2/3 + \sigma^2}$$

and $\pi^2/3$ is the variance of the logistic distribution corresponding to the link function.

There were 100 runs of each of the one hundred parameter combinations. Only the briefest details of the results can be given here, but the overall conclusions are sufficiently clear. For the method of Schall, the Genstat GLMM procedure was abbreviated to fit only the one–way random effects model, for family sizes of 4, 8 and 16. Problems of run time and storage space forced a different approach for families of size 2. Sufficient statistics for fitting the model are the numbers of families with either 0, 1 or 2 members affected, and the EM approach given in Schall's paper was programmed to use such data. This approach was also used for all

simulations fitted by the method of Engel and Keen.

For the method of Breslow and Clayton, the estimate of θ is simply the observed proportion affected from the 512 individuals. For this model, the estimate of σ^2 is obtained from a one–way ANOVA on the binary data from the approximation

$$E(\text{Between family mean square}) \approx \sigma_e^2 + N\left(\frac{d\theta}{d\eta}\right)^2\sigma^2$$
$$= \sigma_e^2 + N\,\hat{\theta}^2(1-\hat{\theta})^2\sigma^2$$

where σ_e^2 is the within family variance component. For the method of Breslow and Clayton the estimated within family variance of a binary observation is $\hat{\sigma}_e^2 = \hat{\theta}(1-\hat{\theta})$, giving

$$\hat{\sigma}_B^2 = \frac{\text{Between family mean square} - \hat{\theta}(1-\hat{\theta})}{N\hat{\theta}^2(1-\hat{\theta})^2}.$$

Similarly for the method of Gilmour *et al*, the estimate of the mean is the same, and that of the variance is

$$\hat{\sigma}^2 = \frac{N}{N-1}\hat{\sigma}_B^2$$

because the estimate of σ_e^2 is $\pi_{01}/2 \approx \hat{\theta}(1-\hat{\theta}) - \hat{\theta}^2(1-\hat{\theta})^2\sigma^2$ (see Section 4.3 for details). All negative estimates of $\hat{\sigma}^2$ were reset to 10^{-5}.

3. Results

3.1. Means

Tables 1a–c are for illustration of the following descriptions.

The mean values of $\hat{\theta}$ for the two marginal models for family size $N = 16$ are given in Table 1a. All are within 5% of the approximation

$$\hat{\theta} = \text{logit}^{-1}\left(\frac{\eta}{(1+0.35\sigma^2)^{0.5}}\right)$$

given by Williams (1988), irrespective of family size. The four tables generated by the marginal models for different family sizes 2, 4, 8 or 16 differ from each other by less than 0.001. The dramatic attenuation of estimates generated from low values of θ towards 0.5 (zero on the logit scale) caused by large variance components can be seen. It is worth restating that these are the means of the *observed* proportions affected. This is the effect of adding considerable symmetric noise on the scale of the working variate, which is then transformed asymmetrically by the function logit^{-1} onto the observed binomial scale. These means are appropriate for predicting population responses.

The estimates of θ from the method of Engel and Keen were very similar to the marginal estimates of Gilmour/Breslow for smaller family sizes. For $N = 16$ and $\hat{\sigma} > 4$ they were slightly less attenuated (Table 1b). Similarly for the Schall estimates, the larger the family size the closer $\hat{\theta}$ was to θ. For $N = 2$ they were similar to the Engel estimates with $N = 16$, and mean values for $N = 16$ are given in Table 1c. There is still some attenuation as σ^2 increases. Zeger *et al* (1988) suggest that $\hat{\theta}$ might be interpreted as the probability for a 'typical' family.

Table 1. Means of $\hat{\theta}$ from 100 simulations, for some combinations of methods and parameters

Table 1a. Gilmour/Breslow

$N = 16$

	θ			
σ^2	.05	.1	.2	.5
0.4	.06	.12	.22	.5
1.0	.07	.13	.24	.49
4.0	.13	.20	.30	.5
16.0	.25	.31	.38	.49

Table 1b. Engel

$N = 16$

	θ			
σ^2	.05	.1	.2	.5
0.4	.06	.11	.21	.5
1.0	.07	.12	.22	.49
4.0	.11	.17	.26	.5
16.0	.21	.27	.35	.49

depend on $\theta^{1/2}$?

Table 1c. Schall

$N = 16$

	θ			
σ^2	.05	.1	.2	.5
0.4	.06	.11	.21	.5
1.0	.06	.12	.21	.49
4.0	.08	.13	.23	.51
16.0	.13	.19	.28	.49

The mean of the estimated variances of $\hat{\eta}$, var($\hat{\eta}$), for the 100 simulations for each parameter combination was compared with the observed variance of the estimates of η. No pattern in their ratios across the parameters was discernable, so their distributions over the 100 parameter combinations are given in Table 2. They were not calculated for Gilmour's method. On average, the estimates of the variance of $\hat{\eta}$ are quite acceptable.

Table 2. Distribution, across 100 parameter combinations, of the ratio of the mean of var $\hat{\eta}$ to the observed variance of $\hat{\eta}$, for 100 simulations for each parameter combination, where $\eta = \log(\theta/(1-\theta))$.

	Min	Lower Quartile	Mean	Upper Quartile	Max
Breslow	.72	.93	1.01	1.07	1.54
Engel	.56	.94	1.03	1.11	1.65
Schall	.58	.90	1.00	1.09	1.54

3.2 Variances

For a family size of two almost all methods underestimate the variances σ^2 for all underlying proportions θ (Table 3). The exceptions are the method of Gilmour for low values of both σ^2 and θ, and that of Breslow for $\theta = 0.05$ and $\sigma^2 = 0.4$. Although mean estimates of variance increase with σ^2 for these parameter combinations, all methods severely underestimate the highest value of $\sigma^2 = 16$.

Table 3. Ratios of the mean estimate of σ^2 over 100 simulations to the true values for all four methods and for some parameter combinations. Family size $N = 2$.

		Gilmour	Breslow	Engel	Schall
$\theta = 0.05$					
	0.4	2.4	1.2	0.2	0.7
	1	1.5	0.7	0.2	0.5
σ^2	4	0.7	0.3	0.1	0.3
	16	0.2	0.1	0.1	0.1
$\theta = 0.5$					
	0.4	0.9	0.4	0.3	0.5
	1	0.7	0.4	0.2	0.4
σ^2	4	0.4	0.2	0.1	0.3
	16	0.2	0.1	0.1	0.1

When the family size is large ($N = 16$) the estimates of Breslow and Gilmour converge (Table 4). There are similar, though less extreme, patterns for underestimation of σ^2 for $\theta = 0.05$ as were seen for families of size 2. The estimates from Engel's method are generally improved. When the underlying proportion θ increases to 0.5 all methods give similar average estimates for $\sigma^2 = 0.4$, and those from Engel's method are similar to the two marginal models for all four underlying variances. All three perform poorly in estimating large variances. The performance of Schall's method is notably improved for large σ^2 as family size increases, although the

underestimation is still considerable. The method of Gilmour is the only one which gives similar estimates for both family sizes when $\sigma^2 \geq 4$, and also for smaller values of σ^2 when $\theta = 0.5$.
There were no negative estimates of σ^2 when the true value was 4 or 16, for any of the methods. But for combinations of small σ^2 and θ there were considerable numbers of negative estimates of σ^2, particularly for the method of Schall, for all family sizes (Table 5). In contrast, larger family sizes resulted in fewer negative estimates for the other three methods. The constraining of negative estimates to the value 10^{-5} may be responsible for the overestimation of σ^2 in Gilmour's method.

Table 4. Ratios of the mean estimate of σ^2 over 100 simulations to the true value for all methods and for some parameter combinations. Family size $N = 16$.

		Gilmour	Breslow	Engel	Schall
$\theta = 0.05$					
	0.4	1.1	1.0	0.5	0.6
	1	1.0	0.9	0.5	0.7
σ^2	4	0.6	0.6	0.4	0.6
	16	0.2	0.2	0.2	0.5
$\theta = 0.5$					
	0.4	1.0	0.9	0.9	1.0
	1	0.7	0.6	0.7	0.9
σ^2	4	0.4	0.4	0.5	0.8
	16	0.2	0.2	0.2	0.5

Table 5. Numbers of negative estimates of σ^2 from 100 simulations for all four methods and for some parameter combinations.

		σ^2	Gilmour	Breslow	Engel	Schall
$N = 2$	$\theta = 0.05$	0.4	37	37	42	37
		1.0	15	15	15	15
	$\theta = 0.5$	0.4	7	7	8	10
		1.0	0	0	0	0
$N = 16$	$\theta = 0.05$	0.4	19	19	18	54
		1.0	1	1	1	11
	$\theta = 0.5$	0.4	0	0	0	0
		1.0	0	0	0	0

4. Additional Observations

4.1 Relationships of individual simulation estimates

All methods used the same random samples generated for each particular combination of parameters. The estimates for individual simulated data sets may therefore be compared. Scatter plots for parameter values N = 16, θ = 0.2 and σ^2 = 4 and 16 are shown as examples. Figure 1 compares the estimates of Engel and Breslow. The estimates for both thresholds η and variances σ^2 for Engel's method are approximately linear functions of the Breslow estimates. Slopes for both parameters are close to 1, confirming that the smoothing of the iterative weights in the method of Engel and Keen, by integrating out the random term, has produced model estimates similar to the marginal ones for these parameter combinations. Figure 2 compares the estimates of Schall and Breslow. The linearity between estimates is still apparent, but the slopes change more markedly with different combinations of parameters. These linear relationships hold for almost all of the 100 combinations of parameters used in the simulations.

Figure 1. Scatter plots of Engel estimates versus Breslow estimates for N = 16, θ = 0.2, σ^2 = 4 (*) and 16 (o). Top graph: Estimates of η. Bottom graph: Estimates of σ^2.

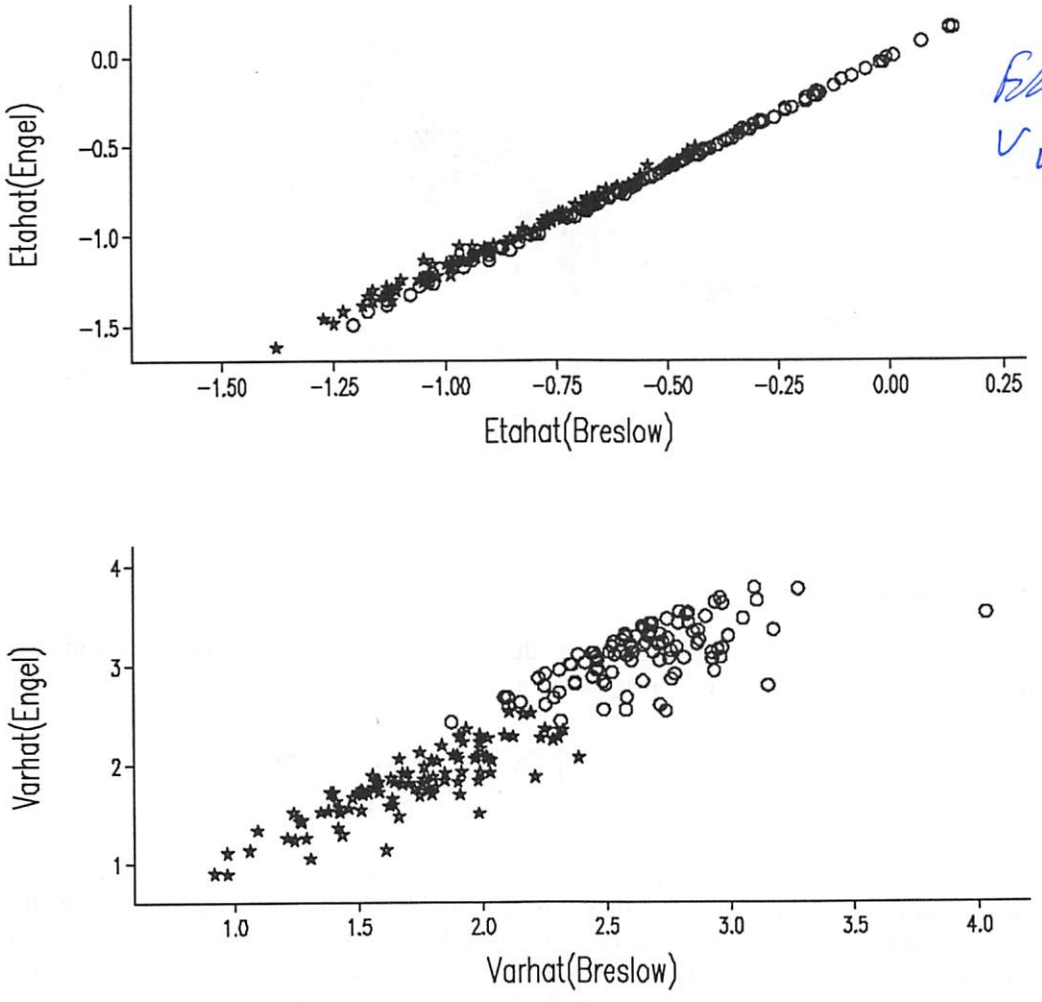
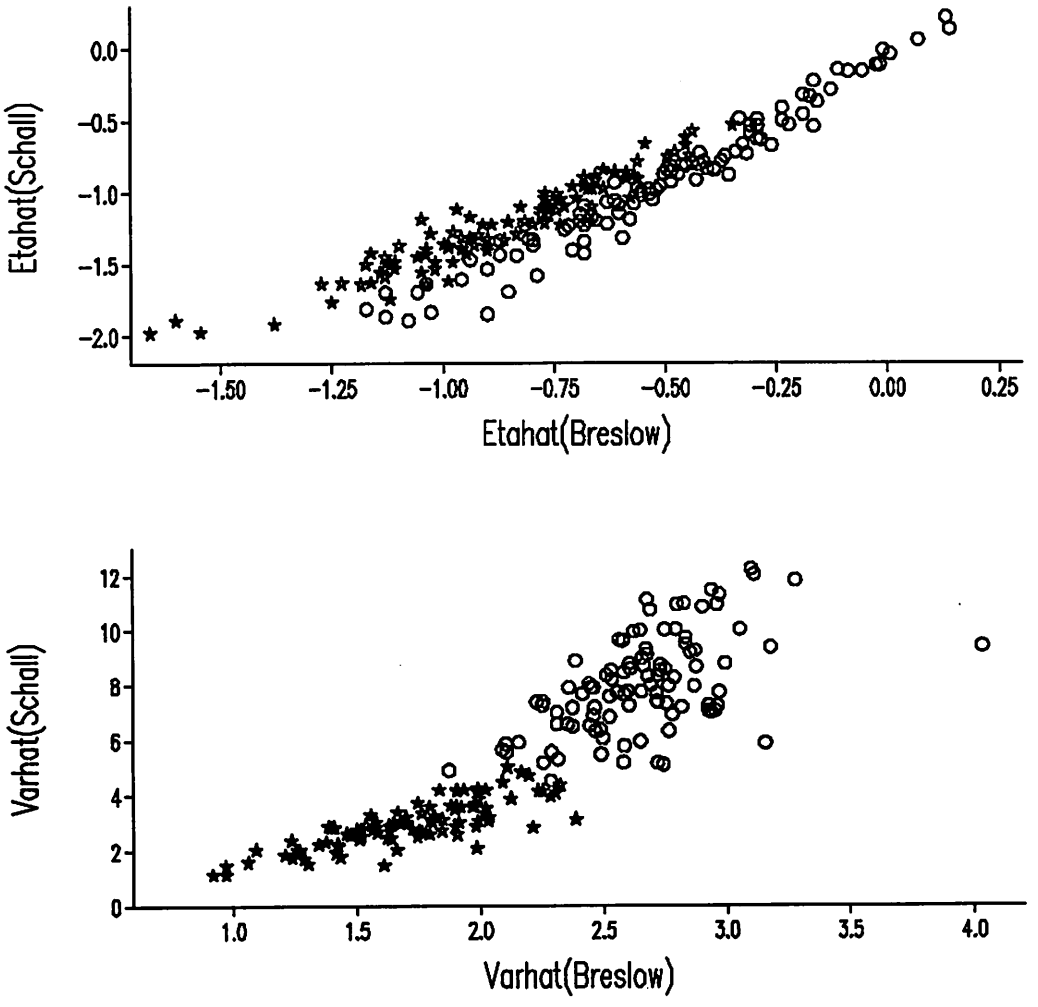


Figure 2. Scatter plots of Schall estimates versus Breslow estimates for N = 16, θ = 0.2, σ^2 = 4 (*) and 16 (o). Top graph: Estimates of η. Bottom graph: Estimates of σ^2.



4.2 Schall estimates of variance

The expected probability of observing r affected individuals from a family of size N may be readily calculated, using NAG Fortran routines for numerical integration, as

$$\pi_{rN} = \text{Prob}(R=r) = \int_{-\infty}^{\infty} \binom{N}{r} p^r q^{N-r} \frac{\exp(-u^2/2\sigma^2)}{(2\pi\sigma^2)^{0.5}}\, du$$

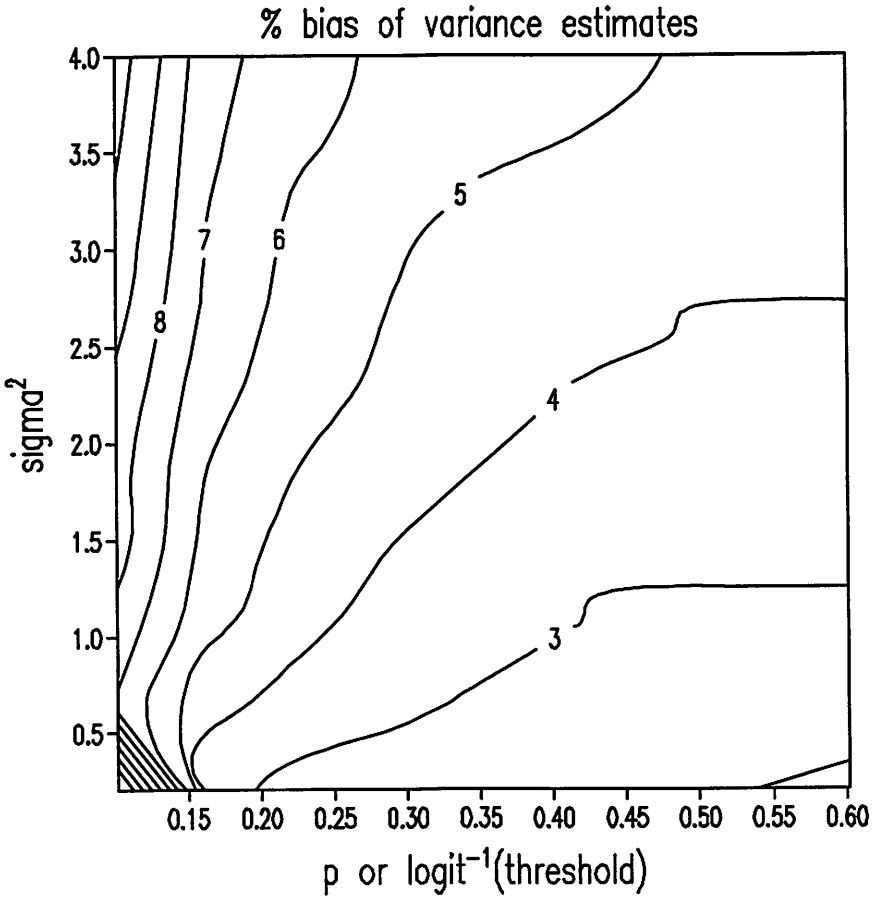
where r = 0...N and q = 1−p. The probability p in the integral is conditional, i.e. $p = \exp(\eta+u)/(1+\exp(\eta+u))$. To examine the asymptotic behaviour with numbers of families of the Schall estimates, the probabilities were multiplied by 10^5 and rounded. These frequency distributions for R, of 10^5 families of either size 2 or 16, were then used as input to the Schall estimation routine. The estimated variances are similar to those obtained from simulation, particularly for higher values of σ^2 and θ. Table 6 illustrates this for a family size of 16.

Table 6. Mean estimates of σ^2 for Schall's method from simulations ($\hat{\sigma}_{sim}$) and asymptotic estimates for 10^5 families with the expected frequency distribution of affected numbers ($\hat{\sigma}$), for two values of θ and family size $N = 16$.

	$\theta = 0.05$		$\theta = 0.5$	
$\hat{\sigma}$	$\hat{\sigma}^2$	$\hat{\sigma}^2_{sim}$	$\hat{\sigma}^2$	$\hat{\sigma}^2_{sim}$
0.4	0.32	0.24	0.36	0.41
1.0	0.73	0.67	0.86	0.85
4.0	2.43	2.41	3.00	3.11
16.0	6.88	7.18	7.92	7.89

For family size $N = 2$ there is considerable downward bias in the simulated estimates of σ^2 for all values of θ (Table 3) and the percentage bias increases with σ^2. The asymptotic behaviour is similar. When $N = 16$ the pattern is more complex (Figure 3). For θ near 0.5 and small values of σ^2 (0.2) the (negative) bias is less than 10%, and as σ^2 increases the bias increased only slowly. As θ becomes smaller the bias starts to increase more rapidly, and when both θ and σ^2 are small more rapidly still.

Figure 3. Percentage bias (negative) of Schall variance estimates for a range of values of θ (= p) and σ^2 (= sigma^2) for $N = 16$ in steps of 5% (level 1,2,3... = 5,10,15... % bias). Values calculated from the expected frequency distribution of numbers affected per family of 10^5 families.



4.3 Gilmour estimates of variance

The following argument applies to families of any size. Consider two individuals I_1 and I_2 in a family. The probability that they are both affected is $\tilde{p}^2$, where $\tilde{p}$ is the marginal estimate of θ. Similarly for the other combinations of one or both unaffected. The random effect for family on the logistic scale induces a covariance between members of a family which modifies these probabilities by an amount Δ, say. So we have:

		I_2: 1	I_2: 0
I_1	1	$\tilde{p}^2+\Delta$	$\tilde{p}\tilde{q}-\Delta$
	0	$\tilde{p}\tilde{q}-\Delta$	$\tilde{q}^2+\Delta$

The limits on Δ are 0, for $\sigma^2 = 0$, and $\tilde{p}\tilde{q}$, for $\sigma^2 \to \infty$, when the probability of the two individuals being affected becomes $\tilde{p}$.

Now, $\Delta \approx \left(\frac{d\theta}{d\eta}\right)^2_{\theta=\tilde{p}}$, $\sigma^2 = (\tilde{p}\tilde{q})^2\sigma^2$ and we can evaluate the probability of $I_1 = 0$, $I_2 = 1$ as $\pi_{12}/2$, given previously.

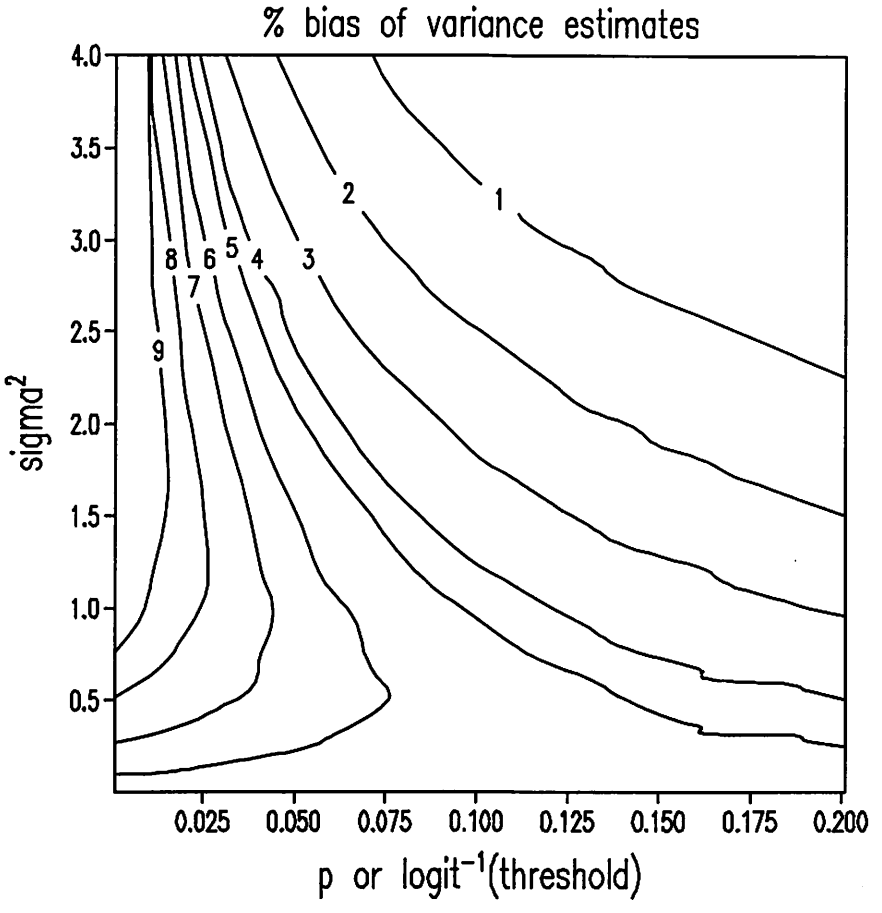
Then we have $\hat{\Delta} = \tilde{p}\tilde{q} - \pi_{12}/2$ and $\hat{\sigma}^2 = (\tilde{p}\tilde{q} - \pi_{12}/2)/(\tilde{p}\tilde{q})^2$.

Table 7. Mean estimates of σ^2 for Gilmour's method from simulations ($\hat{\sigma}^2_{sim}$) for $N = 16$, and estimates from the covariance between 2 members of a family ($\hat{\sigma}^2$), for two values of θ.

	$\theta = 0.05$		$\theta = 0.5$	
$\hat{\sigma}^2$	$\hat{\sigma}^2$	$\hat{\sigma}^2_{sim}$	$\hat{\sigma}^2$	$\hat{\sigma}^2_{sim}$
0.4	0.42	0.42	0.34	0.38
1.0	1.12	1.00	0.69	0.69
4.0	2.69	2.55	1.58	1.64
16.0	3.26	3.41	2.54	2.60

These estimates agree well with those from the simulations with $N = 16$, for example (Table 7). They show the same overestimation for small values of both θ and σ^2, and varying levels of underestimation as θ and σ^2 increase. Also, there is no bias in estimating the variance as $\sigma^2 \to 0$. This is illustrated in Figure 4, which also shows that as σ^2 increases the range of θ values within which $\hat{\sigma}^2$ is within ± 5% of σ^2 becomes increasingly small. For most combinations of θ and σ^2 the bias in $\hat{\sigma}^2$ is negative, but for very small values of θ the method will overestimate quite large variances.

Figure 4. Percentage bias of Gilmour variance estimates for a range of values of θ(= *p*) and σ^2 (= sigma^2) in steps of 10% (level 1,2... = –40,–30,–20,–10,–5,5,15,30,50 % bias). Values are calculated from the Δ approximation given in the text.



5. Conclusions

- For most parameter combinations the methods give linearly related estimates.
- The estimates of the mean values from the method of Keen and Engel are similar to marginal estimates, particularly for small family sizes and small values of σ^2. The marginal estimates correspond to the observed means.
- The average estimated variances of the threshold values are correct. However, this design has only one fixed effect parameter in the model. This means that there is essentially only one variance and one covariance to estimate. Models with several fixed effects may give poorer estimates of the variances of threshold values.
- The conditional mean models underestimate σ^2 for all parameter combinations. The bias of estimates of σ^2=0.4 is fairly small for all methods when family size is large and θ close to 0.5, but increases for larger values of σ^2. When θ is small Gilmour/Breslow, unusually, overestimate σ^2, more severely for small family sizes. This may be a result of constraining $\hat{\sigma}^2$ to be positive.

Acknowledgements

D Waddington and R Thompson gratefully acknowledge funding from the Ministry of Agriculture, Fisheries and Food, U.K.

References

Breslow N E and Clayton D G (1993) Approximate inference in Generalized Linear Mixed Models. *J. Amer. Statist. Assoc.* **88** 9–25.

Engel B and Keen A (1994) A simple approach for the analysis of Generalized Linear Mixed Models. *Statistica Neerlandica* **48** 1–22.

Gianola D and Foulley J L (1983) Sire evaluation for ordered categorical data with a threshold model. *Genet. Sel. Evol.* **15** 201–223.

Gilmour A R, Anderson R D and Rae A L (1985) The analysis of binomial data by a Generalized Linear Mixed Model. *Biometrika* **72** 593–599.

Schall R (1991) Estimation in Generalized Linear Models with random effects. *Biometrika* **78** 719–727.

Welham S J (1993) Procedure GLMM. In: *Genstat 5 Procedure Library Manual, Release 2[3]*, Eds. R W Payne and G M Arnold. Numerical Algorithms Group, Oxford.

Williams D A (1988) Extra–binomial variation in toxicology. *Proceedings of the Fourteenth International Biometric Conference, Namur, Belgium,* 301–313.

Zeger S L, Liang K Y and Albert P S (1988) Models for longitudinal data: a Generalized Estimating Equation approach. *Biometrics* **44** 1049–1060.

Maximum likelihood in a finite mixture model by exploiting the GLM facilities of Genstat

Ritsert C Jansen
Centre for Plant Breeding and Reproduction Research (CPRO-DLO)
PO Box 16, 6700 AA Wageningen
The Netherlands

1. Introduction

In this paper a general and flexible Genstat procedure for estimating the parameters of mixture models is described. The use of the procedure will be illustrated by means of a genetic example. In this example a quantitative trait is considered, which is controlled by a single, diallelic gene. It is assumed that the frequencies of the three possible (but unobserved) genotypes AA, Aa and aa conform to Hardy-Weinberg equilibrium; the genotype follows a multinomial distribution and the frequencies of the genotypes AA, Aa and aa will then be p^2, $2pq$ and q^2, respectively, where p and $q = 1\text{-}p$ are the frequencies of the two alleles A and a in the population. Furthermore, it is assumed that the quantitative trait is normally distributed $N(\mu_{AA},\sigma^2)$ for genotype AA, $N(\mu_{Aa},\sigma^2)$ for genotype Aa, and $N(\mu_{aa},\sigma^2)$ for genotype aa. Finally, it is assumed that $\mu_{aa} < \mu_{AA}$ and that the effects of the alleles are additive (no dominance), so that $\mu_{Aa}=(\mu_{AA}+\mu_{aa})/2$. The probability density function for the quantitative trait y is

$$f(y) = p^2 f(y;\mu_{AA},\sigma^2) + 2pq f(y;\frac{\mu_{AA}+\mu_{aa}}{2},\sigma^2) + q^2 f(y;\mu_{aa},\sigma^2)$$

where $f(\cdot\,;\mu,\sigma^2)$ is the probability density function of the Normal distribution with mean μ and variance σ^2.

2. Algorithm

Suppose that an individual's trait has been observed and that its value is equal to y. Its genotype cannot be observed, but it is either AA, Aa or aa. Thus the complete data would be either (y,AA), (y,Aa) or (y,aa). An individual with a small value of y is likely to have genotype aa. This might be expressed by assigning prior weights of, for instance, 0.1 to (y,AA), 0.3 to (y,Aa) and 0.6 to (y,aa). Similarly, an individual with an intermediate value of y is most likely to have genotype Aa and an individual with a large value of y is most likely to have genotype AA. Again prior weights might be specified for these cases.

The basic idea of the iterative EM algorithm described by Jansen (1993) is to replace the single incomplete observation y by its three complete observations (y,AA), (y,Aa) and (y,aa), weighting the three complete observations by specified or updated (conditional) probabilities. Each iteration consists of two steps:

Step 1 (E–step) specify or update weights P(aa | y), P(aA | y) and P(AA | y)
Step 2 (M–step) (a) update the estimate of p by fitting a GLM for multinomial data to the weights;
(b) update estimates of μ_{AA}, μ_{aa} and σ^2 by fitting a weighted GLM for normal data.

The conditional probability P(AA | y) that an individual with observed trait y has genotype AA equals

$$P(AA|y) = \frac{p^2 f(y;\mu_{AA},\sigma^2)}{f(y)}.$$

Similarly,

$$P(Aa|y) = 2pq\frac{f(y;(\mu_{AA}+\mu_{aa})/2,\sigma^2)}{f(y)}; \quad P(aa|y) = q^2\frac{f(y;\mu_{aa},\sigma^2)}{f(y)}.$$

Conditional probabilities are calculated by using the current parameter estimates. The algorithm is conveniently started by setting parameters equal to (well-chosen) initial values.

The log-linear formulations for the genotype frequencies are $\log(p^2) = 2\log(p)$, $\log(2pq) = \log(2)+\log(p)+\log(q)$ and $\log(q^2) = 2\log(q)$. The parameterization for the GLMs for the multinomial part of the model and for the normal part of the model are shown in Table 1. A Genstat procedure, **NORMALMIX**, is given in the appendix. Also, a Genstat program for estimating the parameters of our genetic example is given.

Table 1. Coefficients of regression parameters.

Genotype	Multinomial			Normal	
	$\log(p)$	$\log(q)$	offset	μ_{AA}	μ_{aa}
AA	2	0	0	1	0
Aa	1	1	log(2)	½	½
aa	0	2	o	0	1

3. Discussion

In the algorithm used above, the mixture problem is converted into a complete data problem with additional weighting. A justification of the algorithm is given by Jansen (1993). As a result, the mixture problem could be split into two solvable non-mixture problems. These non-mixture problems can readily be solved by exploiting the options **OFFSET** and **WEIGHTS** of the GLM facilities in Genstat. This makes it possible to transfer all GLM facilities to the corresponding finite mixture equivalent. The distribution of the component counts may be either multinomial or Poisson. The mixing distribution can be for example univariate Normal, exponential, binomial or Poisson. In this paper a simple Genstat procedure for mixtures of Normal distributions is given. However, a general Genstat procedure, which requires specification of a GLM for the mixing proportions and specification of a GLM for the mixing distributions is a straightforward extension of **NORMALMIX**. Jansen (1993) presents a practical example were a generalized linear finite mixture model of ten Weibull distributions is adopted. The computational work was done in Genstat.

References

Jansen R C (1993) Maximum likelihood in a generalized linear finite mixture model by using the EM algorithm. *Biometrics* **49** 227–231.

Appendix: Genstat Program and Procedures

```
JOB         "Genstat program"

"Example of procedure NORMALMIX in the genetic example above; using simulated data"

"simulate data"
VARIATE     [NVALUES=100] TRAIT
VARIATE     [VALUES=2 ,0,-2 ] GENEEFFECT
CALCULATE   TRAIT = NED(URAND(2987;100))
SCALAR      P; 0.3
CALCULATE   Q=1-P
CALCULATE   PAA = P*P
CALCULATE   PAa = 2*P*Q
CALCULATE   URAND = URAND(9183;100)
CALCULATE   GENE = 1 + (URAND.GE.PAA) + (URAND.GE.(PAA+PAa))
CALCULATE   TRAIT = TRAIT + ELEM(GENEEFFECT;GENE)

"triplicate data and construct explanatory variables"
FACTOR      [LABELS=!T(AA,Aa,aa); VALUES=(1,2,3)100] QTL
FACTOR      [LEVELS=100; VALUES=3(1...100)] OBSERVATION
VARIATE     [VALUES=3(#TRAIT)] Y
```

```
CALCULATE XNORMAL[1] = (QTL.EQ.1) + (QTL.EQ.2)/2
CALCULATE XNORMAL[2] = (QTL.EQ.3) + (QTL.EQ.2)/2
CALCULATE XMULTINOMIAL[1] = 2*(QTL.EQ.1) + (QTL.EQ.2)
CALCULATE XMULTINOMIAL[2] = 2*(QTL.EQ.3) + (QTL.EQ.2)
CALCULATE OFFSETMULTINOMIAL = LOG(2)*(QTL.EQ.2)

"initial values for parameters of the normal distribution: μaa, μAA and σ²"
VARIATE    [VALUES=0.5,-0.5, 2] STARTNORMAL
" initial values for P and Q"
VARIATE    [VALUES=0.4,0.6] STARTMULTINOMIAL
CALCULATE STARTMULTINOMIAL = LOG(STARTMULTINOMIAL)
NORMALMIX DATA=Y; OBSERVATION=OBSERVATION;XNORMAL=XNORMAL;\
    XMULTINOMIAL=XMULTINOMIAL;OFFSETMULTINOMIAL=OFFSETMULTINOMIAL;\
    STARTNORMAL=STARTNORMAL; STARTMULTINOMIAL=STARTMULTINOMIAL
STOP


PROCEDURE 'NORMALMIX'
PARAMETER 'DATA','OBSERVATION','XNORMAL','XMULTINOMIAL','STARTNORMAL', \
            'STARTMULTINOMIAL','OFFSETMULTINOMIAL'

UNIT   DATA
CALCULATE N,NN,NM  = NVAL(DATA,XNORMAL,XMULTINOMIAL)

VARIATE    NORMALFIT,MULTFIT
MATRIX     [NN;N] NORMALX
MATRIX     [NM;N] MULTX
VARIATE    [NN] NORMALESTI
VARIATE    [NM] MULTESTI
VARIATE    [VAL=(10000)5] SAVELOGL
SCALAR     VARI

" calculate conditional probabilities from initial parameter values "
EQUATE     XNORMAL; NORMALX
EQUATE     XMULTINOMIAL; MULTX
EQUATE     STARTNORMAL; !P(NORMALESTI,VARI)
EQUATE     STARTMULTINOMIAL; MULTESTI
CALCULATE NORMALFIT  = LTPROD(NORMALX; NORMALESTI)
CALCULATE MULTFIT = EXP(LTPROD(MULTX; MULTESTI))
CALCULATE E  = (DATA - NORMALFIT)/SQRT(VARI)
CALCULATE F  = EXP(E**2/-2)/SQRT(2*ARCCOS(-1)*VARI)
TABULATE   [CL=OBSERVATION; WEIGHT=MULTFIT] F; TOT=FMIX
CALCULATE PCONDITIONAL = MULTFIT*F / ELEM(!(#FMIX);OBSERVATION)

" iterations of the EM algorithm "
FOR    I=1...100
" normal part of the model "
MODEL  [WEIGHTS=PCONDITIONAL] DATA
TERMS  XNORMAL[]
FIT    [PR=*; CONS=OMIT] XNORMAL[]
RKEEP  ESTI=NORMALESTI; FIT=NORMALFIT
CALCULATE VARI = SUM(PCONDITIONAL*(DATA-NORMALFIT)**2)/SUM(PCONDITIONAL)
" multinomial part of the model "
MODEL  [DIST=MULTINOMIAL; LINK=LOG; OFFSET=OFFSETMULTINOMIAL] PCONDITIONAL
TERMS  XMULTINOMIAL[]
FIT    [PR=*; CONS=OMIT] XMULTINOMIAL[]
RKEEP  ESTI=MULTESTI; FIT=MULTFIT
" calculate log-likelihood and update conditional probabilities "
CALCULATE E  = (DATA - NORMALFIT)/SQRT(VARI)
CALCULATE F  = EXP(E**2/-2)/SQRT(2*ARCCOS(-1)*VARI)
TABULATE   [CL=OBSERVATION; WEIGHT=MULTFIT] F; TOT=FMIX
CALCULATE LOGL = SUM(LOG(!(#FMIX)))
PRINT  [IPR=*;SQ=Y] I,LOGL; FIELD=3,10; DECI=0,4
CALCULATE PCONDITIONAL = MULTFIT*F / ELEM(!(#FMIX); OBSERVATION)
CALCULATE SAVELOGL = SHIFT(SAVELOGL ;1)
EQUATE     [NEW=!(1,*)] LOGL; SAVELOGL
EXIT   ABS(SAVELOGL$[1] - SAVELOGL$[5]).LT.0.01
ENDFOR

PRINT  [IPR=*;SQ=Y;SE=Y;OR=AC]\
    !T('ESTIMATES OF REGRESSION PARAMETERS OF NORMAL PART OF MODEL:'),NORMALESTI
PRINT  [IPR=*;SQ=Y;SE=Y] \
    !T('ESTIMATE OF VARIANCE PARAMETER OF NORMAL PART OF MODEL:'),VARI
PRINT  [IPR=*;SQ=Y;SE=Y;OR=AC]\
    !T('ESTIMATES OF REGRESSION PARAMETERS OF MULTINOMIAL PART OF MODEL:'),MULTESTI
ENDPROC
```

The GLM–approach for fitting a growth–curve model

A Keen
Agricultural Mathematics Group
PO Box 100, 6700 AC Wageningen
The Netherlands

1. Introduction

In Genstat Newsletter 29, Ridout (1993) explains a way to fit a growth–curve model to a set of data. His solution is easy enough and induces no problems for a particular dataset. However, the solution is restricted to the Normal distribution for the log transformed response variable. An alternative approach is to consider the model as a generalized linear model (GLM) with a parameter in the link function. A specific feature of the growth–curve model is, that the parameter of the link function is linear, so that in fact all parameters of the model are linear. The class of GLMs is well–known and satisfactory general solutions have been developed to fit its members. Also, GLMs belong to the more general class of generalized linear mixed models (GLMM), useful for situations with more units, possibly allocated to different treatments and affected by covariates. This note explains the GLM–approach, illustrating it with the example of Ridout. Special attention is given to a comparison of the gamma distribution with the lognormal distribution.

2. The GLM-approach using IRLS

The observations of a response variable y with mean μ, are supposed to be the result of the following process:

$$\begin{aligned} y &= \mu + \varepsilon \\ \mathrm{Var}(y) &= \phi V(\mu) \\ \eta &= g(\mu) = X\beta \end{aligned} \qquad (2.1)$$

The variance of y is a function of μ except for the Normal distribution; ϕ is called the dispersion parameter. Function g is called the link function, linking the linear model $X\beta$ to the mean μ of y. X is the design matrix and β the vector of parameters, consisting of elements β_j. A GLM can be fitted by a succession of weighted regressions on a new dependent variate. Both the dependent variate and the weights may involve estimates of μ and, therefore, estimates of the parameters of the regression model. Asterisks will be used to indicate that current values of the parameters are involved. This solution is commonly denoted as iteratively reweighted least squares (IRLS). For a detailed exposition of GLMs, see McCullagh and Nelder (1989). People familiar with GLMs and IRLS may skip the rest of this section. I will present a short review of how IRLS results from maximum likelihood (ML) for a few members of the exponential family, emphasizing the consequences of non–normality of the distribution and consequences of the (nonlinear) link function. The two extensions from the linear regression model with normally distributed response variable are considered first, before combining them.

Non–normal distribution, identity link

Let the linear model be specified at the scale of y:

$$\mu = X\beta$$

and let the log–likelihood be l, with element l_i for observation i. Because of mutual independence of the observations $l=\Sigma_i l_i$. The partial derivative of l_i with respect to parameter β_j in case of Poisson, gamma (with $V(\mu)=\mu^2$), inverse Normal (with $V(\mu)=\mu^3$) and binomial distributions, can be written as

$$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j} = \frac{y_i - \mu_i}{V(\mu_i)} x_{ij} \tag{2.2}$$

where x_{ij} is the derivative of μ_i with respect to β_j; i.e. x_{ij} is the element in row i and column j of the design matrix X in (2.1).

By integrating $$\frac{y_i - \mu_i}{V(\mu_i)}$$

it can easily be verified that this is ML not only for the normal distribution (where $V(\mu)=1$), but also for the binomial, Poisson (when $\phi=1$), gamma and inverse normal distributions. Equating the sum over i in (2.2) to 0 results in the normal equations of the regression with response variate y and weights

$$w^* = \frac{1}{V(\mu^*)} . \tag{2.3}$$

Regressors remain the columns of design matrix X. Iteration is necessary, because in each step μ^* and so the weights w^* are updated. This solution shows the intuitive justification for GLMs: it is just least squares, with heterogeneity of variance corrected for by using the inverse of the variance as weights.

Normal distribution, non–linear link

In case of other than identity link, the derivative of the log likelihood elements l_i with respect to β_j now become

$$\frac{\partial l_i}{\partial \beta_j} = \frac{\partial l_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \eta_i}\frac{\partial \eta_i}{\partial \beta_j} = (y - \mu_i)\frac{1}{g'(\mu_i)} x_{ij} \tag{2.4}$$

Equating the sum over i of (2.4) to 0 can not be solved by regression on y with regressors x_j, because x_{ij} is not the derivative of μ_i with respect to β_j. A solution now is to introduce the variate ζ:

$$\zeta = \eta + (y - \mu) g'(\mu)$$

which is the linear approximation of $g(y)$ in μ. Equation (2.4) becomes

$$\frac{\zeta_i - \eta_i}{[g'(\mu_i)]^2} x_{ij} \tag{2.5}$$

where $g'(.)$ is the derivative of η with respect to μ.

Now equating the sum over i of (2.5) to 0 is equivalent to solving the normal equations of the regression of

$$\zeta^* = \eta^* + (y - \mu^*) g'(\mu^*) \tag{2.6}$$

on x_j, using weights

$$w^* = \frac{1}{[g'(\mu^*)]^2}$$

because x_{ij} is the derivative of η_i with respect to β_j.

So IRLS with this new response variate handles the nonlinearity due to the link function. This solution is commonly used for GLMs. The new response variate ζ^* is often called 'working variate' or 'adjusted dependent variate'. Both names, however, are not very specific and may be used for variates in other situations, having another meaning. A more specific indication is: 'link–adjusted dependent variate', a term I will use in the sequel. Observe, that the link–adjusted dependent variate consists of an estimated mean at the link scale plus a deviation.

Combining distribution and link function

If the distribution is non–normal and a nonlinear link function applies, the solution is just a combination of the separate solutions for non–normal distributions and for nonlinear link functions. The solution then is to use the link–adjusted dependent variate of (2.6) in combination with weights:

$$w^* = \frac{1}{V(\mu^*)\{g'(\mu^*)\}^2} .$$

3. The gamma and the lognormal distribution

The gamma distribution belongs to the GLM system if $V(\mu)=\mu^2$. The link–adjusted dependent variate ζ_g^* is

$$\zeta_g^* = \eta^* + (y-\mu^*)g'(\mu^*) \tag{3.1}$$

and the weights w_g^* are

$$w_g^* = \frac{1}{\{\mu^* g'(\mu^*)\}^2} . \tag{3.2}$$

The lognormal distribution satisfies the same mean–variance relationship $V(\mu)=\mu^2$, but only belongs to the GLM–system if looked at it as the normal distribution for $\ln(y)$. Let

$$E\{\ln(y)\} = \xi \quad \text{and} \quad \text{Var}\{\ln(y)\} = \tau^2.$$

Then

$$\mu = e^{\xi+\tau^2/2} \quad \text{or} \quad \xi = \ln(\mu) - \tau^2/2.$$

For fixed τ, the ML solution follows from IRLS on the link–adjusted dependent variate:

$$\zeta_1^* = \eta^* + (\ln(y) - \xi^*)h'(\xi^*)$$

where $h(\xi) = \eta = g(\mu)$ and $h'(\xi^*)$ is the partial derivative of h with respect to ξ, evaluated in ξ^*. The weights are

$$w^* = \left(\frac{1}{h'(\xi^*)^2}\right) .$$

Note that $h'(\xi) = g'(\mu)\mu$, so the link–adjusted dependent variate and the weights in terms of μ^* are

$$\zeta_1^* = \eta^* + \{\ln(y) - \ln(\mu^*) + \tau^2/2\}\mu^* g'(\mu^*) \tag{3.3}$$

$$w^* = \frac{1}{\{\mu^* g'(\mu^*)\}^2} . \tag{3.4}$$

The term $\tau^2/2$ appears because, although the derivation starts from the lognormal distribution, the mean of y is modeled. This in contrast to Ridout (1993) who models the mean of $\ln(y)$ directly, which implies modelling the median of the lognormal distribution rather than the mean μ. Of course this can be justified just as well. This situation is referred to as 'Normal at log scale'. If $\tau^2/2$ is omitted it is clear that IRLS yields the maximum likelihood solution for the parameters of μ. The same applies if τ^2 is known. An estimate of τ^2 can be obtained from the residual mean square error in each step of the IRLS algorithm with response variate ζ_1^* defined in (3.3) and weights w^* defined in (3.4). This is a degrees of freedom corrected estimate and therefore not a maximum likelihood estimate. Then the estimates of the parameters of μ will not be ML also, but conditional ML–estimates, conditional on the estimate of τ^2.

Comparison of the gamma and the lognormal distribution

Comparison of (3.2) with (3.4) shows that weights are the same function of μ^* for both distributions. The only difference between IRLS for the gamma distribution and IRLS for the lognormal distribution is the link–adjusted dependent variate. Note that

$$\ln(y) - \ln(\mu^*) + \frac{\tau^{*2}}{2} \approx \frac{y-\mu^*}{\mu^*} + \frac{1}{2}\left[\tau^{*2} - \left(\frac{y-\mu^*}{\mu^*}\right)^2\right] \tag{3.5}$$

The expectation of the last term at the right–hand side of (3.5) equals 0 if $\mu^*=\mu$. So when τ^2 is small ζ_l^* and ζ_g^* are approximately equal and the two distributions lead to approximately the same solution.

4. GLM–solutions for the growth curve model with fixed K_1

The growth curve model (Ridout 1993) in the notation of (2.1) is

$$\mu + K_1\ln(\mu) = \mu_0 + K_1\ln(\mu_0) + K_2(t - t_0) \tag{4.1}$$

with μ_0 the value of μ at time $t=t_0$. Note that the right–hand side of the model is linear in t. The intercept is a function of parameters K_1 and μ_0. In the GLM–representation

$$\eta = g(\mu;K_1) = \mu + K_1\ln(\mu). \tag{4.2}$$

So this is a GLM with a link function containing an unknown parameter. Model (4.1) may be written as

$$\eta(K_1) = \eta_0(K_1) + K_2(t - t_0)$$

where $\eta_0(K_1)$ is the value of η (dependent on K_1) at time $t = t_0$. The partial derivatives in the link–adjusted dependent variates for the gamma and lognormal distribution are

$$g'(\mu) = 1 + \frac{K_1}{\mu} \quad \text{and} \quad h'(\xi) = g'(\mu)\mu = \mu + K_1$$

The IRLS–solution now is straightforward for given value K_1^*. The adjusted dependent variate for the gamma distribution equals

$$\zeta_g^* = \eta^* + (y-\mu^*)\left(1 + \frac{K_1^*}{\mu^*}\right) = \eta^* + \frac{y-\mu^*}{\mu^*}(\mu^* + K_1^*)$$

and for the lognormal distribution:

$$\zeta_l^* = \eta^* + \left(\ln(y) - \ln(\mu^*) + \frac{\tau^2}{2}\right)(\mu^* + K_1^*).$$

The weights in both cases are $w^* = (\mu^* + K_1^*)^{-2}$. Each weighted LS–step results in a updated value for η. From the values for η the corresponding values for μ have to be calculated from equation (4.2). A solution is the Gauss–Newton scheme, based on linearization of the right–hand side of (4.2):

$$\eta \approx \eta^* + (\mu-\mu^*)\left(1 + \frac{K_1^*}{\mu^*}\right) \tag{4.3a}$$

which results in the approximation for μ:

$$\mu \approx \mu^* + \frac{\eta - \eta^*}{1 + (K_1^*/\mu^*)} = \mu^* \left(1 + \frac{\eta - \eta^*}{\mu^* + K_1^*} \right) \qquad (4.3b)$$

The calculated value of μ is then taken as the new μ^*, from which the new η^* is calculated. This process may be repeated until convergence, with a suitable convergence criterion based on the difference between μ and μ^* or between η and η^*. However, iteration may not be necessary at all, or may even be disadvantageous, as each step of the IRLS–iteration is a one–step linear approximation, with which a one–step linear approximation for μ may agree best. The results indicate that for fixed K_1 any number of steps for calculating μ from η results in the same convergence behaviour. So one step seems to be sufficient. When discussing the estimation of K_1, this point will be discussed further.

The estimates of K_2 (with standard errors) for the example of Ridout when $K_1 = 1$ are:

	Estimate of K_2 (with standard errors)
Gamma	0.109 (0.00651)
Lognormal	0.110 (0.00702)
Normal at log scale	0.108 (0.00681)

The IRLS requires not more than four iterations until convergence.

5. Estimation of K_1

K_1 can be estimated, using Gauss–Newton optimization, as described by Pregibon (1980). The solution is obtained by a linear approximation of η with respect to K_1:

$$\eta \approx \eta^* + (K_1 - K_1^*)\ln(\mu^*) .$$

Because K_1 is a linear parameter of η, this approximation is perfect for given value of μ. The approximate model for η becomes

$$\eta = \eta_0 + K_2(t - t_0) - (K_1 - K_1^*)\ln(\mu).$$

This means that an extra regressor ($x_{K_1}^* = -\ln(\mu^*)$) has to be added to the model for ζ^*, while ζ^* itself remains unchanged. After each step the updated parameter K_1 is derived as follows:

$$K_1 = K_1^* + \hat{\Delta}_{K_1}$$

with $\hat{\Delta}_{K_1}$ the estimate of the regression coefficient for $x_{K_1}^*$. Each step of the IRLS now results in a updated value for K_1, again called K_1^*, and a updated value for η. From the values for η the corresponding values for μ are derived as explained above, solving equation (4.2), using equation (4.3b). Repeating this updating now seems to obstruct proper convergence. Fitting the model for Ridout's example with starting value 1 for K_1 results in 31 iterations when μ is updated according to equation (4.2) and 8 iterations when μ is updated according to the one–step linear approximation in equation (4.3b). Furthermore, with the one–step updating, convergence is practically independent of the initial value for K_1, while the iterated updating requires a good initial value for K_1 for convergence. This difference in convergence behaviour must be due to the estimation of K_1, as for fixed K_1 it seemed to be no point. An explanation is as follows The regression updates the part $K_1\ln(\mu)$ of η only and not directly $\mu+K_1\ln(\mu)$. So the new η obtained is not a good approximation of the true η in the new value of K_1 that was obtained in the same regression. A better approximation of this η is obtained by the first–order approximation of μ using (4.3b) and calculating from this μ the corresponding new η using (4.2).

The estimate of the standard error of K_1 is obtained from the linear approximation, as the standard error of $\hat{\Delta}_{K_1}$ in the final weighted regression. The resulting estimates (with standard errors) for the example are:

	Estimates (with standard errors)	
	K_1	K_2
Gamma	1.77 (0.527)	0.149 (0.0271)
Lognormal	1.81 (0.512)	0.151 (0.0262)
Normal at log scale	1.79 (0.506)	0.149 (0.0259)

6. Testing $K_1 = K_1^*$

A score test for the adequacy of the value K_1^* for K_1 can be obtained from the linearization according to equation (3.1). With fixed K_1 the test is a *t*–test with 4 degrees of freedom, derived directly by carrying out one extra regression with extra covariate $\ln(\mu^*)$ added to the model after fitting the model with known $K_1 = K_1^*$. The score test is the *t*–test for testing value 0 for the regression coefficient of $\ln(\mu^*)$. An alternative is the likelihood ratio test, comparing the deviances of the restricted model with known value $K_1 = K_1^*$ with the full model without restricting K_1. Assuming chisquare distributions for the deviances the test is a *F*–test, with 1 degree of freedom for the numerator and 4 degrees of freedom for the denominator. This test can also be expressed as a *t*–test in case of two–sided alternative, due to the relationship between the *F* and *t* distribution. Ridout applied the likelihood ratio test. The results of testing $K_1 = 1$ for Ridout's example are:

	Score test		Likelihood ratio test	
	t	*P*	*t*	*P*
Gamma	2.04	0.111	1.93	0.126
Lognormal	2.36	0.079	2.15	0.098
Normal at log scale	2.28	0.085	2.08	0.106

7. Final remarks

Ridout's results were exactly the same as reported above for the Normal distribution, modelling the mean at the log scale, as should be. Note that Ridout's approach can be extended to other distributions using weights as in equation (2.3) within **FITNONLINEAR** and response variate *y*, using the arguments in Section 2 about non–normal distributions. His model must then be formulated in terms of μ and not $\ln(\mu)$.

Note that an alternative IRLS solution for the Normal distribution is obtained using Gauss–Newton's method for estimating the parameters of a nonlinear function when applying least squares. Write

$$\mu \approx \mu^* + \sum_j (\beta_j - \beta_j^*) \left(\frac{\partial \mu}{\partial \beta_j} \right)_{\mu^*} = \mu^* + \sum_j \Delta_j \frac{x_{ij}}{g'(\mu^*)} = \mu^* + \sum_j \Delta_j \chi_j^*$$

which is an approximately linear model in $\beta_j - \beta_j^* = \Delta_j^*$. In the regression with response variate *y* the regressors now are $\chi_j^* = x_{ij}/g'(\mu^*)$, while μ^* is the offset. So not the response variate now is link–adjusted, but the regressors are. Obviously, iteration is necessary to update β_j^*, χ_j^* and μ^*. This approach, however, is in a GLM less adequate than the standard approach with the link–adjusted dependent variate. In the standard approach all nonlinearity is summarized in two places, namely in the link–adjusted dependent variate and in the weights variate. Everything else remains linear. This approach therefore is as close as possible to the linear model. For understanding what is happening and keeping track on the numerical process, only those two variates have to be considered. Also, extensions to more general situations may benefit from the similarity with the linear model.

Usually η_0 will not be a parameter of much interest, at least not more than η at other time points. Possibly μ_0 or μ at other time points are of more interest. Estimates of μ, including μ_0, are derived within the IRLS procedure. Standard errors for these estimates can be obtained from the approximate linear relationship between

μ and η in equation (4.3b), while standard errors for η^* are obtained in the usual way for weighted linear regression. If μ_0 is a parameter of interest, then also an estimate and related standard error can be obtained directly with the IRLS procedure. To see this, write the relationship between μ_0 and η_0 equivalent to equation (4.3a):

$$\eta_0 \approx \eta_0^* + (\mu_0 - \mu_0^*)\left(1 + \frac{K_1}{\mu_0^*}\right)$$

then use the constant η_0^* as offset in the regression, so that the constant in the regression then estimates $(\mu_0 - \mu_0^*)(1 + K_1/\mu_0^*)$. After each step μ_0^* is updated. At convergence the standard error of the constant estimate is the standard error of the estimate of μ_0.

The derivation and example show that the results for the gamma and lognormal distribution agree well. It also hardly matters whether the model is specified for the mean at the original scale or at the log scale. The estimates are approximately the same, whereas standard errors seem to be slightly higher for the gamma distribution. Of course the agreement of the distributions must be close, because otherwise the basis of maximum likelihood as a proper statistical method for practical problems would be very weak as in practice the assumed distribution almost always is just a rough description of the true distribution. Choice of the variance function is much more important than choice of the distribution. This is illustrated by the results for the example obtained from assuming other variance functions $V(\mu)$.

			Estimates (with standard errors)			Tests (t–values)	
$V(\mu)$	Distribution	K_1	K_2	K_1	K_2	Score	LR
1	Normal	1	0.119 (0.0074)	0.288 (0.872)	0.105 (0.0199)	–0.42	–0.53
μ	Quasi-Poisson	1	0.121 (0.0073)	1.128 (0.609)	0.124 (0.0198)	0.31	0.35
μ^2	Gamma	1	0.109 (0.0065)	1.772 (0.527)	0.149 (0.0271)	2.04	1.93
μ^3	Inverse normal	1	0.099 (0.0067)	2.638 (0.746)	0.197 (0.0454)	3.21	3.49

This table shows the importance of the variance function for the conclusions about K_1 and K_2. The choice of variance function may be based on past experience, or on knowledge about the error generating process. If it is required to obtain the choice based on the data, a possible criterion is extended quasi–likelihood (Nelder and Pregibon 1987). Applying the method for the example with K_1 and K_2 free parameters, the result is that the gamma distribution fits best, followed by the lognormal, (quasi–)Poisson, Normal and inverse Normal.

A Genstat program to derive the estimates and tests for the distributions discussed for Ridout's example is given in the appendix.

References

McCullagh P and Nelder J A (1989) *Generalized linear models* (2nd Edn) Chapman and Hall, London.
Nelder J A and Pregibon D (1987) An extended quasi–likelihood function. *Biometrika* **74**(2) 221–232.
Pregibon D (1980) Goodness of link tests for generalized linear models. *Appl. Statst.* **29** 15–24.
Ridout M S (1993) A note on fitting a growth–curve model. *Genstat Newsletter* **29** 6–8, Numerical Algorithm Group, Oxford.

Appendix

```
JOB 'Fitting a growth-curve with GLM'

"================================================================="
" First the model is fitted with K1=1                              "
" One step with extra covariate -ln(Mu0) gives score test for K1.  "
" Extra steps estimate K1.                                         "
"================================================================="
VARI y ; ! (0.09, 0.76, 2.37, 5.12, 6.22, 8.37, 11.5)
VARI t ; !(135, 168, 188, 207, 230, 251, 274)
VARI [NVAL = y] xK1, VMu, Vy
SCAL K1, t0 ; 1, 135

SCAL DifK1, SEK1
SCAL i ; 0 ; DECI = 0
SCAL tK1, CritMu, Crit, Tau2, CondK1, Deviance, Phi, RSS

FOR Distrib = 'Normal', 'Poisson', 'Gamma', 'Lognormal', \
        'Normallog', 'Inverse'
  PAGE
  PRINT Distrib
  CALC Mu0 = y
  &    Eta0 = Mu0 + K1 * LOG (Mu0)
  &    Time = t - t0
  &    i, K1, tK1, CritMu, Crit, Tau2, CondK1 = 0, 1, 1, 1, 1, 0, 1

  IF Distrib .IN. 'Normal'
    EXPR EZetaRes ; !E (ZetaRes = y - Mu0)
    &    EVs ; !E (VMu, Vy = 1)
    &    EDeviance ; !E (Deviance = RSS)

  ELSIF Distrib .IN. 'Poisson'
    EXPR EZetaRes ; !E (ZetaRes = (y - Mu0))
    &    EV ; !E (VMu, Vy = Mu, y)
    &    EDeviance \
         ; !E (Deviance = 2 * SUM (y * LOG (y / Mu0) - (y - Mu0)))

  ELSIF Distrib .IN. 'Gamma'
    EXPR EZetaRes ; !E (ZetaRes = (y - Mu0))
    &    EVs ; !E (VMu, Vy = (Mu, y) ** 2)
    &    EDeviance \
         ; !E (Deviance = 2 * SUM ((y / Mu0) - LOG (y / Mu0) - 1))

  ELSIF Distrib .IN. 'Lognormal'
    EXPR EZetaRes \
         ; !E (ZetaRes = Mu0 * (LOG(y) - LOG (Mu0) + Tau2 / 2))
    &    EVs ; !E (VMu, Vy = (Mu, y) ** 2)
    &    EDeviance ; !E (Deviance = RSS)

  ELSIF Distrib .IN. 'Normallog'
    EXPR EZetaRes ; !E (ZetaRes = Mu0 * (LOG (y) - LOG (Mu0)))
    &    EVs ; !E (VMu, Vy = (Mu, y) ** 2)
    &    EDeviance ; !E (Deviance = RSS)

  ELSIF Distrib .IN. 'Inverse'
    EXPR EZetaRes ; !E (ZetaRes = y - Mu0)
    &    EVs ; !E (VMu, Vy = (Mu, y) ** 3)
    &    EDeviance ; !E (Deviance = \
         2 * SUM (y ** (-1) + y * Mu0 ** (-2) - 2 * Mu0 ** (-1)))
  ENDIF

  FOR [NTIMES = 50]
    CALC i = i + 1
    &    # EZetaRes
    &    # EVs
    &    # EDeviance
    &    Zeta = Eta0 + ZetaRes * (1 + K1 / Mu0)
    &    W = 1 / (VMu * ((1 + K1 / Mu0) ** 2))
    &    xK1 = - LOG (Mu0)

    MODEL [WEIG = W] Zeta ; FITT = Eta ; RESI = yRes
    IF Crit > .000001
      IF CondK1 == 1    : FIT [PRINT = *] Time
      ELSE              : FIT [PRINT = *] xK1 + Time
```

```
      ENDIF
    ELSE
      IF CondK1 == 1
        FIT Time
        FIT xK1 + Time
        CALC StanRes = ABS (yRes) ** (2/3)
        GRAPH StanRes ; Mu
        CALC CondK1 = 2
      ELSE
        FIT xK1 + Time
        CALC StanRes = ABS (yRes) ** (2/3)
        GRAPH StanRes ; Mu
        EXIT
      ENDIF
    ENDIF

    RKEEP ESTI = Pars ; SE = SEPars ; DEVI = RSS ; DF = DF
    CALC Phi = RSS / DF
    CALC Tau2 = Phi

    IF CondK1 == 2
      CALC DifK1, SEK1 = (Pars, SEPars) $ [2]
      &    K1 = K1 + DifK1
      &    tK1 = DifK1 / SEK1
    ENDIF

    FOR [NTIMES = 1]
      CALC Mu = Mu0 * (1 + (Eta - Eta0) / (Mu0 + K1))
      &    Mu = Mu * (Mu > .00001) + .00001 * (Mu <= .00001)
      &    Mu0 = Mu
      &    Eta0 = Mu0 + K1 * LOG (Mu0)
      &    CritMu = MEAN (ABS (Eta - Eta0))
    ENDFOR

    IF CondK1 == 1
      CALC Crit = ABS (CritMu)
    ELSE
      CALC Crit = ABS (tK1) + ABS (CritMu)
    ENDIF

    IF i == 1
      PRINT 'Monitoring information'
      PRINT [SQUA = y] 'Cycle', 'K1', 'tK1', 'CritMu', 'RSS', 'Deviance'
      PRINT [SQUA = y ; IPRINT = *] i, K1, tK1, CritMu, RSS, Deviance
    ELSE
      PRINT [SQUA = y ; IPRINT = *] i, K1, tK1, CritMu, RSS, Deviance
    ENDIF
  ENDFOR
  CALC EQL = - .5 * SUM (LOG (2 * 3.1416 * Phi * Vy) + Deviance / Phi)
  PRINT EQL
ENDFOR

STOP
```

Genstat analysis of Taguchi experiments

Robert E Kempson
ASRU Ltd, University of Kent, UK
Present address:
Glaxo Research and Development Ltd
Park Road, Ware, Herts SG12 0DF, UK

Abstract

Since the Second World War, Japanese industry has become the world leader in many fields of production, especially electronics. One person who has contributed much to this success is Taguchi, who used his experience as an engineer to adapt designed experiments in order to improve process capability and produce a more constant product. This paper reviews relevant techniques in statistical process control and shows how Genstat is used to provide a Taguchi analysis of the raw data and the signal–to–noise ratio appropriate to the required objectives. The particular facilities available in Genstat are shown to be an advantage in subsequent runs of the program where more suitable analyses may be selected in order that an appropriate error term may be formed.

1. Introduction

Designed experiments originated in agricultural research, but in recent years it has become apparent that the methods can be used to great advantage in industry. In Japan there have been dramatic developments in quality improvement in the last thirty years and to a large extent designed experiments have played an important part.

Agricultural experiments in general have few problems with replication since it is often possible to use a little more land or make the plots slightly smaller, use more animals or more plants. Plants are often quite cheap and plentiful, but large animals can be expensive to maintain and so must be used sparingly, and the same applies to patients in many medical experiments. In industry it is often the case that there is a large number of treatments but a small number of experimental units. This means that the experimenter must rely heavily upon his armoury of tools for small experiments, such as fractional replication and Plackett–Burman designs.

2. Orthogonal arrays

Fractional replication owed its origins to Finney around the end of the Second World War and shortly afterwards Plackett and Burman devised other designs useful for small experiments with large numbers of factors. Taguchi incorporated these designs and others into a set of orthogonal arrays which have been extensively tabulated.

Traditionally experimenters have investigated factors one–at–a–time when trying to identify sources of variation, however this is well–known to be an inefficient technique, as evidenced by the following table.

	Levels of factors A,B,C,D used	
Run	A B C D	A B C D
1	1 1 1 1	1 1 1 1
2	2 1 1 1	1 1 2 2
3	1 2 1 1	1 2 1 2
4	1 1 2 1	1 2 2 1
5	1 1 1 2	2 1 2 1
6		2 1 1 2
7		2 2 1 1
8		2 2 2 2

The second column of the table indicates a change in levels of factors A,B,C,D one at a time whereas the second method uses an orthogonal array. The first method has only five runs but each comparison of level 2 against level 1 of a factor is a comparison of only two observations, whereas in the third column the design provides each comparison as the difference in means of four observations, so the latter method is more efficient.

3. Signal–to–Noise Ratios

Taguchi's other important contribution is through the use of the Signal–to–Noise ratio *SN*. He divides the factors of interest in the experiment into two groups:

(a) control factors,
(b) noise factors.

The control factors are those that can be adjusted by the experimenter and may be used deliberately to vary the mean in order to reach a target value. The noise factors are likely to cause variation in the experiment but may not be controllable in practice, for example, factors that may be controlled in the laboratory but not in the factory. The objective is to choose levels of the control factors such that the product is less susceptible to variability over the noise factors.

The control factors are usually tested through an orthogonal array called the inner array, while the noise factors may be replications or they could be examined over another orthogonal array called the outer array. The purpose of the outer array is to evaluate a Signal–to–Noise ratio *SN*, which is a quantity analogous to the inverse of the coefficient of variation (*CV*). A high value of *SN* is desirable as a measure of good experimental control, which corresponds to a low *CV*. The value of *SN* depends upon the objective of the study, which is not always the same. Sometimes it is desirable that the quantity measured is close to its target value, such as when components must fit together, but in other cases it is required for the variable to be as large as possible, as in lifetimes of electronic components, but small values may also be important, such as the level of wear.

Suitable values for *SN* are as follows:

Nominal is best (NIB): $$SN = 10 \log\left(\frac{\bar{y}^2}{s} - \frac{1}{n}\right)$$

Smallest is best (SIB): $$SN = -10 \log\left(\frac{1}{n}\sum y^2\right)$$

Largest is best (LIB): $$SN = -10 \log\left(\frac{1}{n}\sum \frac{1}{y^2}\right)$$

Logvariance (LS2): $$SN = \log(s^2)$$

Some authors argue that the logvariance should always be used in preference to the other forms of the *SN*.

4. Example

ASI describes an example on the manufacture of tiles in the 1950s first reported by Taguchi (1987). A manufacturer of tiles purchased a tunnel kiln and produced tiles by baking them in the kiln as they rested on a truck which passed slowly along a track through the kiln. Although the tiles at the centre of the stack attained the specification standards the tiles on the outside were very variable and over half of them fell outside the specification limits. The target value for the tile measurements was 150 mm. The standard engineering approach

is to try to identify the source of the variation and remove it, but the Taguchi approach is to choose factor levels so that they are not susceptible to noise. The procedure is to maximise the *SN* values through a suitable choice of factor levels then adjust the nominal value by altering the factors which affect the mean but not the *SN*.

Code	Factor	L1	L2
A	Limestone content	5%	1%
B	Limestone texture	Coarse	Fine
C	Agalmatolite content	43%	53%
D	Agalmatolite type	Old	New
E	Charging quantity	1300 kg	1200 kg
F	Proportion of reuse	0	4%
G	Feldspar content	0	5%

These factors can all be controlled because they are affected by the *composition* of the material used in the production of the tiles. The variability of the response was clearly affected by the *positions* of the tiles within the kiln, but it was too expensive to control the variability between positions so position had to be treated as a noise factor. The positions within the stack were as follows:

Block 1	Middle
Block 2	Bottom
Block 3	Side
Block 4	Top
Block 5	Top corner

The design and sizes of the tiles in mm were as given in the following table.

	Factor							Block				
Run	A	B	C	D	E	F	G	1	2	3	4	5
1	1	1	1	1	1	1	1	151.9	151.4	150.4	150.2	149.6
2	1	1	1	2	2	2	2	151.5	150.8	150.0	149.4	149.1
3	1	2	2	1	1	2	2	153.1	151.8	151.8	151.4	150.6
4	1	2	2	2	2	1	1	152.2	151.3	151.1	150.6	150.0
5	2	1	2	1	2	1	2	151.5	150.8	150.6	150.2	149.7
6	2	1	2	2	1	2	1	156.5	152.1	150.3	148.5	144.6
7	2	2	1	1	2	2	1	154.5	153.3	151.8	150.4	149.6
8	2	2	1	2	1	1	2	153.0	152.0	151.3	150.0	149.5

Analysis of the data as a randomised blocks design gives the standard analysis:

Source	DF	S.S.	M.S.	F
Blocks	4	72.506	18.126	(11.40)
A	1	0.100	0.100	0.06
B	1	10.201	10.201	6.42
C	1	0.025	0.025	0.02
D	1	2.916	2.916	1.83
E	1	0.064	0.064	0.04
F	1	0.361	0.361	0.23
F	1	0.121	0.121	0.08
Residual	28	44.522	1.590	
Total	39	130.816		

It is apparent that factor B accounts for substantially more of the variation than could be expected by chance, so clearly the analysis suggests that the manufacturer is best able to control the size of tiles by adjusting the texture of the limestone content.

Now the variance ratio for Blocks is over 11, so it is clear that there is substantial variation in tile size within the stack. The standard engineering approach is to identify sources of variation and control them, which means that fans would be installed within the kiln to circulate the heat evenly. This was an expensive option and especially annoying as the kiln was new when the problem was noticed.

Taguchi's solution is to make the tiles from a clay which is less susceptible to change over the stack. The NIB criterion is appropriate here since tiles should be neither undersize nor oversize to fit well.

Run	$\bar{y}$	s	*SN*
1	150.70	0.933	44.17
2	150.16	0.991	43.61
3	151.74	0.904	44.49
4	151.04	0.820	45.30
5	150.56	0.673	46.99
6	150.40	4.398	30.68
7	151.92	2.017	37.54
8	151.16	1.433	40.46

Run 6 gives a very low *SN*, and runs 7 and 8 are also poor. The *SN*–values may be calculated for each level of the factors.

Factor	level 1	level 2	*SN* diff	Preference
A	44.39	38.92	–5.47	A1
B	41.36	41.95	0.59	B2
C	41.44	41.87	0.43	C2
D	43.30	40.01	–3.29	D1
E	39.95	43.36	3.41	E2
F	44.23	39.08	–5.15	F1
G	39.42	43.89	4.47	G2

Some of the preferences were strong in the cases where large differences in the *SN* were found. As well as the *SN*–values, the practical and cost considerations were taken into account in the decision over the final composition of the clay, which was a choice between A1, B1 or B2, C1, D2, E2, F1, G2. When the improved tiles were manufactured the results were very close to target.

5. Genstat analysis

A Genstat program was written to perform the analysis described above, and several interesting points emerged from the construction of the program. It was necessary to allow the user the choice of the different *SN* measures, depending on the objective of the study. It was necessary to allow for a variable number of factors in order to make the program general, and the statement

```
VARIATE [VALUES=1...#nf] vch1
```

permits the construction of a variate choice for `nf` factors. The default setting of `vch1` is `1,2,3,4...nf` which represents the complete factor set. This is a sensible choice for the first run of the program but can be overridden by the user on the next run when he has decided which variates to keep as sources of variation. Since industrial experiments are often small, the user may have to declare the error term as the sum of several effects, and this option needs to be available.

The following code permits the input of the `nf` sets of factor levels prior to analysis.

```
SCALAR l[1...#nf]
READ [CHANNEL=2] l[1...#nf]
UNIT [NVALUES=#nd]
VARIATE [NVALUES=#nt] fl[1...#nf]
READ [CHANNEL=2] fl[1...#nf]
FOR ld=l[1...#nf] ; fld=fl[1...#nf] ; fd=f[1...#nf]
 FACTOR [LEVEL=ld ] fd ; VALUES=!(#nn(#fld))
ENDFOR
READ [CHANNEL=2] y
TREATMENT f[#vch1]
ANOVA y
```

This section of code exemplifies the use of the `#` operator as a prefix in order that variables may take algebraic values in arrays and lists. It was necessary to insert algebraic values to make the program general.

The parameters `f[#vch1]` and `f[#vch2]` in the `TREATMENTS` statement allow the user to make a selection from the full factor set to perform a subset analysis on the second run. This facility is a useful labour–saving device.

The scalar `nsd` was used to adjust the number of significant working digits to allow correspondence in the analysis of published calculations. (There are many examples in the literature where intermediate calculations are prematurely rounded so subsequent analysis is only approximate.)

6. Discussion

The appended program was found to be a convenient tool for the examination of analyses of Taguchi designs. In particular it was most helpful to run the program a second time using subsets from the full variable sets, and Genstat's subsetting facilities are particularly convenient. However one of the most contentious aspects of small experiments is the somewhat arbitrary decision as to which effects should be treated as error terms. Daniel (1959) proposed a normal probability plot of residuals against order statistics provides a useful rule to decide which effects should be absorbed. A Daniel plot would have been a useful addition to the program but it would still be up to the user to decide which effects to absorb. The program performs an analysis of variance of the *SN*–values, which is not usual but the `ANOVA` statement provides tables of means, which are required. It is normal practice to supply graphs of the *SN* for each variable, but these are easily obtained from the output.

Further reading is available in the hefty tomes of Taguchi (1987) and a encyclopedia of orthogonal arrays is available in Taguchi and Konoshi (1987). More recent and easily readable accounts of the techniques are to be found in Logothetis and Wynn (1989) and Bendell *et al* (1989).

References

Bendell A, Disney J and Pridmore W A (1989) *Taguchi methods: applications in world industry*, IFS Publications.

Daniel C (1959) Use of half–normal plots in interpreting factorial two–level experiments *Technometrics* **1** 311–341.

Logothetis N and Wynn H P (1989) *Quality through design* Clarendon, Oxford

Taguchi G (1987) *System of experimental design* (2 volumes), ASI, Michigan USA.

Taguchi G and Konoshi S (1987) *Orthogonal arrays and linear graphs* ASI, Michigan USA.

Appendix: Program listing

```
"
"Taguchi analysis program:

Input scalars : numbers of treatments, noise factors, control factors, signal/noise
type (1: nominal, 2: smallest, 3: largest - is best, 4: logvar), number of significant
working decimals, indicator of variable subsetting for mean and for SN (zero for all
factors)

Input choice of factors for mean (if indicator non-zero),
Input choice of factors for SN (if indicator non-zero).

Input number of levels of each factor;
Input factor levels;
Input data by treatments
"

SCALAR nt,nn,nf,snt,nsd,ch1,ch2

OPEN 'tag.dat' ; CHANNEL=2 ; FILETYPE=input
READ [CHANNEL=2 ; END=*] nt,nn,nf,snt,nsd,ch1,ch2
CALCULATE  nd=nt*nn

" Variable selection for Anova table for (a) mean, (b) SN "
IF ch1 .NE. 0
  VARIATE [NVALUES=#ch1] vch1
  READ [CHANNEL=2] vch1
ELSE
  VARIATE [VALUES=1...#nf] vch1
ENDIF
IF ch2 .NE. 0
  VARIATE [NVALUES=#ch2] vch2
  READ [CHANNEL=2] vch2
ELSE
  VARIATE [VALUES=1...#nf] vch2
ENDIF

" Read in factor levels for control factors "
SCALAR l[1...#nf]
READ [CHANNEL=2] l[1...#nf]
UNIT [NVALUES=#nd]
VARIATE [NVALUES=#nt] fl[1...#nf]
READ [CHANNEL=2] fl[1...#nf]
FOR ld=l[1...#nf] ; fld=fl[1...#nf] ; fd=f[1...#nf]
  FACTOR [LEVEL=ld ] fd ; VALUES=!(#nn(#fld))
ENDFOR

" Analyse mean "
READ [CHANNEL=2] y
TREATMENT f[#vch1]
ANOVA y

" Calculate SN values "
UNIT [NVALUES=#nt]
VARIATE [n=#nn] v[1...#nt]
EQUATE OLDSTRUCTURE=y ; NEWSTRUCTURE=!P(v[1...#nt])
SCALAR sn[1...#nt]
FOR vd=v[1...#nt] ; snd=sn[1...#nt]
  IF snt==1
```

```
    CALCULATE a1=MEAN(vd)
    CALCULATE a2=VAR(vd)
    CALCULATE snd=10*LOG10(a1*a1/a2-1/nn)
  ELSIF snt==2
    CALCULATE a3=vd*vd
    CALCULATE a4=SUM(a3)/nn
    CALCULATE snd=-10*LOG10(a4)
  ELSIF snt==3
    CALCULATE a5=1/vd/vd
    CALCULATE a6=SUM(a5)/nn
    CALCULATE snd=-10*LOG10(a6)
  ELSIF snt==4
    CALCULATE a2=VAR(vd)
    CALCULATE snd=-10*LOG(a2)
  ELSE
    PRINT 'Signal-to-noise type out of range'
  ENDIF
ENDFOR
VARIATE [nvalues=#nt] sny
EQUATE OLDSTRUCTURE=sn ; NEWSTRUCTURE=sny

" Adjustment for significant decimals "
SCALAR p
CALCULATE p=10**nsd
CALCULATE sny=INT(sny*p+0.5)/p

" Printout signal-to-noise values "
PRINT fl[1...#nf],sny ; FIELD=(#nf(6),10) ; DEC=(#nf(0),2)

" Analyse SN values "
FOR ld=l[1...#nf] ; fld=fl[1...#nf] ; fd=f[1...#nf]
  FACTOR [LEVEL=ld ] fd ; VALUES=!(#fld)
ENDFOR
TREATMENT f[#vch2]
ANOVA sny
```

A Genstat procedure to assess the performance of models with independent data

A J Rook
AFRC Institute of Grassland and Environmental Research
North Wyke, OKEHAMPTON, Devon EX20 2SB, UK

and

M S Dhanoa
AFRC Institute of Grassland and Environmental Research
Plas Gogerddan, ABERYSTWYTH, Dyfed SY23 3EB, UK

The mean square error of prediction (MSPE) provides a concise quantitative summary of the predictive ability of a model. It can also be decomposed in a manner which provides an insight into model inadequacies and thus aids in the model building process.

With some models, for example the standard linear model, it is possible to calculate the MSPE from a knowledge of the distribution of the independent variables without the need to calculate the predicted values. However, this cannot be done with complex mechanistic models or with 'rules of thumb'. In such situations the MSPE must be calculated directly from a set of actual values independent of those used to estimate the parameters of the model and the corresponding predicted values. This approach is often used during model building by dividing the available data in two, using half for estimation and half for validation.

A *post–hoc* evaluation should always begin with an examination of a scatter graph of actual versus predicted values. This allows a qualitative assessment of the adequacy of the model by comparing the points so obtained with the 1 : 1 line of perfect prediction. An example of such a graph is given in Figure 1. The MSPE supplements this with a quantitative summary which is particularly useful when comparing several models.

MSPE is calculated from the actual and predicted values as

$$\text{MSPE} = \frac{1}{n}\sum_{i=1}^{n}(P_i - A_i)^2$$

Theil (1966) suggested two readily interpretable decompositions of the MSPE,

$$\text{MSPE} = (\bar{P}-\bar{A})^2 + (s_P - s_A)^2 + 2(1-r)s_P s_A$$

or

$$\text{MSPE} = (\bar{P}-\bar{A})^2 + (s_P - rs_A)^2 + (1-r^2)s_A^2$$

where $\bar{P} = \frac{1}{n}\sum P_i$; $s_P^2 = \frac{1}{n}\sum (P_i-\bar{P})^2$; $\bar{A}$ and s_A^2 are defined similarly; $rs_A s_P = \frac{1}{n}\sum (P_i-\bar{P})(A_i-\bar{A})$.

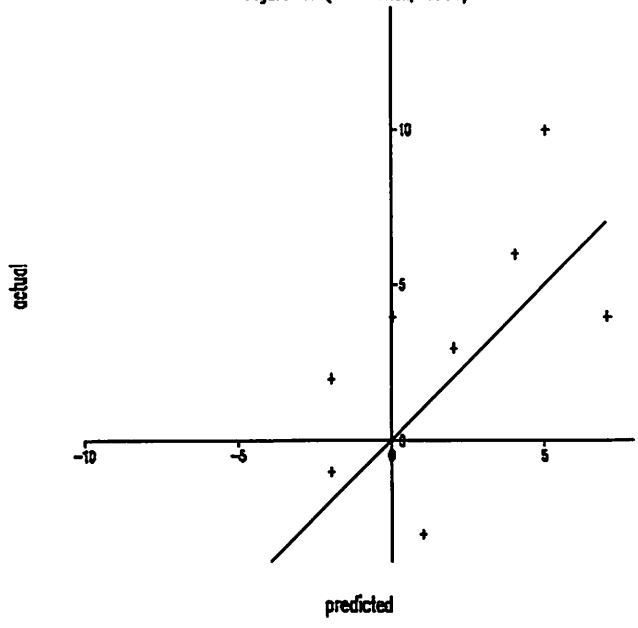
The first term in both decompositions may be interpreted as error due to central tendency (bias) and is 0 only when the mean predicted value equals the mean actual value. In the first decomposition, the second term is 0 only when the predicted variance equals the observed variance, and may be interpreted as error due to unequal variation, while the third term is zero only when there is a perfect positive correlation between actual and predicted values and may be interpreted as error due to incomplete covariation. The first term in the second decomposition is 0 only when the slope of the least squares regression of actual on predicted values is 1 and may be interpreted as error due to regression whereas the final term in this decomposition is interpreted as error due to disturbance, that is 'unexplained variance' which cannot be eliminated by linear correction of the predictions. These decompositions are not sensitive to nonlinear departures from perfect prediction and some caution is therefore need in their interpretation.

A procedure called **MSPE** has been written which calculates the MSPE and its decompositions and also prints the component terms in the formulae. It has two parameters, **ACTUAL=** and **PREDICTED=** which are set to variates

containing the actual and predicted values respectively. The procedure ignores units with missing values in either variate. The variates may be restricted but both must be restricted in the same way.

An example output is shown below. The data are those used in Theil's (1966) original example and is also shown in Figure 1. The mean actual and predicted values are printed together with the MSPE and decompositions. These are also expressed as percentages of MSPE to allow comparison of different models. The 'mean prediction error' is √MSPE and indicates how large on average the prediction error will be. It is also expressed as a percentage of the actual mean. The components of the formulae are also printed, namely: bias, slope of actual on predicted values, correlation of actual and predicted values and the standard deviations of actual, predicted and their differences.

Figure 1. (after Theil, 1966)




```
*** MSPE ANALYSIS ***

  Means of actual and predicted values
   Actual        Predicted
   1.400         0.9000

  Components of MSPE
   MSPE          Bias     Regression  Disturbance     Variance    Covariance
   10.10         0.250        0.1151        9.735        2.168         7.682

  Percentage Contribution of components of MSPE
   MSPE          Bias     Regression  Disturbance     Variance    Covariance
   100.0         2.475         1.140        96.39        21.47         76.06

  Mean Prediction Error
   Absolute   % of mean        Bias   A on P Slope   corr(R)   R_sq
    3.178       227.0     -0.5000          1.100    0.7669   0.5882

  Standard Deviations
   differences    actual    predicted
    3.308         5.125      3.573
```

References

Bibby J and Toutenberg H (1977) *Prediction and Improved Estimation in Linear Models,* Wiley, London.
Theil H (1966) *Applied Economic Forecasting,* North Holland Publishing Company, Amsterdam.

Appendix: Genstat Procedure

```
PROCEDURE 'MSPE'
"
A. J. Rook
AFRC Institute of Grassland and Environmental Research,
North Wyke Okehampton, Devon EX20 2SB

M. S. Dhanoa
AFRC Institute of Grassland and Environmental Research,
Plas Gogerddan, Aberystwyth, Dyfed, SY23 3EB

Version 5.0 3/12/92

Procedure to calculate mean squared prediction error and its components from actual
and predicted values of a response variate.
Printing will be to default output file set up prior to call by user.

References:
Bibby, J. and Toutenberg, H. (1977). Prediction and Improved Estimation in Linear
  Models. London, Wiley.
Theil, H. (1966). Applied Economic Forecasting, North Holland Publishing Company.
"

PARAMETER NAME= \
  'ACTUAL', "(I: variate) the actual values of the response variate" \
  'PREDICTED'; "(I: variate) the predicted values of the response variate" \
  SET=yes,yes; DECLARED=yes,yes; TYPE=!T(variate),!T(variate); \
  PRESENT=yes,yes; COMPATIBLE=!T(type,nvalues)

"
Harmonise missing values for ACTUAL and PREDICTED. Done in this way to rather than
with restrict so that restricted variates can be input. Adjusted variates are stored
in new variates (lower case) to avoid carrying this setting outside the procedure.
"
SCALAR miss
CALCULATE miss = CONSTANTS('*')
  & mva,mvp = ACTUAL,PREDICTED.EQ.miss
  & act,pred = MVINSERT(ACTUAL,PREDICTED;mva,mvp)
"
Calculate means of actual and predicted values
"
CALCULATE mact,mpred = MEAN(act,pred)
"
Calculate difference and squared difference of predicted and actual values.
"
CALCULATE dif = pred-act
  & dif2 = dif*dif
"
Calculate bias and mspe.
"
CALCULATE bias = MEAN(dif)
  & mspe = MEAN(dif2)
"
Calculate root mean square (mean prediction error) and square of bias and express rms
as percentage of mean actual value.
"
CALCULATE rms = SQRT(mspe)
  & bias2 = bias*bias
  & %rms = (rms*100)/mact
"
Calculate (uncorrected) standard deviations of actual and predicted values, and their
differences.
"
CALCULATE sdact,sdpred,sddif = SQRT( VAR(act,pred,dif) )
```

```
"
"Calculate correlation, R-squared and regression of actual on predicted values.
"
MODEL act
FIT [PRINT=*] pred
RKEEP ESTIMATE=coef; FITTED=fitted
CALCULATE  slope=coef$[2]
  &  corr = slope*sdpred/sdact
  &  corr_sq = corr*corr
"
"Calculate corrected standard deviations of actual and predicted values.
"
CALCULATE n1 = NOBS(dif)
  & correct = SQRT((n1-1)/n1)
  & csdact = sdact*correct
  & csdpred = sdpred*correct
"
"Calculate components of mspe due to regression and disturbances.
"
CALCULATE line = (csdpred-corr*csdact)**2
  & random = (1-corr_sq)*csdact*csdact
"
"Calculate components of mspe due to unequal variation and incomplete covariation
"
CALCULATE uv = (csdpred-csdact)**2
  & inco = 2*(1-corr)*csdpred*csdact
"
"Calculate percentage contribution of components of mspe.
"
CALCULATE %bias2,%line,%random,%uv,%inco,%mspe = \
           bias2,line,random,uv,inco,mspe*100/mspe
"
"Print overall heading.
"
PAGE
PRINT [SQUASH=yes] '*** MSPE ANALYSIS ***'
"
"Print breakdown of mspe.
"
PRINT   '       Means of actual and predicted values'
  & [SQUASH=yes] '       Actual        Predicted'
  & [IPRINT=*; SQUASH=yes] mact,mpred
  & [SQUASH=no] '       Components of MSPE'
  & [SQUASH=yes] '        MSPE        Bias     Regression',\
                 ' Disturbance     Variance    Covariance'
  & [SQUASH=yes] mspe,bias2,line,random,uv,inco
  & [SQUASH=no] '       Percentage Contribution of components of MSPE'
  & [SQUASH=yes] '        MSPE        Bias     Regression',\
                 ' Disturbance     Variance    Covariance'
  & [SQUASH=yes] %mspe,%bias2,%line,%random,%uv,%inco
  & [SQUASH=no] '    Mean Prediction Error                                    ',\
                '                                  Standard Deviations'
  & [SQUASH=yes] '    Absolute    % of mean       Bias    A on P Slope  ',\
                 'corr(R)       R_sq      differences   actual    predicted'
  & [SQUASH=yes] rms,%rms,bias,slope,corr,corr_sq,sddif,sdact,sdpred
ENDPROCEDURE
RETURN
```

Double and triple Youden rectangles and Genstat ANOVA

D A Preece
Institute of Mathematics and Statistics
Cornwallis Building
The University
Canterbury, Kent CT2 7NF, UK

As defined by Bailey (1989), a 'double Youden rectangle' is a $k \times v$ row–and–column design ($k < v$) for two non–interacting sets of treatments, such that

(i) each of the v treatments from the first set appears exactly once in each row and no more than once in each column;

(ii) each of the k treatments from the second set appears exactly once in each column and either n or $n + 1$ times in each row, where n is the integral part of v/k;

(iii) each treatment from each set occurs with each treatment from the other set exactly once;

(iv) any two treatments from the first set occur together in the same number of columns, i.e. the column-subsets of first–set treatments constitute the blocks of a symmetric balanced incomplete block design; and

(v) if n occurrences of each second-set treatment are removed from each row, leaving $m = v - nk$ second–set treatments in each row, then either $m = 1$ or the subsets of second-set treatments remaining within the rows constitute the blocks of another symmetric balanced incomplete block design.

A 4×7 example is the following, where upper–case letters are used for treatments of the first set, and lower–case letters for the second:

Cd Ba Gb Ad Eb Fc Dc
Bc Cb Da Ec Aa Gd Fd
Fb Ac Ed Db Bd Ca Ga
Ea Dd Cc Fa Gc Ab Bb

In any example of this size, any two treatments from the first set occur together in 2 columns (e.g., C and F occur together in columns 1 and 6), and any treatment from the second set occurs once or twice in each row.

In a double Youden rectangle, the treatments of the first set are disposed in a 'Youden square' and are, in a standard statistical sense, 'balanced' with respect to columns, and the treatments of the second set are balanced with respect to rows. Otherwise the factors of the design are orthogonal to one another. As the two relationships of balance are independent of one another, the design is 'balanced overall' in a sense that implies that the design can be analysed straightforwardly by Genstat's ANOVA. Such an analysis is as follows:

```
 1  UNITS [28]
 2  FACTOR [LEVELS=4; VALUES=7(1...4)] Row
 3  FACTOR [LEVELS=7; VALUES=(1...7)4] Column
 4  FACTOR [LEVELS=7; LABELS=!T(A,B,C,D,E,F,G)] T1
 5  FACTOR [LEVELS=4; LABELS=!T(a,b,c,d)] T2
 6  READ [PRINT=data; LAYOUT=fixed; \
 7        FORMAT=!(((1,1,-1)7,*)4)] T1,T2;
          FREPRESENTATION=labels,labels
 8  Cd Ba Gb Ad Eb Fc Dc
 9  Bc Cb Da Ec Aa Gd Fd
10  Fb Ac Ed Db Bd Ca Ga
11  Ea Dd Cc Fa Gc Ab Bb
12  :
13  BLOCKSTRUCTURE Row * Column
14  TREATMENTSTRUCTURE T1 + T2
```

```
  15  ANOVA

15.....................................................

***** Analysis of variance *****

Source of variation           d.f.

Row stratum
T2                               3

Column stratum
T1                               6

Row.Column stratum
T1                               6
T2                               3
Residual                         9

Total                           27


***** Information summary *****

Model term                     e.f.  non-orthogonal terms

Row stratum
  T2                          0.020

Column stratum
  T1                          0.125

Row.Column stratum
  T1                          0.875  Column
  T2                          0.980  Row
```

The efficiency factors of 0.875 and 0.980 indicate that the bottom stratum holds 0.875 of the information on treatments from the first set and 0.980 of the information on treatments from the second set.

A double Youden rectangle can be obtained for any size $k \times v$ where $k = v - 1 > 3$ (see Preece 1994). However, knowledge of double Youden rectangles of other sizes is sparse, as is indicated by Table 1, which covers sizes for which Youden squares exist.

Preece (1994b) has shown that, in certain very special cases, a further set of k treatments can be added to a double Youden rectangle to form what he has called a 'triple Youden rectangle'. This is a row–and–column design for three non–interacting sets of treatments, and has the properties that

(a) if either set of k treatments is omitted, the design becomes a double Youden rectangle;

(b) each of the two sets of k treatments is balanced with respect to the other in the same way that each of these two sets is balanced with respect to rows;

(c) the design has overall balance for either of the two sets of k treatments.

We here omit formal details of how condition (c) is met in general and of how condition (b) is formulated precisely. Instead we give the following 4×13 triple Youden rectangle, where the 13 treatments of the first set are denoted (A, 2, 3, ... , 9, T, J, Q, K) in conformity with the denominations of standard English playing cards, and the 4 treatments of each of the two remaining sets are denoted (s, d, c, h) for (spades, diamonds, clubs, hearts):

Ass Jch Qhd Kdc 5cs 6hs 7ds 2hc 3dh 4cd 8sc 9sh Tsd
2dd Add 6sh Tcs Qdc Jdc 4sc 9cd 8cd Kss 7hh 3hs 5hh
3cc 8hs Acc 7sd 2sh Kch Qch Jss Thc 9hc 6dd 5dd 4ds
4hh 5sc 9ds Ahh Khd 3sd Jhd Tdh Qss 8dh 2cs 7cc 6cc

Table 1: Present knowledge of double Youden rectangles (DYRs) with $k < v - 1$ and $k < 13$

$k \times v$	Whether DYRs exist	References
3 × 7	No	Preece (1966)
4 × 13	Yes	Preece (1982)
4 × 7	Yes	Clarke (1967), Preece (1991)
5 × 21	?	
5 × 11	Yes	Preece (1994a)
6 × 31	?	
6 × 16	No	
6 × 11	Yes	Preece (1991, 1994a)
7 × 43	No	
7 × 22	No	
7 × 15	Yes	Preece (1971), Vowden (1994)
8 × 57	?	
8 × 29	No	
8 × 15	Yes	Preece (1993)
9 × 73	?	
9 × 37	?	
9 × 25	No	
9 × 19	Yes	Vowden (personal communication))
9 × 13	No	
10 × 91	?	
10 × 46	No	
10 × 31	?	
10 × 19	?	
10 × 16	No	
11 × 111	No	
11 × 56	?	
11 × 23	Yes	Preece (1971), Vowden (1994)
12 × 133	?	
12 × 67	No	
12 × 45	No	
12 × 34	No	
12 × 23	?	

This design can be represented by mounting each of the 52 small cards of a pack of patience cards on top of one of the 52 larger cards from a whist pack; for example, if the suits of the whist and patience cards are used for the treatments of, respectively, the second and third sets, then the last entry in the first row of the design is represented by mounting the 10 of diamonds from the patience pack on the 10 of spades from the whist pack.

The following output from Genstat's **ANOVA** confirms that the design does indeed have the overall balance that has been claimed. The identifier **Value** is used for the factor with levels (A, 2, ... , 9, T, J, Q, K), and the identifiers **Suit1** and **Suit2** for the second and third sets of treatments.

```
1  UNITS [52]
2  FACTOR [LEVELS=4; VALUES=13'(1...4)] Row ,
```

```
  3  FACTOR [LEVELS=13; VALUES=(1...13)4] Column
  4  FACTOR [LEVELS=13; \
  5  LABELS=!T('A','2','3','4','5','6','7','8','9','T',
           'J','Q','K')] Value
  6  FACTOR [LEVELS=4; LABELS=!T(s,d,c,h)] Suit1
  7  FACTOR [LEVELS=4; LABELS=!T(s,d,c,h)] Suit2
  8  READ [PRINT=data; LAYOUT=fixed; \
  9        FORMAT=!(((3(1),-1)13,*)4)] Value,Suit1,Suit2; \
 10                                    FREPRESENTATION=labels,
                                       labels,labels
 11  Ass Jch Qhd Kdc 5cs 6hs 7ds 2hc 3dh 4cd 8sc 9sh Tsd
 12  2dd Add 6sh Tcs Qdc Jdc 4sc 9cd 8cd Kss 7hh 3hs 5hh
 13  3cc 8hs Acc 7sd 2sh Kch Qch Jss Thc 9hc 6dd 5dd 4ds
 14  4hh 5sc 9ds Ahh Khd 3sd Jhd Tdh Qss 8dh 2cs 7cc 6cc
 15  :
 16  BLOCKSTRUCTURE Row * Column
 17  TREATMENTSTRUCTURE Value + Suit1 + Suit2
 18  ANOVA

* MESSAGE: non-orthogonality between treatment terms.  The effects (printed or used
  to calculate means), the efficiency factor and the sum of squares for each treatment
  term are for that term eliminating previous terms in the TREATMENT formula and
  ignoring subsequent terms.

18.....................................................

***** Analysis of variance *****
Source of variation          d.f.

Row stratum
Suit1                          3

Column stratum
Value                         12

Row.Column stratum
Value                         12
Suit1                          3
Suit2                          3
Residual                      18

Total                         51


***** Information summary *****

Model term                    e.f.  non-orthogonal terms

Row stratum
  Suit1                      0.006

Column stratum
  Value                      0.187

Row.Column stratum
  Value                      0.813  Column
  Suit1                      0.994  Row
  Suit2                      0.989  Row  Suit1
```

In interpreting this last output, we must take careful heed of the **MESSAGE**, which reminds us of the sequential fitting of the model terms.

Thus the bottom–stratum efficiency factor of 0.989 for **Suit2** takes account of the partial confounding of **Suit2** with **Suit1** in the bottom stratum. Similarly, a bottom-stratum efficiency factor of 0.989 (not 0.994) could have been obtained for **Suit1** if **Suit1** had been fitted after **Suit2**; the printed bottom-stratum efficiency factor of 0.994 for **Suit1** is calculated ignoring **Suit2**. In the **Row** stratum, the printed output correctly reports an efficiency factor of 0.006 for **Suit1** ignoring **Suit2**, but unfortunately does not go on to report that, in the **Row** stratum, the **Suit1** information is aliased with the corresponding **Suit2** information.

References

Bailey R A (1989) Designs: mappings between structured sets. In: *Surveys in Combinatorics* (ed. J Siemons), 22–51, Cambridge University Press.

Clarke G M (1967) Four–way balanced designs based on Youden squares with 5, 6 or 7 treatments. *Biometrics* **23** 803–812.

Preece D A (1966) Some row and column designs for two sets of treatments. *Biometrics* **22** 1–25.

Preece D A (1971) Some new balanced row–and–column designs for two non–interacting sets of treatments. *Biometrics* **27** 426–430.

Preece D A (1982) Some partly cyclic 13 × 4 Youden 'squares' and a balanced arrangement for a pack of cards. *Utilitas Mathematica* **22** 255–263.

Preece D A (1991) Double Youden rectangles of size 6 × 11. *Mathematical Scientist* **16** 41–45.

Preece D A (1993) A set of double Youden rectangles of size 8 × 15. *Ars Combinatoria* (to appear).

Preece D A (1994a) Double Youden rectangles - an update with examples of size 5 × 11. *Discrete Mathematics* (to appear).

Preece D A (1994b) Triple Youden rectangles - a new class of fully balanced combinatorial arrangements. *Ars Combinatoria* (to appear).

Vowden B J (1994) Infinite series of double Youden rectangles. *Discrete Mathematics* (to appear).

Genstat mode for Gnu Emacs

R D Ball
DSIR Applied Mathematics
Private Bag 92169 Auckland
New Zealand
Email: rod@marcam.dsir.govt.nz

Abstract

Genstat mode, an interface to Genstat for Gnu Emacs, is described. The interface allows users to run Genstat within an Emacs window and provides facilities for loading Genstat source files by file, by selected region or by line editing. Errors in Genstat source files can be found quickly using the next error function. Also provided is a facility for editing Genstat factors, variates and texts. Emacs mode for Genstat combines the advantages of batch mode (where the user has a record of exactly what calculations were done, and where the user can view an output file using a text editor) and interactive mode (where the user may respond to errors or adjust calculations on a line by line basis depending on the results of intermediate calculations). The ability of Emacs to split windows horizontally or vertically allows users to simultaneously view Genstat output, data files and source code.

1. Introduction

Emacs

Gnu Emacs (Stallman 1986) is an editor which features multiple windows, multiple buffers, advanced search and replace capabilities (e.g., incremental search whereby the search string is searched for as the user types it in and full regular expression capabilities, i.e. searching and/or replacing patterns), parenthesis matching (Emacs will let you know if you type in a mismatched parenthesis, and will flash the matching parenthesis; this is useful for avoiding syntax errors when programming), and the ability to run processes in windows. Emacs can be customised for particular users or applications. Simple customisation may consist of choosing between case sensitive or case insensitive searches or by making one's own definitions for keys. Major customisation is done using Emacs Lisp.

Emacs Lisp is a programming language similar to Common Lisp, the modern standard Lisp, with a clever mechanism for obtaining values of arguments of functions from users. It is this Lisp language which makes Emacs fully customisable; one can interactively define new editing functions or redefine existing functions. This can be done at the press of a key and does not require any recompiling or linking.

Keys are denoted by, e.g., `M-x` for *meta*-x, `C-x` for *control*-x, `C-x b` for *control*-x followed by b. The spaces are for readability only and are not typed. On most terminals the meta key is the escape key. Any single or multiple key sequence can be defined to run any command except where a sequence is an initial subsequence of a sequence already defined. In practice this is not as complicated as it sounds: note that most multiple key commands begin with `M`, `M-x`, `C-x`, `C-x 4`, `C-h` or `C-c` and end with one further key. The `4` means 'alternate' (e.g., `C-x f` is `find-file`, `C-x 4 f` is `find-file-other-window`), the prefix `C-h` is for help and the prefix `C-c` is used for mode specific commands.

By writing programs in Emacs Lisp users can do almost anything (except leap tall buildings in a single bound). Special modes have been created for many programs and languages including Ada, C, Fortran, ftp, lisp, mail, news, Pascal, prolog, S, telnet, TeX, LaTeX, and now Genstat.

Emacs functions (either internal functions or Emacs Lisp functions written by users) can be activated in several ways:

- automatically when the user types a key sequence which has been bound to the function

- by name when the user types **M-x <command-name>**, e.g., **M-x spell-buffer**
- by being called by another Emacs function
- when the user invokes a Lisp expression e.g., **M-M** followed by **(describe-bindings)** or **(delete-region (point-min)(point-max))**

Novice users need only consider the first and possibly the second method.

Customisation for Genstat – Advantages of using Emacs

Genstat mode consists of a number of Emacs Lisp functions, together with modifications of syntax tables. The modified syntax tables allow Emacs to match the parentheses used by Genstat. This helps catch syntax errors as they occur.

The function **genstat** runs an inferior Genstat process with input and output in an Emacs window. An Emacs window is simply part of the user's screen delimited by an inverse video *mode line*, which contains the name of the file or *buffer* (one edits *buffers* which may or may not correspond to files on the system) being edited. Thus the advantages of multiple windows are available for users with an ordinary character based terminal. Even when using a window system such as X–windows, experienced Emacs users prefer running programs such as Genstat within Emacs, to running an editor and Genstat in several windows and cutting and pasting between windows. This is because Emacs commands are more quickly accessible and more powerful than mouse based scrolling and menu operations. For example, incremental search is much quicker than scrolling back through a large output file; **M-! ls** is quicker than **SUSPEND ['ls']**; repeatedly typing **C-c C-n** in a Genstat source buffer enables evaluating a source file one line at a time much more quickly than repeatedly cutting and pasting with a mouse; **help** does not interfere with output; and so on.

Similarly, when editing Genstat data objects (cf. Lane 1991) it is not necessary to enter and exit an editor, **C-c d x** is much quicker to type than **EDATA x**, and the edited data remains visible in its buffer after being sent to back to Genstat. At present, editing one or more Genstat factors, variates or texts in a buffer is supported (multiple edit buffers are possible). Help is available on Genstat directives and library functions: online help appears in a pop-up window without cluttering up the Genstat output and forcing calculations to scroll off the screen. A history of previous commands is maintained (a feature not available in some current Genstat releases when running Genstat directly), and previous commands can be searched for with **C-c r**. The next error function **C-c n** (**genstat-eval-buffer-to-next-error**) facilitates corrections of Genstat source files containing errors. This works by automatically sending one line at a time starting from the cursor position to the Genstat process until Genstat reports an error **C-c n**. The cursor is left on the offending line. The user then need only correct this line and retype **C-c n**, continuing until all errors have been fixed.

2. Genstat modes for Emacs

When editing in Emacs, a buffer is usually in some *major mode* and one or more *minor modes*. There are three major modes for Genstat: *inferior genstat mode*, *genstat mode* and *genstat object mode* for running Genstat in a window, editing Genstat source files, and editing Genstat variates, factors or texts respectively. The mode of the current buffer is indicated by the mode line at the bottom of the window. Genstat mode can be invoked for a buffer by typing **M-x genstat-mode**. Normally, however, Genstat mode is invoked automatically when a file with suffix **.gen** is edited.

Running Genstat – Inferior Genstat Mode

I will assume the appropriate installation has been carried out. If not see the section on installation below.

Type **M-x genstat**. You should now be in a buffer running Genstat. You can type in commands to Genstat as if you were running Genstat directly. You can also do other editing commands such as scrolling, searching, query-replace etc. Typing **<return>** sends the line up to the current cursor position to the Genstat process, if you are not at the end of the line, or the whole line otherwise. Multiple lines can be typed in by ending them with **C-j**. Typing **<return>** after these lines sends them all. Similarly text can be cut and pasted from other buffers.

Some useful commands are:

Key	Emacs command	Definition
`<return>`	`comint-send-input`	send input to Genstat
`M-p`	`comint-previous-input`	get the previous command (from the command history) on command line
`M-n`	`comint-next-input`	get next command on command line
`C-c r`	`comint-previous-input-matching`	reverse search command history
`C-c d`	`genstat-dump-object-into-scratch`	dump an object or objects into a scratch buffer for editing (see below)

Editing Genstat source – Genstat Mode

Type `C-x 4 f test.gen` (get the file `test.gen` in the other window). If there is not already Genstat source in this file include some with `C-x i <file>` where `<file>` is the name of a file containing Genstat source, or type some in. You should now be in Genstat mode.

Some useful commands are:

`C-c b`	`genstat-eval-buffer`	send the buffer to the Genstat process [1]
`C-c C-b`	`genstat-eval-buffer-and-go`	send the buffer to the Genstat process and switch to the Genstat process buffer
`C-c r`	`genstat-eval-region`	send the region to the Genstat process [2]
`C-c R`	`genstat-eval-region-and-go`	send the region and switch to the Genstat buffer
`C-c l`	`genstat-load-file`	load a file, i.e. send a disk file to the Genstat process
`C-c h`	`genstat-display-help-on-command`	get help on a directive or library procedure (see below)
`C-c k`	`genstat-eval-line`	send the current line to Genstat
`C-c C-n`	`genstat-next-line-and-eval`	send the current line to Genstat and move down one line ready to send the next (useful for stepping through a file one line at a time)

New commands for finding errors

`C-c C-e`	`genstat-eval-buffer-to-first-error`	send lines one at a time start from the start of the buffer until an error occurs leaving the cursor on the line containing the error
`C-c n`	`genstat-eval-buffer-to-next-error`	send lines one at a time starting from cursor position until an error occurs, leaving the cursor on the line containing the error
`C-c ?`	`genstat-show-last-error`	display the full text of the last Genstat error message generated by `genstat-eval-buffer-to-first-error` and `genstat-eval-buffer-to-next-error`.

Notes:

(1) use `genstat-eval-buffer-to-first-error` or `genstat-eval-buffer-to-next-error` if the buffer may contain errors

(2) the block of text between the cursor position and where the *mark* was last set

The rationale for these key definitions is as follows. Where possible the last key of a key sequence is the first letter of the type of object to which the command applies, e.g., `b` for buffer, `k` for current line (`C-c c` would

be dangerous as the similar sequence `C-c C-c` sends the `C-c` signal to the process, and this kills Genstat), `l` for load and `r` for region. Also where possible definitions are kept compatible with similar modes especially S (mode for new S (Becker *et al* 1988) upon which Genstat mode is based) or definitions for other prefixes e.g., `C-x C-e` is for compile, `C-c C-e` is for the mode specific equivalent of compiling, namely finding the first error, `C-x C-n` is for next-error (after compiling) and `C-c n` is used for the mode specific equivalent namely finding the next error starting from the cursor position. `C-x C-n` was reserved for sending one line at a time as this sequence can be repeated rapidly, and `C-n` is the key for next line. Note that control keys are often variants of the corresponding non-control key, e.g., `C-c b` for evaluating the buffer and `C-c C-b` for evaluating the buffer and going to the genstat window.

Editing data – Genstat Object Mode

Type `C-c d x` in the Genstat process buffer to edit a single factor, variate, or text `x`. A buffer appears with a representation of `x` and, if applicable, nvalues, levels, labels, etc. If `x` has many values rows of output are labelled with the ordinal number of the first element, namely

```
 [1]  1  2  3  4  5  6  7 ...
[31] 31 32 ...
```

Multiple objects of the same length can be edited with, e.g., `C-c d x,y,z` or `C-c d x[]`. Emacs prompts you for the list of Genstat objects to edit after you type `C-c d`. When multiple objects are being edited the edit buffer contains values only and rows of output are labelled with their ordinal number. Values can be edited and the results read back into Genstat with `C-c b` or `C-c C-b`.

3. Getting help

Type `C-h c`, `C-h k` (`describe-key`) to see what a given key or sequence does, `C-h m` (`describe-mode`) for information on the current mode, `C-h b` (`describe-bindings`) for information on all key bindings.

To get help on Genstat directives and library procedures type, e.g., `C-c h kolmogorov`. A window pops up containing full help information on the directive or library procedure. Names may be abbreviated; if abbreviating type a space after the abbreviation and Emacs will complete the command or show alternatives.

4. Summary

Emacs mode for Genstat provides an integrated environment for running Genstat, editing data, editing and debugging Genstat programs, and getting help. The system is both easy to learn with online help and ideal for power users, providing the best of both worlds: running in batch and interactive mode. Further customisation can be readily done by users who can program in Emacs Lisp (e.g., for reading particular data formats), but for most users no Lisp knowledge is needed.

Acknowledgement

Genstat mode is based on the equivalent mode for the New S language (Becker *et al* 1988) which was written by Doug Bates, Ed Kademan and Frank Ritter.

References

Becker R A, Chambers J M and Wilks A R (1988) *The new S language, a programming environment for data analysis and graphics.* Wadsworth & Brooks/Cole.

Lewis B, LaLiberte D and the GNU Manual Group (1989) *GNU Emacs Lisp Reference Manual.* Free Software Foundation, Cambridge MA, USA.

Lane P W (1991) Editing Data Structures, *Genstat Newsletter No. 26*, Numerical Algorithms Group, Oxford.

Stallman R (1986) *GNU Emacs Manual* (Fifth Edition) Free Software Foundation, Cambridge MA, USA.

Appendix A: Getting Gnu Emacs

Gnu Emacs is available free (subject to Gnu copyright provisions, the main condition is roughly that you cannot sell or restrict access to Emacs) via ftp from, e.g., tut.cis.ohio-state.edu, cc.utah.edu (VAX-VMS).

Appendix B: Installation

If Gnu Emacs has not been installed on your system see the section Getting Gnu Emacs. If necessary edit the lines :

```
(defvar inferior-genstat-program "genstat" ...
(defvar explicit-genstat-args "S=5" ...
```

in the file **genstat.el**, replacing **genstat** by the command needed to run genstat and **S=5** by the appropriate explicit arguments given to Genstat. Install the files **comint.el** (available from, e.g., tut.cis.ohio-state.edu under **/pub/gnu/emacs/elisp-archive/modes/** in the **cmushell.shar.*** files), **genstat.el** in the directory where Emacs looks for Emacs Lisp files. Add the following lines to your **.emacs** file or equivalent after changing **"\~/elisp"** to the appropriate path.

```
(autoload 'genstat "~/elisp/genstat.el" "" t)
(autoload 'genstat-mode "~/elisp/genstat.el" "" t)
(setq auto-mode-alist
  (cons (cons "\\.gen$" 'genstat-mode) auto-mode-alist))
```