Issue No. 22 *1988(B)*

The GENSTAT Newsletter

NAG®

Editors

P W Lane
Rothamsted Experimental Station
HARPENDEN
Hertfordshire
United Kingdom AL5 2JQ

K I Trinder
NAG Limited
Wilkinson House
Jordan Hill Road
OXFORD
United Kingdom OX2 8DR

Printed and produced by the Numerical Algorithms Group

ISSN 0269-0764

Genstat Newsletter
Issue No. 22

NP1831

October 1988

Contents

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

Editorial

Following the request in the editorial of Issue 21, we are glad to see several short articles in this issue, focussing on particular applications and details of Genstat. We hope this will encourage more users to write similar articles for future issues.

This issue also contains a report on the one-day conference on 'The analysis of repeated measure in Genstat', held at Rothamsted on 6 October 1988. Following this, from 21-23 November 1988 there was a three-day Genstat conference in Melbourne, Australia, attended by over 60 users from Australia and New Zealand. We hope to see a report on this in the next issue, but note now that a decision was taken at that conference to set up an Australasian users' group, based on communication by electronic mail. How about some more of these? If you would be interested in laying the groundwork for such a group, please contact the editors. It is, of course, essential for healthy activity that users' groups are set up independently from the developers and distributors of Genstat, so we ask you to give the idea serious consideration. We will be glad to publish in this newsletter any news about user groups.

Two more implementations of Genstat 5 have recently been completed and are now available. These are for the HP9000 800 Series HP-UX and Sun 3 SunOS systems. The Sun 3 version has interfaces to the SunCGI and GKS graphical systems. Regretably, the HP9000 800 series version does not have any graphics interfaces in this release, although it is still planned that a later release will have the standard interfaces. Versions for the Sun 386i and Sun 4 should follow the Sun 3 version quite soon and in fact a version has been formed on a Sun 4 although it had not been fully tested at the time of writing this editorial.

Also at the time of writing, Genstat 5 Release 1.3 for VAX VMS was about to be sent to sites and should arrive within a few weeks of this Newsletter, depending on the Christmas post. The main features are the inclusion of interfaces to the Regis, GKS, Ghost80 and Gino-F graphical systems and the updated Procedure Library, which is described fully in Roger Payne's article in this issue.

The PC version has quickly become the second most popular version (in terms of numbers of sites), in spite of it not having a graphics capability yet. We are pleased to report that progress has been made with incorporating graphics, but it is too early to say when this will be available to sites. Also, some users have told us of problems regarding the size of free memory required by Genstat (about 582 Kb) and we are looking at ways of improving the situation.

The next Genstat training course run by NAG will be held from 18-20 January 1988 in Birmingham. Further courses will be organized: contact NAG to find out details.

Arrangements are being made for the Genstat Conference in Edinburgh from 11-15 September 1988. Details of registration, and an invitation to submit an abstract for a presentation at the conference, are in an enclosure distributed with this issue.

One-day Conference on the Analysis of Repeated Measures in Genstat, 6 October 1988

P W Lane
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Hertfordshire
United Kingdom *AL5 2JQ*

This was the second in what is intended to be a series of conferences about selected topics in Genstat. (The first was held on 28 April 1988, and was about the new methods for extending Genstat with Fortran.) It was again held in the new Conference Hall at Rothamsted, and attracted over 100 participants. As well as the eight invited presentations, there was the opportunity to see and try out Genstat on an Opus PC-V, an IBM PS/2 and two Sun workstations.

D.E. Walters, of the Institute of Animal Physiology and Genetic Research, started the proceedings with an overview of available methods for analysing repeated measures. These range from what some authors misleadingly call the 'usual method', which means totally ignoring the effects of correlation between observations, through the modified split-plot analysis, to detailed models for the covariance structure of the observations. His opinion was that the analysis of summary statistics, derived for the set of repeated measurements on each subject, was nearly always the best approach, being straightforward to comprehend, compute and communicate.

M.G. Kenward, of the University of Reading, followed with a detailed look at one of the more recently developed methods, called 'ante-dependence analysis'. This involves the use of covariance analysis, treating previous observations in the series for each subject as the covariates. The method is readily implementable in Genstat, as was shown later by M.S. Ridout of the Institute of Horticultural Research. He has developed a procedure with R.W. Payne of Rothamsted for selection of a suitable order of ante-dependence; that is, the number of covariates to include in the analysis. This procedure will be added to the Genstat Procedure Library.

A. Keen, of the Agricultural Science Group at Wageningen in The Netherlands, gave several examples to illustrate ante-dependence analysis. He concentrated particularly on techniques of curve analysis, where polynomial curves are fitted for each subject and the resulting scores are assessed by analysis of variance or covariance, using lower-degree scores as the covariates.

G. Tunnicliffe Wilson, of the University of Lancaster, showed how some problems in repeated measures could be tackled using standard methods available in Genstat for analysing time series. Models for autoregressive or moving-average behaviour of a series of observations for a subject can be assessed, with the distinction between subjects being made by including a 'seasonal' effect in the model.

Two speakers from the Glasshouse Crops Research Institute talked about specific projects with which they had been involved. R. Edmondson described the analysis of growth curves for data from tomato experiments, where many successive harvests are taken from individual plants during the growing season. He fitted linear polynomials to provide an analysis of the cumulative yield curves with respect to the effects of treatments and the variation of treatment effects over environments. Then J.S. Fenlon described the analysis of time-to-response in insect assays, where quantal data are collected at a series of times after application of an insecticide to monitor its effectiveness. Often, a statistic such as the time taken to destroy 50% of the insects, is the goal of an analysis, but such statistics should not be produced by standard methods such as probit analysis, which ignore correlation.

R.W. Payne, of Rothamsted, concluded the meeting by summarizing the facilities available in Genstat for the analysis of repeated measures. As well as standard techniques available in all the major statistical areas (regression analysis, analysis of variance, multivariate analysis and analysis of time series) there was considerable scope for the use of Genstat procedures to provide more complicated, recent techniques. He stressed the convenience of the standard facilities for data handling and presentation, particularly the use of high-resolution graphics to display data and models for repeated measures, as had been demonstrated by many of the speakers.

Genstat Procedure Library for Release 1.3

R W Payne
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Hertfordshire
United Kingdom *AL5 2JQ*

The second release of the Library, which will be sent out with Release 1.3, contains 51 procedures and covers many of the more popular analyses in the Genstat 4 Macro Library – as well as some new facilities. Index lines for the procedures in the Library are as follows:

ALIAS	finds out information about aliased model terms in analysis of variance
AONEWAY	provides one-way analysis of variance for inexperienced users
APLOT	plots residuals from an ANOVA analysis
BARCHART	plots a bar chart using line-printer or high-quality graphics
BIPLOT	produces a biplot from a set of variates
CANCOR	does canonical correlation analysis
CENSOR	pre-processes censored data before analysis by ANOVA
CHECKARGUMENT	checks the arguments of a procedure
CLASSIFY	obtains a starting classification for non-hierarchical clustering
CONCORD	calculates Kendall's Coefficient of Concordance for a set of variates
CORRESP	does correspondence analysis, or reciprocal averaging
DDENDROGRAM	draws dendrograms with control over structure and style
DESCRIBE	saves and/or prints summary statistics for variates
DISCRIMINATE	performs discriminant analysis
DSHADE	produces a shaded similarity matrix by high-quality graphics
FIELLER	calculates effective doses or relative potencies
GENPROC	performs a generalized Procrustes analysis
GLM	analyses non-standard generalized linear models
GRBETA	generates pseudo-random numbers from the beta distribution
GRCHI	generates pseudo-random numbers from the chi-square 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
KOLMOG2	performs a Kolmogarov-Smirnoff two-sample test
KRUSKAL	carries out a Kruskal-Wallis one-way analysis of variance
LATTICE	analyses square and rectangular lattice designs
LIBEXAMPLE	accesses examples and source code of Genstat 5 Library procedures
LIBHELP	provides help information Genstat 5 Library procedures
LIBINFORM	prints information about the contents of the Procedure Library
LIBMANUAL	prints a manual containing information about Library procedures
MANCOVA	performs a multivariate analysis of covariance

MANNWHITNEY	performs a Mann-Whitney U test
MANOVA	performs a multivariate analysis of variance
MULTMISS	estimates missing values for units in a multivariate data set
NOTICE	gives access to the Genstat Notice Board (news, errors etc.)
NPCHECK	checks the validity of input data for nonparametric procedures
NPRANK	produces ranks, allowing for ties, for the nonparametric procedures
ORTHPOL	calculates orthogonal polynomials
PERCENT	expresses the body of a table as percentages of one of its margins
QUANTILE	calculates quantiles of the values in a variate
RCHECK	checks the fit of a linear or generalized linear regression
REPMEAS	checks if a set of repeated measures can be analysed as a split plot
SKEWSYMM	provides an analysis of skew-symmetry for an asymmetric matrix
SPEARMAN	calculates Spearman's Rank Correlation Coefficient
SUBSET	forms vectors containing subsets of the values in other vectors
TTEST	performs a one-sample or two-sample t-test
VHOMOGENEITY	tests homogeneity of variances
WILCOXON	performs a Wilcoxon Matched-Pairs (Signed-Rank) test

These were produced from within Genstat by the library procedure LIBINFORM, the relevant statement being

```
LIBINFORM [PRINT=index]
```

It is also possible to produce a 'manual' collating the Help information on the Library, using the procedure LIBMANUAL. (Full details of the syntaxes of these or any of the other procedures in the Library can be obtained using procedure LIBHELP, as explained on page 597 of the Genstat 5 Reference Manual.)

Most of the procedures in the current library have been written by authors or have involved co-authors from Rothamsted, but procedures are beginning to be submitted from other sites and others would be very welcome. Aspiring authors are encouraged to contact the secretary of the Library's Editorial Committee, at Rothamsted, for advice and to try to avoid duplication of effort. Instructions for authors were published in Genstat Newsletter No. 20. Authors may also find it useful to study some of the existing procedures; procedure LIBEXAMPLE has been revised to allow the source code of any Library procedure to be copied into a Genstat text.

The next release of the Library will be formed in January 1989. Among the additions will be procedures to help with the analysis of time series, using graphical displays to select and check the fit of suitable Box-Jenkins models; also further procedures for analysing repeated measures data, produced for the one-day meeting described elsewhere in this issue.

Comparison of Algorithms for Generalised Procrustes Analyses

G M Arnold
Department of Agricultural Sciences
University of Bristol
AFRC Institute of Arable Crops Research
Long Ashton Research Station
Bristol
United Kingdom *BS18 9AF*

Generalised Procrustes analysis is a method for matching several two-way configurations, such as data on V variables observed on N subjects. The method makes use of translation to a common origin, rotation/reflection of axes and, possibly, an isotropic scale change. With more than two configurations this matching is done iteratively to a common consensus configuration to minimize a goodness-of-fit statistic which is the sum of the Procrustes statistics of each adjusted configuration to the consensus. In recent years this technique has been used extensively in sensory work, particularly for analysis of profile data [1,2]. In this context each of the M configurations represents the scores for one assessor of N samples on V sensory attributes. With the possibility of many assessors, samples and attributes, large data sets may be generated and, in these circumstances, a generalised Procrustes analysis can be very expensive in computing time.

The Genstat 4 Macro Library contained a macro, GENPROC, for generalised Procrustes analysis which used the method described by Gower [3]. Other approaches to the rotation/reflection and scaling stages of the analysis have been described in the literature [4,5,6]. This article discusses the programming of some of these algorithms in Genstat 5 and compares their performance on three large data sets of dimensions typical of those produced in sensory profiling.

The steps for all the algorithms compared herein can be summarised similarly:

1. Centre each input matrix X_i $(i = 1...M)$ and scale each X_i by $\frac{M}{\sum trace(X_i'X_i)}$
2. Evaluate an initial estimate of the centroid C by setting $C = X_1$; then for $i = 2...M$ rotate X_i to C and re-evaluate C as the mean of the new $X_1...X_i$.
3. Calculate the initial residual sum-of-squares and set the scaling factors p_i $(i = 1...M)$ to have initial values of 1.
4. For $i = 1...M$, rotate the current X_i to C. Evaluate the new centroid as the mean of the new X_i and calculate a new residual sum-of-squares.
5. If isotropic scaling is not required, go to step 7.
6. For $i = 1...M$, evaluate the new scaling factors p_i'. Calculate the new X_i as $p_i'X_i$. Evaluate the new centroid as the mean of these new X_i $(i = 1...M)$ and calculate a new residual sum-of-squares.
7. If the reduction in residual sum-of-squares from step 3 is greater than a preset tolerance (e.g. 0.0001) save the current residual sum-of-squares from the previous step and go to step 4 to repeat the process.
8. Refer the final centroid to its principal axes to give the final consensus configuration. Refer the individual final configurations X_i $(i = 1...M)$ to the same axes and print results.

Steps 1-3, 5, 7 and 8 are common to all the approaches. The algorithms differ only in the method of rotation/reflection in step 4, and the calculation of the scaling factors in step 6; all methods converge to the same solution.

The differences in the approaches will now be described and the respective Genstat 5 statements given, operating on the following data structures:

Scalars

`Nconfig`	– number of configurations/assessors (M)
`Nrows`	– number of rows/samples (N)
`Ncols`	– number of columns/attributes (V)
`Nentry`	– number of values per configuration (NV)
`S2`	– trace of $C'C$ where C is the current centroid
`S`	– calculated new scaling factor for a configuration

Variates of length NV

`Xvar[1...Nconfig]`	– each current X_i stored as a variate

Variate of length M

`ScalingF`	– scaling factors for all configurations stored as a variate

Matrices of dimension N and V

`Xout[1...Nconfig]`	– current X_i, the configuration for assessor i
`X`	– dummy for X_i in FOR loops
`Y`	– current centroid $C = \frac{\sum X_i}{M}$
`Z`	– current $\sum X_i$ or zero

For the rotation/reflection stage (step 4) Gower [3] rotates each current X_i ($i = 1...M$) to the initial centroid C. After these M rotations the new centroid is calculated as the mean of the new X_i ($i = 1...M$). At the start of these calculations, `Z` holds the sum of the current X_i and `Y` the current centroid.

```
CALCULATE Z = 0
FOR Config=1...Nconfig
  ROTATE [STANDARDIZE=centre] XINPUT=Y; \
     YINPUT=Xout[Config]; YOUTPUT=Xout[Config]
  CALCULATE Z = Z+Xout[Config]
ENDFOR
CALCULATE Y = Z/Nconfig
```

Kristof and Wingersky [4] suggest that instead of updating the centroid after each cycle of M passes, it should be updated after each pass.

```
FOR Config=1...Nconfig
  CALCULATE Z = Z-Xout[Config]
  ROTATE [STANDARDIZE=centre] XINPUT=Y; \
     YINPUT=Xout[Config]; YOUTPUT=Xout[Config]
  CALCULATE Y = (Z = Z+Xout[Config])/Nconfig
ENDFOR
```

If p_i is the current scaling factor for configuration X_i, Gower [3] calculates the new estimate of the scaling factor, p'_i, by

$$p'_i = p_i \sqrt{\frac{trace(X'_i C)}{trace(C'C)\ trace(X'_i X_i)}}$$

```
CALCULATE Z = 0
FOR X=Xout[1...Nconfig];Config=1...Nconfig
  CALCULATE Z = Z + (X = X* \
     (S = SQRT(TRACE(TRANSPOSE(X)*+Y)/(S2*SUM(X*X)))))
  CALCULATE ScalingF$[Config] = ScalingF$[Config]*S
ENDFOR
CALCULATE Y = Z/Nconfig
```

The values of the scaling factors p'_i thus calculated are not the best possible estimates at this stage. Langron [5] suggests amending this calculation to estimate the p'_i more accurately by inserting an extra test of the differences in residual sums-of-squares. If the reduction in residual sums-of-squares before and after step 6 as calculated by the Gower method above is greater than a preset tolerance, this step is repeated until the reduction obtained is less than the tolerance.

Ten Berge [6] gives an alternative method of estimating the scaling factors at step 6, as well as recommending the use of the rotation/reflection procedure of Kristof and Wingersky [4]. Writing each configuration X_i as a variate of length NV, the matrix of correlation coefficients between these variates is calculated. If E_1 is the eigenvector corresponding to the largest eigenvalue of this correlation matrix, the new best estimate of the scaling factor, p'_i, is given by

$$p'_i = p_i \sqrt{\frac{M}{trace(X'_i X_i)}} e_{i1}$$

where e_{i1} is the i^{th} element of E_1.

```
SSPM [TERMS=Xvar[1...Nconfig]] XvarSSPM
LRV [ROWS=Nconfig; COLUMNS=1] XcorLRV
CALCULATE Z = 0
EQUATE OLD=Xout[1...Nconfig]; NEW=Xvar[1...Nconfig]
FSSPM XvarSSPM
CALCULATE XvarSSPM['Sums'] = CORRMAT(XvarSSPM['Sums'])
FLRV XvarSSPM['Sums']; LRV=XcorLRV
FOR X=Xout[1...Nconfig]; Config=1...Nconfig
  CALCULATE Z = Z + (X = X* \
     (S = ABS(SQRT(Nconfig/SUM(X*X))*XcorLRV['Vectors']$[Config;1])))
  CALCULATE ScalingF$[Config] = ScalingF$[Config]*S
ENDFOR
CALCULATE Y = Z/Nconfig
```

The comparison of timings for these different methods programmed in Genstat 5 has been carried out on three large data sets with the same overall number of elements.

Dataset Number	No. of configurators (assessors: M)	No. of rows (samples: N)	No. of columns (attributes: V)	Total elements
1	32	48	25	38400
2	16	48	50	38400
3	8	192	25	38400

Table 1

Dimensions of Datasets used for timing comparisons

Firstly the two methods of rotation/reflection were compared with no isotropic scaling allowed. Then the methods of rotation/reflection were compared simultaneously with the three different scaling methods, giving six possible combinations. The timings presented are for Genstat 5 Release 1.3, running on a VAX 11/750, with the tolerance in step 7 set to 0.0001.

Method		Dataset					
Rotation/ Reflection	Scaling	1		2		3	
Gower	–	100	(208)	100	(184)	100	(87)
K & W	–	82		90		90	
Gower	Gower	100	(319)	100	(271)	100	(136)
Gower	Ten Berge	92		71		86	
Gower	Langron	109		92		112	
K & W	Gower	75		90		94	
K & W	Ten Berge	67		63		79	
K & W	Langron	87		84		111	

(Note: K & W refers to Kristof and Wingersky)

Table 2

Comparative timings (CPU) presented as a percentage of the baseline method (Gower). Absolute timings in minutes for these baseline runs are given in parentheses.

For each of the datasets used, the quickest run was for the method of Ten Berge which incorporates the rotation/reflection method of Kristof and Wingersky. Percentage savings in time ranged from 21% for dataset 3 to 37% for dataset 2. Kristof and Wingersky's rotation/reflection method gave greatest savings for dataset 1 which has the largest number of configurations (assessors). Comparing scaling methods, that of Ten Berge gave the greatest percentage reduction for dataset 2, which has the largest number of columns (attributes). The alternative scaling method of Langron does not appear to be efficient. These results suggest that the algorithm for generalised Procrustes analysis described by Ten Berge can reduce computation time considerably compared to that suggested by Gower, particularly for data sets with large numbers of configurations and/or columns.

Care should be taken in extending these results to programming languages other than Genstat. For example, if one were programming at the Fortran level, the number of program statements executed in the eigenvector calculations required by Ten Berge might well dominate the timings. Eigenvectors are calculated as a single Genstat statement, so the algorithmic part of the calculation is dominated by similar overheads to those required for the more simple executable arithmetic statements.

These findings have been taken into account in the design of the Genstat 5 procedure GENPROC, for generalized Procrustes analysis (written by G.M. Arnold and R.W. Payne) in the Library that accompanies Release 1.3, with the method of Ten Berge being made available as an alternative to that of Gower. Other options can request isotropic scaling, set the required tolerance and set a limit on the number of iterations allowed. Various results can be printed using the print option and/or saved via several parameters. Full details can be obtained when running Genstat 5 Release 1.3, by using the statement

```
LIBHELP [PRINT=index,authors,description,options,parameters,method] \
   'GENPROC'
```

Acknowledgement

Long Ashton Research Station is financed through the Agriculture and Food Research Council.

References

[1] Arnold, G.M.
A Generalised Procrustes Macro for Sensory Analysis.
Genstat Newsletter, **18**, pp. 61-80, 1986.

[2] Arnold, G.M. and Williams, A.A.
The Use of Generalised Procrustes Techniques in Sensory Analysis.
In: *'Statistical Procedures for the Food Industry'*, pp. 233-253, J.R. Piggott (ed.).
Elsevier, Applied Science Publishers Ltd., 1986.

[3] Gower, J.C.
Generalised Procrustes Analysis.
Psychometrika, **40**, pp. 33-51, 1975.

[4] Kristof, W. and Wingersky, B.
Generalisation of the Orthogonal Procrustes Rotation Procedure to More Than Two Matrices.
American Psychological Association, 79th Annual Convention Proceedings, pp. 89-90, 1971.

[5] Langron, S.P.
The Statistical Treatment of Sensory Analysis Data.
Ph.D. Thesis, University of Bath, 1981.

[6] Ten Berge, J.M.F.
Orthogonal Procrustes Rotation for Two or More Matrices.
Psychometrika, **42**, pp. 267-276, 1977.

Plotting Shade Diagrams and Cluster Dendrograms

C W Ramm
Department of Forestry
Michigan State University
East Lansing
Michigan
USA

Hierarchical cluster analysis in Genstat is a useful tool for the detection of natural groups. It can be difficult, however, to interpret individual clusters. A shade diagram, based on the similarity matrix used for clustering, may aid interpretation. This note reviews some Genstat 5 procedures, available in the standard Library with Release 1.3, for the joint plotting of a dendrogram and a shade diagram.

The most time-intensive procedures are those involved in building a similarity matrix and the dendrogram structure. Define a symmetric matrix (`S1`) and construct it using the `FSIMILARITY` directive. Use `S1` in the `HCLUSTER` directive to do hierarchical cluster analysis and save the amalgamations (`A1`), which will be used in the `DDENDROGRAM` procedure to construct the dendrogram. The similarity matrix (`S1`), amalgamations (`A1`), dendrogram structure (`dd1`) and permutations (`P1`) should be saved in a backing-store file. The permutations are used to sort the units within the similarity matrix so that their order matches the dendrogram. An example program is shown in Table 1; it constructs a dendrogram for 20 plots using a similarity matrix based on the relative abundence of 30 species [3].

```
OPEN 'Shadyden.bak'; CHANNEL=5; FILETYPE=backing
RETRIEVE [CHANNEL=5] SPP[1...30]
SYMMETRICMATRIX [ROWS=20] S1
FSIMILARITY [SIMILARITY=S1] SPP[1...30]; TEST=3
HCLUSTER [METHOD=groupaverage] S1; AMALGAMATIONS=A1
DDENDROGRAM [STYLE=centroid; GRAPHICS=lineprinter] A1; \
   TITLE='Dune Meadows'; PERMUTATIONS=P1; SAVE=dd1
STORE [CHANNEL=5; SUBFILE=dendro] S1,A1,P1,dd1
STOP
```

Table 1

Formation of similarity matrix and dendrogram structure.

The example program uses the option setting `GRAPHICS=lineprinter` rather than plotting the dendrogram. The lineprinter is faster, and the printout can be used to evaluate the ordering and style selected for the dendrogram. There are four possible settings for the `STYLE` option in the `DDENDROGRAM` procedure to choose from. It is suggested that `average` be used with average linkage; `centroid` with group linkage; `lower` with single linkage; and `full` with complete linkage.

```
OPEN 'Shadyden.bak'; CHANNEL=5; FILETYPE=backing
RETRIEVE [CHANNEL=5; SUBFILE=dendro] S1,P1,dd1
OPEN 'Shadyden.grd'; CHANNEL=1; FILETYPE=graphics
DEVICE 1
PEN 1...4; COLOUR=1; BRUSH=1,9,5,6
" define frame for shade diagram, key, dendrogram "
FRAME WINDOW=1,2,3; YLOWER=0.2,0.8,0.2; YUPPER=0.8,1,0.8; \
   XLOWER=2(0.5),0; XUPPER=2(1),.5
DSHADE S1; NGROUPS=4; PERMUTATIONS=P1
DDENDROGRAM [CHANGE=display; ORIENT=east; SCREEN=keep] \
   dd1; WINDOW=3; TITLE=' Dune Meadow '
STOP
```

Table 2

Example of program to plot dendrogram and shade diagram.

Shade diagrams may be plotted on any side of the dendrogram. The program given in Table 2 will plot the dendrogram on the left and the shade diagram on the right (Figure 1). The DSHADE procedure does not have the option of defining alternative windows: it will plot the shade diagram in window 1 and its key in window 2. Digby and Kempton [2] discuss alternative ways of combining shade diagrams and dendrograms. Their recommended format has the shade diagram plotted above the dendrogram, which is inverted or 'hanging' (Figure 2). As long as windows 2 and 3 have the same lower and upper limits in X, the two plots will line up. The windows for Figure 2 were defined by:

```
FRAME WINDOW=1,2,3; YLOWER=2(0.5),0.1; YUPPER=2(0.9),0.5; \
   XLOWER=0.25,0.75,0.25; XUPPER=0.75,1.0,0.75
```

and the DDENDROGRAM procedure call was revised to include the options REVERSE=yes and ORIENT=north. The details on the procedures DDENDROGRAM (written by P.G.N. Digby) and DSHADE (written by S.A. Harding) are available through the procedure LIBHELP in Genstat Release 1.3.

The PEN statement used in these examples defines four groups with specific brush selections to produce a gradient of shadings for the similarity matrix. A maximum of three to four groups is suggested. Use brush pattern 16 and different pen colours with a colour monitor.

The graphics file produced should be checked on a graphics terminal before plotting to ensure that labels and titles are legible. A fourth window can be defined to contain a title for the entire plot. Be aware that the shade diagram is generated by plotting individual histograms. Lines along the diagonal of the shade diagram are drawn repeatedly; when plotted, the ink may bleed into the paper. A heavier weight paper and thinner pens can help reduce the problem.

Joint plots appear to work best for relatively small data sets. Sixty or more observations appears to be the upper limit for one joint plot, both for legibility and for understanding the dendrogram. Figure 3, for example, shows the shade diagram and dendrogram based on relative abundence across 141 upland hardwood stands in Michigan of 56 species of ground flora. Anderberg [1] gives several methods to split large data sets for clustering. Finally, cluster analysis is primarily a technique for exploratory data analysis. Cluster analysis may not find patterns in data where they exist, and it may discover patterns where there are none. A single dendrogram, no matter how unique the clusters or complex the plot, should not be accepted as proof.

References

[1] Anderberg, M.R.
Cluster Analysis for Applications.
Academic Press, 1973.

[2] Digby, P.G.N. and Kempton, R.A.
Multivariate Analysis of Ecological Communities.
Chapman and Hall, 1987.

[3] Jongman, R.G.H., ter Braak, C.J.F. and van Tongeren O.F.R.
Data Analysis in Community and Landscape Ecology.
PUDOC, Wageningen. 1987.

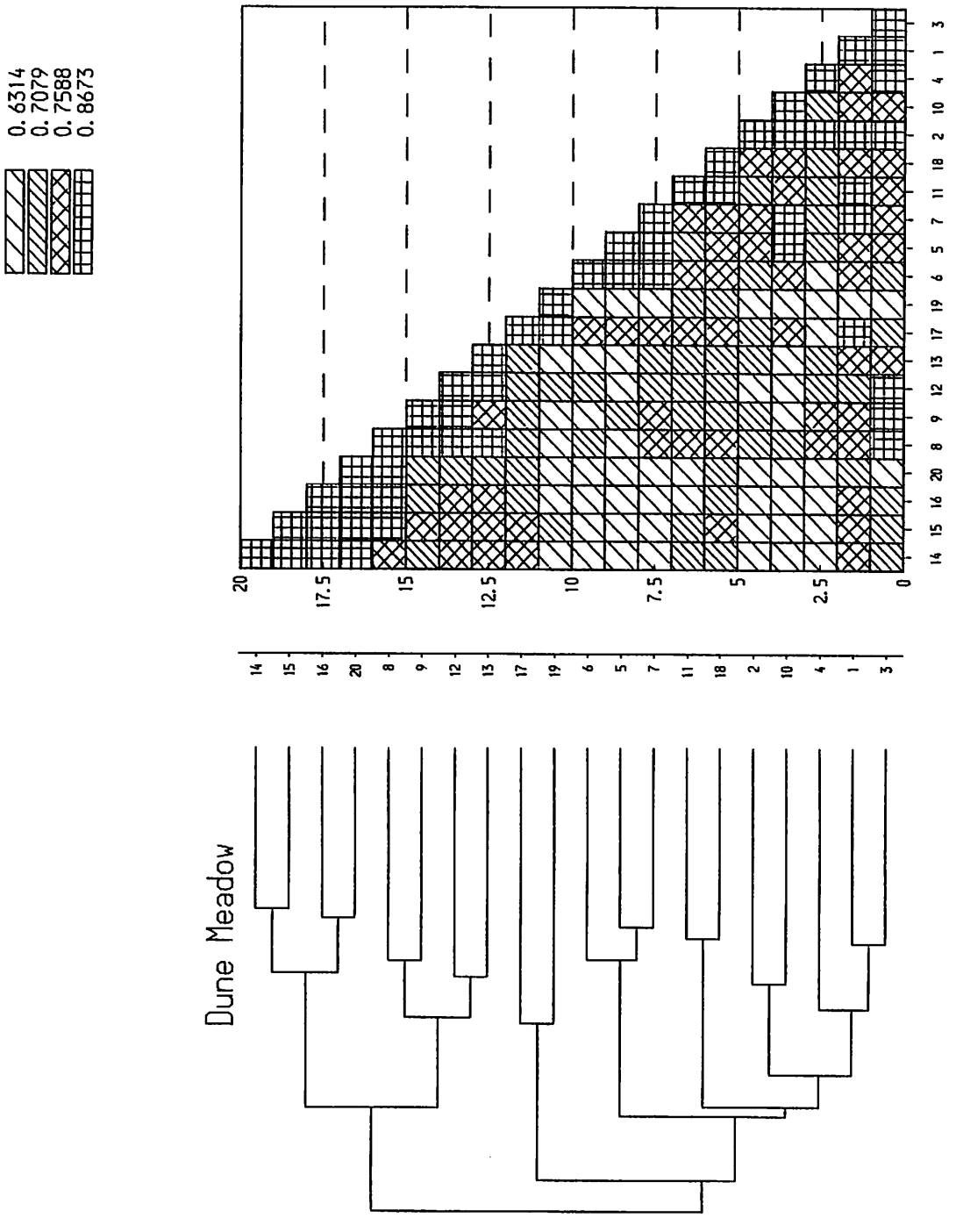


Figure 1

Side-by-side dendrogram and shade diagram for 20 plots, based on abundance values for 30 plant species.

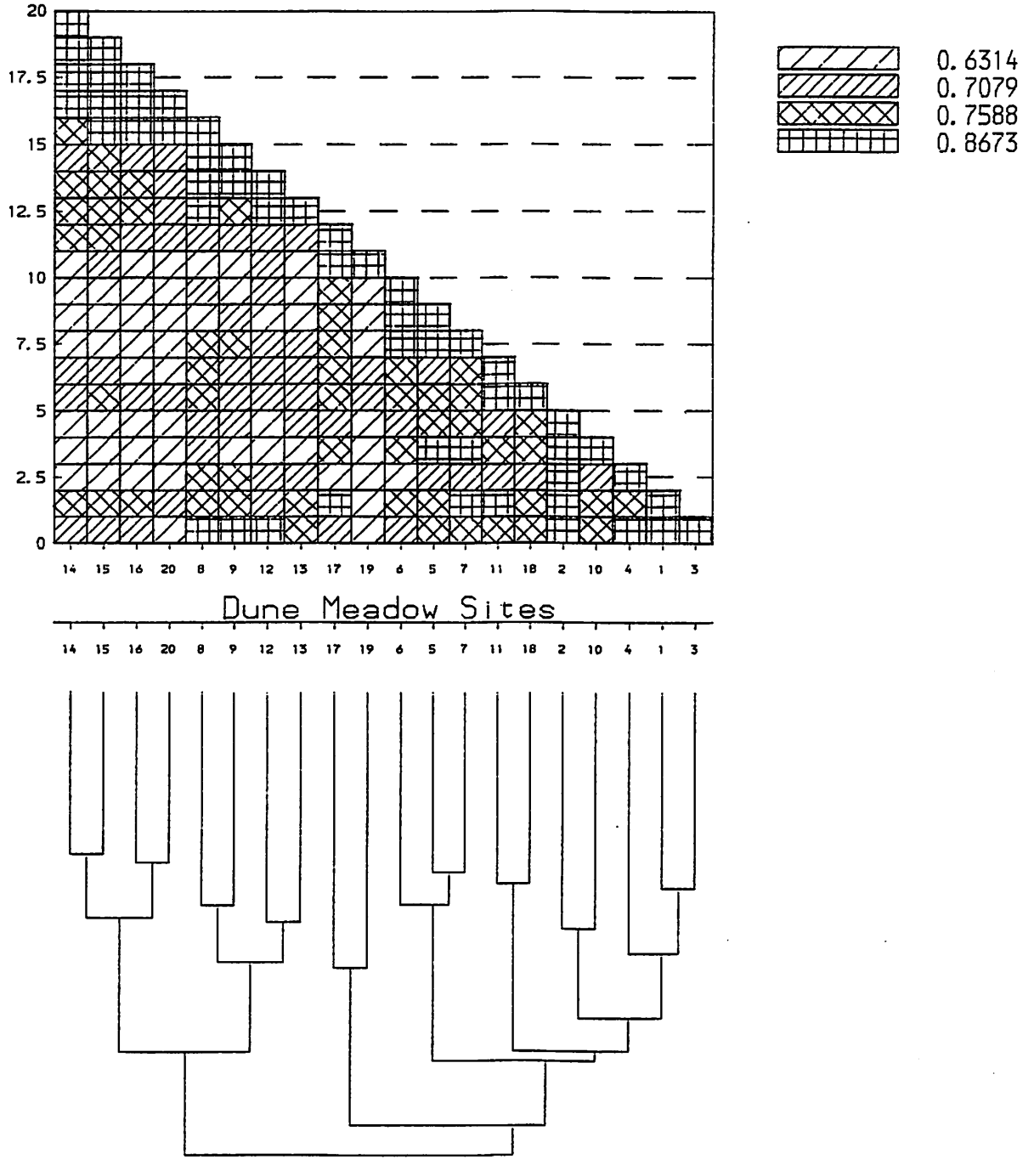


Figure 2

Shade diagram and 'hanging' dendrogram for same data.

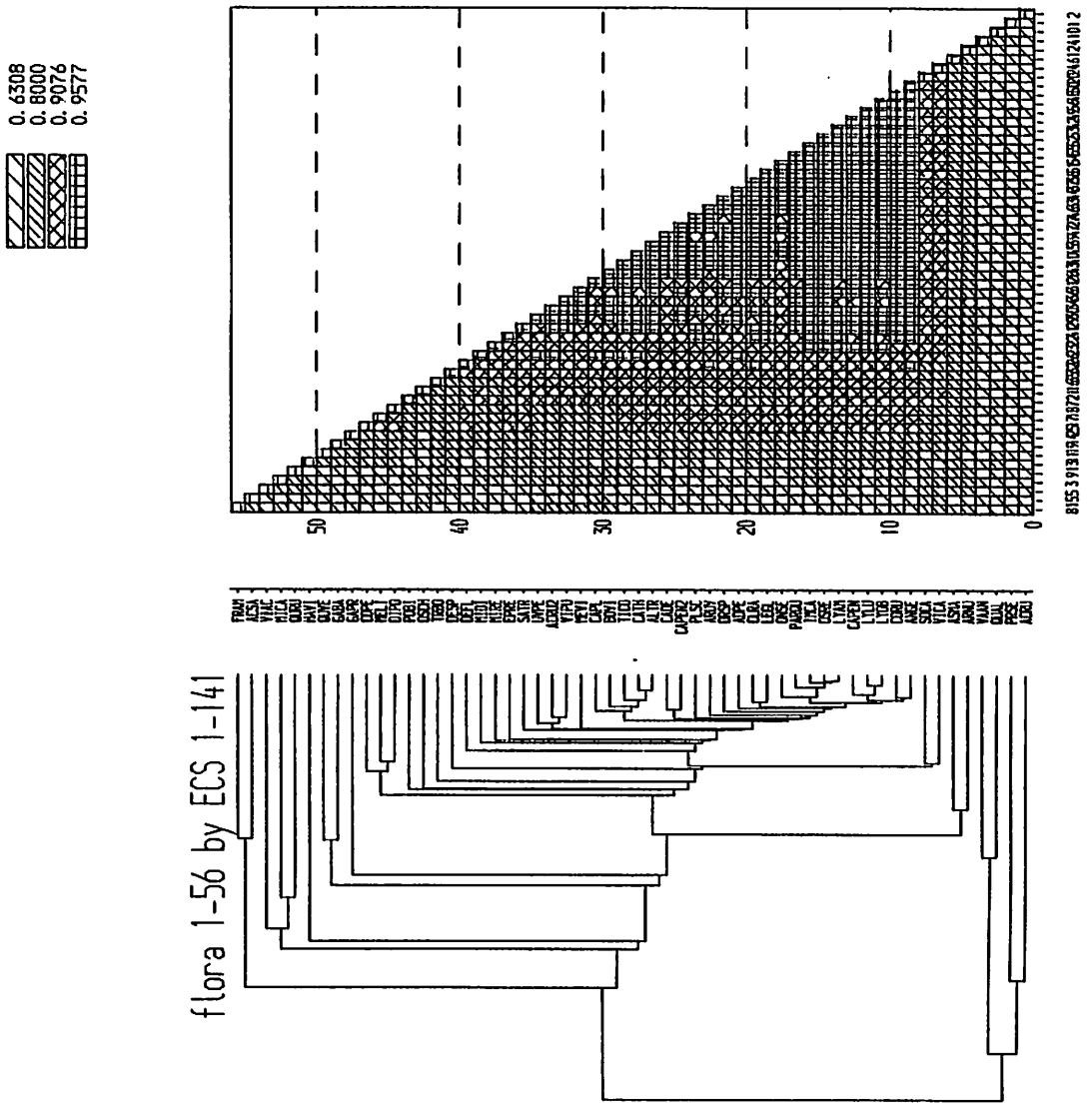


Figure 3

Paired shade diagram and dendrogram based on abundance values on 141 upland hardwood stands for 56 understory species.

Motor Vehicle Speeds in Built-up Areas: Some Comparisons of the Logit, the Log-linear Poisson, and the Log-linear Binomial Models

L K Thomsen
Danish Council of Road Safety Research
Ermelundsvej 101
DK-2820 Gentofte
Denmark

1. Introduction

This paper describes the use of three models all relevant for road safety research. The three models are the log-linear Poisson model, the logit model, and what is called the log-linear binomial model.

The structure of this paper is a brief presentation of the models, their use, and finally some concluding remarks.

2. The Three Models

Expressed as formulae, the three models considered are:

the log-linear Poisson

$$\log(u_{ij}) = \alpha_i + \beta_j \tag{1}$$

the logit

$$\log\left(\frac{\rho_{ij}}{1-\rho_{ij}}\right) = \gamma_i + \delta_j \tag{2}$$

and the log-linear binomial

$$\log(\rho_{ij}) = \varepsilon_i + \varsigma_j \tag{3}$$

Almost any textbook on categorical data deals with the models (1) and (2) often leaving out the log-linear binomial model. A discussion and application of (3) to Bartlett's data is provided by Thyregod and Spliid [7].

3. A Simple Example

Fienberg [3, p. 8] uses, as an introduction to his book on log-linear Poisson and logit models, the example treated below. The point in this paper is that the data presented by Fienberg, shown here as Table 1, is analysed with advantage (in terms of interpretation of results) by the log-linear binomial model.

		Cold	No Cold	Totals
Treatment	Placebo	31	109	140
	Ascorbic Acid	17	122	139
	Totals	48	231	279

Table 1

Incidence of common colds in a double-blind study involving 279 French skiers.
Pauline's data as given by Fienberg [3].

Application of the log-linear binomial model leads under the hypothesis of no treatment effect to

$$\log(\rho_i) = c \tag{4}$$

simply saying that the proportion of skiers catching a cold does not change due to the treatment.

Another hypothesis could be to include a parameter allowing the treatment to affect the proportion of skiers catching a cold, i.e.

$$\log(\rho_i) = c + \alpha_i \qquad (5)$$

The formulation and use of these models is simple by means of packages such as Genstat and GLIM, [4], although it demands the involvement of macros as shown in the Appendix. Table 2 provides the output from fitting models (4) and (5) using GLIM.

```
$F $L %X2 $D MER $CA %EXP(-1.760) $

scaled deviance = 4.8717 at cycle  2
          d.f. = 1

   4.811
Current model:

  number of units is  2

  y-variate  COLD
  weight     *
  offset     *

  probability distribution is defined via the macros M1, M2, M3 and M4
          scale parameter is 1.000

  terms =  1

        estimate        s.e.     parameter
  1      -1.760      0.1313     1
   scale parameter taken as  1.000

  unit  observed    fitted   residual
     1     31.00     24.09      1.548
     2     17.00     23.91     -1.554

         0.1720

$F +TREA $L %X2 $D ER $CA %EXP(-1.508) : %EXP(-.5936) $

scaled deviance = 0.00000 (change =  -4.872) at cycle  4
          d.f. = 0       (change =  -1    )

   4.441e-16
        estimate        s.e.     parameter
   1      -1.508      0.1585     1
   2     -0.5936      0.2770     TREA(2)
   scale parameter taken as  1.000

  unit  observed    fitted   residual
     1     31.00     31.00     -0.000
     2     17.00     17.00     -0.000

         0.2214
         0.5523
```

Table 2

Results from fitting models (4) and (5) to the data of Table 1.

A lot of information can be gleaned from Table 2, but of most interest are the two last parameter estimates telling that in the placebo group, 22% catch a cold and after that we are told that this proportion is 55% of this figure in the treated group. This is another way of saying that the proportion of the skiers with colds in the treatment group is reduced by 45% compared to the placebo group.

The above demonstrates that the parameters of the binomial model are easily interpreted. After this the logit model is applied in the form

$$\log\left(\frac{\rho_i}{1-\rho_i}\right) = c \tag{6}$$

and with treatment effect allowed

$$\log\left(\frac{\rho_i}{1-\rho_i}\right) = c + \tau_i \tag{7}$$

```
$F $L %X2 $D ER $CA %EXP(-1.571) $

scaled deviance = 4.8717 at cycle  3
          d.f. = 1

   4.810
        estimate        s.e.     parameter
 1        -1.571      0.1586     1
  scale parameter taken as  1.000

 unit  observed    out of    fitted   residual
    1        31       140     24.09      1.548
    2        17       139     23.91     -1.554

        0.2078


$F +TREA $D ER $CA %EXP(-1.257) : %EXP(-.7134) $

scaled deviance = 0.0000000 (change =  -4.872) at cycle  3
          d.f. = 0          (change =  -1    )

        estimate        s.e.     parameter
   1      -1.257      0.2035     1
   2     -0.7134      0.3293     TREA(2)
   scale parameter taken as  1.000

 unit  observed    out of    fitted   residual
    1        31       140     31.00     -0.000
    2        17       139     17.00     -0.000

        0.2845
        0.4900
```

Table 3

Results from GLIM fitting the logit-models (6) and (7).

In this case the two last figures have to be interpreted in terms of odds-ratios, this being 28% for the control group and being reduced to 49% of that by the treatment. The interpretation of this is quite troublesome. Trying to express this in terms of skiers catching a cold, one obtains

$$\rho_{cold} = \frac{\exp(-1.257)}{1 + \exp(-1.257)} = 0.2215$$

for the control group and

$$\rho_{cold} = \frac{\exp(-1.257-0.7134)}{1 + \exp(-1.257-0.7134)} = 0.1223$$

for the ascorbic acid group. This is of course in accordance with the results found above.

This example illustrates that the log-linear binomial model copes with the parameters of interest i.e. the proportion of skiers catching a cold and the change in this proportion due to treatment.

4. Choosing a Model

In Sections 5, 6, and 7, three road safety analyses are presented. They represent the three models treated in this paper; the Poisson model, the logit model and the log-linear binomial model.

Poisson

In many accident studies the Poisson distribution is applied. This is also the case in Section 5 where the trend in Danish fatal road accidents is analysed.

The final model chosen becomes

$$A_i = T_i^{\alpha} \cdot SL73_j \cdot SL79_k \cdot SL85_m \cdot YEAR_i^{\beta} \qquad (8)$$

where A_i is the number of total accidents in year i, and T_i the corresponding car traffic flow. The three factors *SL*73, *SL*79 and *SL*85 correspond to the changes in speed-limits in Denmark and *YEAR* is the year under consideration.

Logit

In the Poisson model mentioned above we are fortunate to have information on the car traffic flow. This is often not the case, and in the situation where we want to evaluate the change in the general speed-limit in built-up areas we have to rely on the logit model.

Our reasoning is as follows. The ideal situation would be to have a case-control study [2, p. 94] with accidents and traffic flow data. Table 4 shows the situation.

		period			
		before		after	
area	urban	A_{ub}	T_{ub}	A_{ua}	T_{ua}
	rural	A_{rb}	T_{rb}	A_{ra}	T_{ra}

Table 4

Case-control study involving accidents and traffic flows.

The hypothesis is that the reduction in the speed-limit in urban (built-up) areas should only affect the number of accidents in urban areas in the after-period. Defining the accident rate as

$$\lambda = \frac{A}{T} \qquad (9)$$

one can formulate the hypothesis of no speed-limit effect as

$$H_0: \frac{\frac{\lambda_{ua}}{\lambda_{ub}}}{\frac{\lambda_{ra}}{\lambda_{rb}}} = 1 \qquad (10)$$

with the alternative

$$H_1: \frac{\frac{\lambda_{ua}}{\lambda_{ub}}}{\frac{\lambda_{ra}}{\lambda_{rb}}} \neq 1 \qquad (11)$$

If we were in the situation where we knew the four traffic flow figures of Table 4, we could test by means of the weighted Poisson model. See, for example, Anderson, [1] and Thomsen and Thyregod, [5, 6].

In the present case we have no information on traffic flow and thus make the assumption that the proportion of kilometres travelled in urban areas is the same before and after the change of law from 60 to 50 KMH.

It is convenient to reformulate H_0 into (12)

$$\frac{\lambda_{ui}}{\lambda_{ri}} = C \cdot \delta_i \qquad H_0: \delta_a = 1 \qquad (12)$$

where we apply a parameterization yielding $\delta_{before} = 1$ recalling that λ is the accident rate we obtain

$$\frac{A_{ui}}{A_{ri}} \cdot \frac{T_{ri}}{T_{ui}} = c \cdot \delta_i \qquad (13)$$

The assumption that the proportion of kilometres travelled in urban areas is the same before and after can be expressed as

$$\frac{T_{rb}}{T_{ub}} = \frac{T_{ra}}{T_{ua}} = k \qquad (14)$$

Combining (13) and (14) yields

$$\frac{A_{ui}}{A_{ri}} = \frac{c}{k} \cdot \delta_i = e \cdot \delta_i \qquad (15)$$

Taking logs leaves us with

$$\log\left(\frac{A_{ui}}{A_{ri}}\right) = \log e + \log \delta_i \qquad (16)$$

which is a logit model.

The hypothesis and model formulation above is based on the simple case with only a time factor (δ_i) included. The example below in Section 6 starts off with the model

$$\frac{A_{ui}}{A_{ri}} = SL79*QUAR*YEAR^{\alpha} \qquad (17)$$

where *SL*79 is a factor designating the change in the accident ratio before and after the speed-limit change on 15 March, 1979. *QUAR* is a four-level factor allowing each quarter of the year to have its own accident ratio. *YEAR* is a continuous variable allowing general trends in the safety situation to take place. Finally, "*" indicates that the model (17) includes a three-factor interaction plus all the lower-order effects as the model is hierarchical.

Our main concern is to study the change in traffic safety due to the 50 KPH-limit introduced on 1 October, 1985. We thus add a new factor *SL*85 to (18) and get

$$\frac{A_{ui}}{A_{ri}} = SL79*QUAR*YEAR^{\alpha} \cdot SL85 \qquad (18)$$

The change in the goodness-of-fit statistic is thus a test of the effect of the speed-limit.

Log-linear Binomial

In the logit model we used the accidents in rural areas as a control group adjusting for general trends in the number of accidents. We now move to another design leading us to the log-linear binomial model.

In Denmark, collisions between bus passengers and bicyclists is a well known problem. Often bus stops are located at the kerb of the bicycle path and the bus passengers thus have to cross the bicycle path. In certain situations the bicycle riders have to give way, but sometimes do not thus leading to an accident. To avoid these accidents one measure has been to place rumble lines on the bicycle path. The idea is to slow down the bicycle riders and make them stop when a bus is present. All this leads to the model

$$\log\left(\rho_{stop}\right) = c + \tau_i + \alpha \,.\, \log\left(x_j\right) \qquad (19)$$

where ρ_{stop} is the proportion of bicycle riders stopping, c is a general level, τ_i is the before/after factor of main interest, α is a parameter to be estimated, and x_j is the number of bus passengers crossing the path.

We are by (19) back among the skiers in Section 3. Our factor τ_i is then easily interpretable as the change of the proportion of stopping bike riders. A logit model will not offer this ease in interpretation.

5. The General Road Safety Trend – The Poisson Model

Table 5 shows for the years 1972 to 1986 the number of fatal road accidents and car traffic flow index as provided by the national authorities. The accident rate (defined as the number of accidents divided by the car-flow) is also shown.

year	accidents	flow	accident-rate
1972	1040	100.0	10.4
1973	1003	104.1	9.6
1974	724	100.1	7.2
1975	779	104.4	7.5
1976	789	110.1	7.2
1977	784	113.6	6.9
1978	787	115.4	6.8
1979	667	113.3	5.9
1980	625	107.3	5.8
1981	610	104.7	5.8
1982	606	105.8	5.7
1983	615	109.4	5.6
1984	616	115.4	5.3
1985	697	120.7	5.8
1986	655	126.4	5.2

Table 5

The number of fatal accidents, car-flow, and the accident rate for the period 1972-1986.

Plotting the accidents as a function of the years (Figure 1) and the accident rate as a function of the years (Figure 2) give interesting patterns.

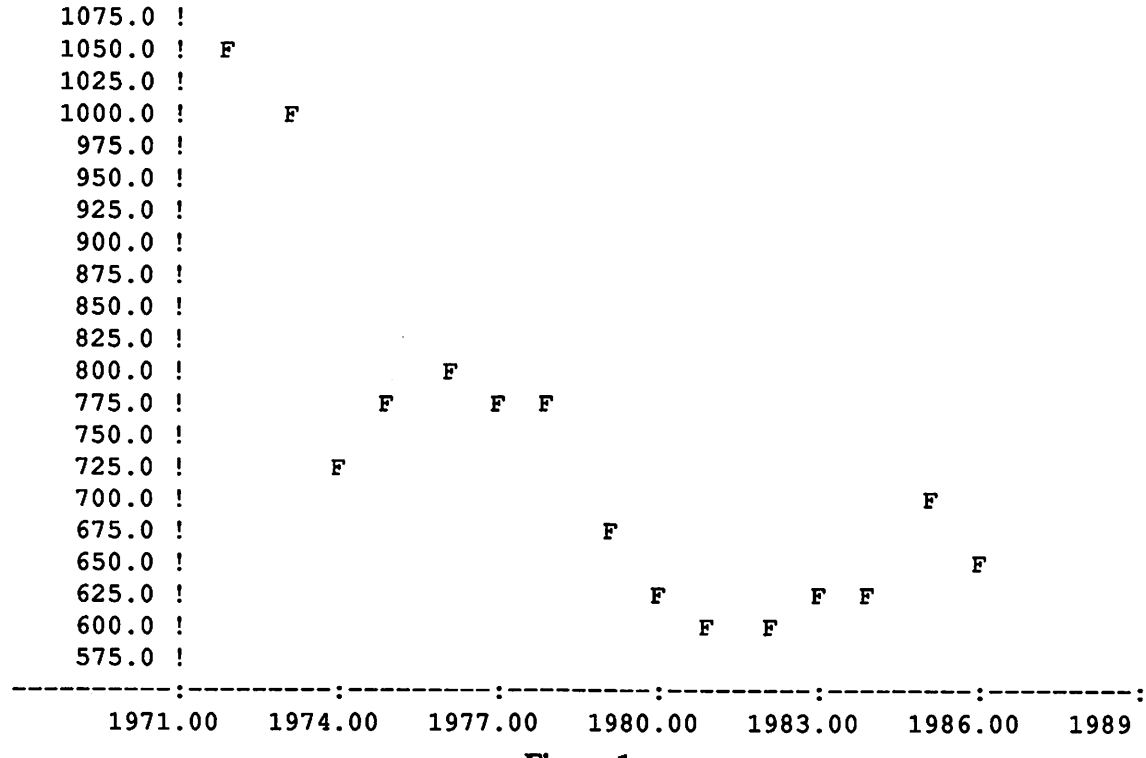


Figure 1

Fatal accidents as a function of the years.

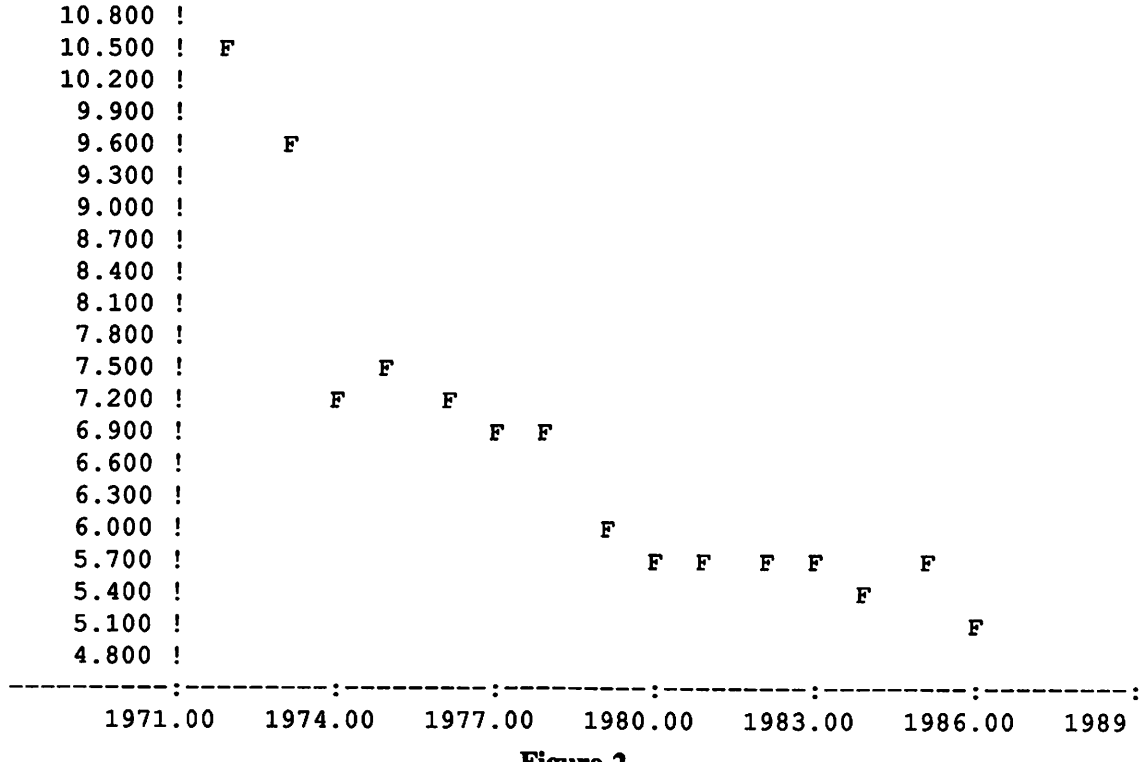


Figure 2

Fatal accident-rate as a function of the years.

Figure 2 in particular reflects the use of general speed limits: the introduction late in 1973, the lowering in rural areas in early 1979, and finally the lowering in late 1985 in urban areas. The model (8) reflects this.

The result of fitting (8) with $\alpha = 1$ is shown as Table 6.

year	observed values	fitted values
1972	1040	1035.8
1973	1003	1007.8
1974	724	745.7
1975	779	756.0
1976	789	780.0
1977	784	790.5
1978	787	790.9
1979	667	666.8
1980	625	624.2
1981	610	602.8
1982	606	603.5
1983	615	618.7
1984	616	647.5
1985	697	672.3
1986	655	655.0

Table 6

Observed and fitted fatal accidents, where fitted values are estimated by the model (8).

The fit of the model seems good. The corresponding likelihood-ratio test statistic has a value of 4.11 being equal to $\chi^2(10)_{0.06}$ thus confirming a good description of the fatal accidents.

The parameter estimates of (8) are shown in Table 7.

Parameter	Estimate and 95% confidence limits
general level	10.3605
SL73 (60/90/110)	0.8884 0.8012 | 0.7226
SL79 (60/80/100)	0.9353 0.8701 | 0.8095
SL85 (50/80/100)	1.0200 0.9366 | 0.8600
β	–0.0267 –0.0984 | –0.1700

Table 7

Parameter estimates from the model designated (8).

After the names of the speed-limit parameters, the limits in urban areas, rural areas and motorways are shown in parentheses. The introduction of 60/90/100 KPH reduced the accident rate (accidents adjusted for changes in flow) by 1–0.8012 = 0.20 i.e. 20%. Together with these three safety improvements we see a general trend estimated by β and during the 15 years included accounts for 23% decrease in the accident rate.

In terms of evaluating the 50 KPH limit the analysis above is quite crude and more detailed work can be found in Section 6.

As stated in Section 4 the analysis of Section 6 is based on the same trend in traffic flow inside urban areas and rural areas. Can we check this assumption?

To some extent, yes. The index of the car traffic flow is mainly counted on rural roads and few urban roads always carrying through going traffic, so the index could be claimed to be a 'rural index'.

Analysis following the lines of the Poisson model above, but using fatalities instead leads to the conclusion that no general time trend is present in the rural fatality rate (i.e. the number of fatalities per flow unit).

6. Evaluation of the 50 KPH Limit – the Logit Model

Table 8 gives the basic structure of the personal injury accidents in the following study.

The first model fitted is (17) allowing interaction between *YEAR*, the speed limit of 1979, and the quarter considered. This model has a likelihood-ratio test value of 44.4 with 28 degrees of freedom thus corresponding to the 0.9746 fractile of the χ^2 distribution. The plot of the Pearsonian residuals is given as Figure 3.

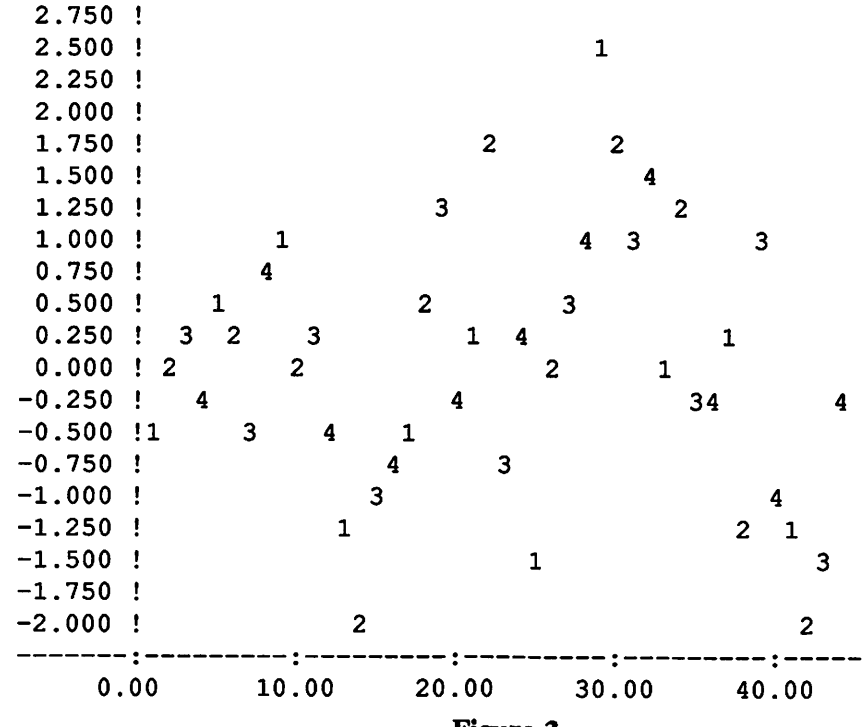


Figure 3

Pearsonian residuals as a function of the number of the quarter.

	Quarter			
Year	January	April	July	October
Rate	2.231	2.232	2.031	2.425
1976 Urban	2233	2864	2979	2913
All	3234	4147	4446	4114
Rate	2.050	2.134	1.965	2.243
1977 Urban	1954	2834	3063	2940
All	2907	4162	4622	4251
Rate	1.954	2.046	1.999	2.015
1978 Urban	2040	2809	2898	2734
All	3084	4182	4348	4091
Rate	1.691	2.184	2.200	2.552
1979 Urban	1512	2564	2693	2447
All	2406	3738	3917	3406
Rate	2.379	2.286	2.294	2.433
1980 Urban	1525	2414	2443	2255
All	2166	3470	3508	3182
Rate	2.278	2.344	2.045	2.346
1981 Urban	1490	2178	2155	1961
All	2144	3107	3209	2797
Rate	1.985	2.145	2.076	2.323
1982 Urban	1229	2061	2184	2160
All	1848	3022	3236	3090
Rate	2.278	2.239	2.063	2.273
1983 Urban	1476	2058	2131	1975
All	2124	2977	3164	2844
Rate	1.927	2.159	1.935	2.053
1984 Urban	1447	2066	2158	2014
All	2198	3023	3273	2995
Rate	1.861	1.929	1.980	1.927
1985 Urban	1323	1983	2315	1925
All	2034	3011	3484	2924
Rate	1.692	1.843	1.768	1.933
1986 Urban	1403	1872	1957	1950
All	2232	2888	3064	2959

Table 8

For the years 1976 to 1986 the ratio between accidents in built-up areas and the accidents in rural areas is given together with the number of accidents in built-up areas and the total number of accidents. The columns represent the four quarters of each of the eleven years considered.

The five last quarters all having the 50 KPH limit show negative residuals suggesting a 50 KPH factor (*SL85*) to be included. Doing this changes the likelihood-ratio to 30.6 with 27 degrees of freedom corresponding to the 0.7114 fractile. Dropping the three-factor interaction yields a likelihood-ratio of 33.2 and a fractile of 0.6847 thus improving the fit. The exclusion of effects from this model does not improve the fit. Table 9 gives the log parameter estimates and Figure 4 shows the plot of the residuals.

	estimate	s.e.	parameter
1	0.8325	0.03260	1
2	0.4628	0.07854	SL79(2)
3	-0.04224	0.04154	QUAR(2)
4	-0.1155	0.04079	QUAR(3)
5	0.05011	0.04187	QUAR(4)
6	-0.1910	0.03380	YEAR
7	-0.08845	0.02378	SL85(2)
8	-0.2124	0.06956	SL79(2).QUAR(2)
9	-0.2031	0.06855	SL79(2).QUAR(3)
10	-0.07879	0.07021	SL79(2).QUAR(4)
11	-0.08641	0.03428	SL79(2).YEAR
12	0.1383	0.04494	QUAR(2).YEAR
13	0.1487	0.04421	QUAR(3).YEAR
14	0.04939	0.04539	QUAR(4).YEAR

scale parameter taken as 1.000

Table 9

Parameter estimates based on the model (17) without the three-factor interaction, but with the 50 KPH effect *SL85* added.

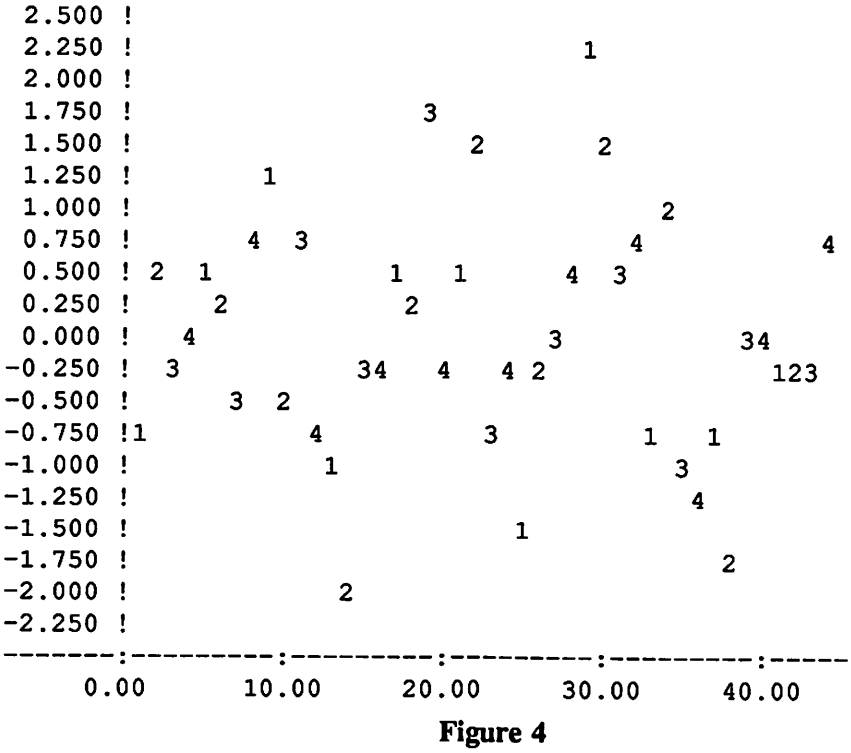


Figure 4

Plot of the residuals from the model described by Table 9.

The plot of residuals (Figure 4) seems quite satisfactory with no alarming outliers. Unit 7 in Table 9 yields the estimate of the change in accidents. Calculating exp(–0.08845) = 0.9153 leading to a 8.5% decrease in the urban accident rate as consequence of our model. The corresponding 95% confidence limits are 4.2% and 12.6%.

7. Rumble Lines at Bus Stops – the Log-linear Binomial Model

We now leave the accidents, and turn to behaviour that sometimes leads to accidents.

As stated in Section 4 and model (19) our main concern is the proportion of bicycle riders stopping when a bus is present. Table 10 gives an idea of what the data look like.

STOP	*N*	*PEDE*	*TIME*
0	3	7	1
0	2	3	1
0	2	5	1
0	3	4	1
0	1	6	1
0	2	1	1
0	1	4	1
0	2	10	1
0	1	3	1
0	6	2	1
0	1	3	1
0	5	4	1
0	1	4	1
0	2	1	1
0	2	0	1
1	5	9	1
.	.	.	.
.	.	.	.
.	.	.	.

Table 10

The four columns give the number of stopping bicycle riders (*STOP*), the total number of bike riders in the sequence (*N*), the number of bus passengers and pedestrians crossing the path (*PEDE*), and a time-period factor indicating before (=1) and after (=2) (*TIME*).

The intention was to analyse the data by Genstat, but the macro used (GLMODEL) could not include interactions. This should be possible when Genstat 5 is ready. GLIM can include interactions, and some of the results are shown as Table 11.

```
Model 1
scaled deviance =  89.511 at cycle 2
          d.f. = 111

   197.6
        estimate        s.e.     parameter
 1      -2.807       0.1714     1
 scale parameter taken as  1.000

Model 2
scaled deviance =  64.777 (change =  -24.73) at cycle  5
          d.f. = 110     (change =   -1    )

   144.1
        estimate        s.e.     parameter
 1      -6.986       0.8483     1
 2       2.109       0.3681     PEDE
 scale parameter taken as  1.000

Model 3
scaled deviance =  64.139 (change = -0.6381) at cycle  3
          d.f. = 109     (change = -1      )

   140.6
        estimate        s.e.     parameter
 1      -7.001       0.8620     1
 2       2.160       0.3886     PEDE
 3     -0.1686       0.3439     TIME(2)
 scale parameter taken as  1.000

Model 4
scaled deviance =  62.464 (change = -1.675) at cycle  6
          d.f. = 108     (change = -1     )

   123.0
        estimate        s.e.     parameter
 1      -5.528       1.341      1
 2       1.449       0.6537     PEDE
 3      -2.322       1.768      TIME(2)
 4      0.9889       0.8020     PEDE.TIME(2)
 scale parameter taken as  1.000

Model 5
scaled deviance =  62.465 at cycle 2
          d.f. = 108

        estimate        s.e.     parameter
 1      -5.528       1.341      1
 2      -2.322       1.768      TIME(2)
 3       1.449       0.6537     TIME(1).PEDE
 4       2.438       0.4645     TIME(2).PEDE
 scale parameter taken as  1.000
```

Table 11

Output from GLIM fitting five models to the data shown (in part) in Table 10.

The first model of Table 11 is the result of having a constant proportion of bicycle riders stopping, not taking into account period (before/after) and the number of pedestrians.

The second model is of more interest as it includes the number of bus passengers and pedestrians crossing the bicycle path. The estimate becomes 2.109 saying that the number of pedestrians squared explains the proportion of bike riders stopping. It is seen that this effect is quite important having a likelihood-ratio test-statistic of 24.73 corresponding to $\chi^2(1)_{0.999999}$.

In the third model (19) we here include the time-period-factor *TIME*. The value exp(–0.1686) = 0.8448 tells that the proportion of stopping bicycle-riders is (non-significantly) 16% lower (!) in the after-period. We shall not go into details here, but the fact that the speeds decreased from before to after might make it unnecessary to stop as merging is then possible. The last two models including interaction are only included for the sake of completeness as the significance probability becomes 0.8044.

8. Concluding Remarks

This paper presents three examples illustrating the use of the log-linear Poisson model, the logit model, and the log-linear binomial model.

The examples show that the choice of model shall ensure parameters that are intuitively interpretable.

To apply models without thought directed towards the communication and application of the results can easily be a waste of time.

9. Bibliography

[1] Anderson, E.B.
Multiplicative Poisson Models with Unequal Cell Rates.
Scandinavian Journal of Statistics, **4**, pp. 153-158, 1977.

[2] Breslow, N.E and Day, N.E.
The Analysis of Case-control Studies.
Statistical Methods in Cancer Research, **1**,
International Agency for Research on Cancer, Lyon, 1980.

[3] Fienberg, S.E.
The Analysis of Cross-Classified Categorical Data.
MIT Press, London, (2nd Edition) 1980.

[4] *The GLIM System Release 3.77 Manual.* Payne, C.D. (ed).
NAG Ltd, Oxford, 1986.

[5] Thomsen, L.K. and Thyregod, P.
Unweighted and Weighted Poisson-Models for Discrete Data.
Symposium i Anvendt Statistik (*Symposium in Applied Statistics*) Höskuldsson *et al* (eds).
NEUCC, Lyngby, 1981.

[6] Thomsen, L.K. and Thyregod, P.
Unweighted and Weighted Poisson-Models for Discrete Data.
Symposium i Anvendt Statistik (*Symposium in Applied Statistics*) Edwards *et al* (eds).
RECKU, University of Copenhagen, 1983.

[7] Thyregod, P. and Spliid, H.
On the Analysis of Factorial Arrangements with a Dominating Binary Response.
IMSOR, The Technical University of Denmark, Research Report 18, Lyngby, 1985.

10. Appendix

All analyses presented have been carried out by means of either GLIM or Genstat. The use of the log-linear binomial model necessitates use of macros M1, M2, M3 and M4 in GLIM, and macros GLMODEL, LINK, WEIGHT, and INVLINK in Genstat.

GLIM

```
$MAC M1 $CAL %FV=N*%EXP(%LP) $ENDMAC
$MAC M2 $CAL %DR=1/%FV $ENDMAC
$MAC M3 $CAL %VA=%FV*(1-%FV/N) $ENDMAC
$MAC M4 $CAL %DI=2*(%YV*%LOG(%YV*(N-%FV)/(%FV*(N-%YV)))
  +N*%LOG((N-%YV)/(N-%FV))) $ENDMAC
$OWN M1 M2 M3 M4
$SC 1
$CAL %LP=%LOG(PROBFIT/N)
$FIT ROW+COL$DI ER$
```

Genstat

```
'REFER/NUNN=1000,NID=1000' BINREG
'GET/FILE=1' GLMODEL$GLMODEL
'INPUT' 2
'UNIT' SEKVENS $ 112
'VARIATE' STOPPING,N,NUMPED
'READ' STOPPING,N,NUMPED,PERIODE
'INPUT' 1
'RUN'
'CALC' INTERACT=(NUMPED+0.1)**PERIODE
'CALC' INTERACT=LOG(INTERACT+0.1)
'CALC' NUMPED=LOG(NUMPED+.1)
'PRINT/P' STOPPING,N,NUMPED,PERIODE $ 10.0,10.0,10.0,10.0
'VARIATE' LIM=0.5
'FACTOR' PERIODEN $ 2
'GROUPS' PERIODEN=LIMITS(PERIODE;LIM)
'RUN'
'CALC' PHAT=STOPPING/(N)
'CALC' STOPPING=STOPPING+.000
'PRINT/P' STOPPING,N,NUMPED,PHAT $ 14.0,15.1,10.0,15.3
'RUN'
'TERMS/PRINT=SCG,TOTAL=N' STOPPING+NUMPED+PERIODEN+INTERACT
'MACRO' LINK $
'CALC' LINPRED=LOG(FVAL/N)
'ENDMACRO'
'MACRO' WEIGHT $
'CALC' W=1.0/FVAL
'ENDMACRO'
'MACRO' INVLINK $
'CALC' FVAL=N*EXP(LINPRED)
'ENDMACRO'
'SET' ERROR=BINOMIAL
'PRINT' LINK,WEIGHT,INVLINK
'SET' DEVPRN=YES
    : Y=STOPPING
    : MODEL=NUMPED,PERIODEN
'RUN'
'USE' GLMODEL $
'RUN'
```

```
'SET' MODEL=NUMPED
'RUN'
'USE' GLIMODEL $
'RUN'
'SET' MODEL=NUMPED,PERIODEN,INTERACT
'RUN'
'USE' GLMODEL $
'RUN'
'TERMS/PRINT=SCG,TOTAL=N' STOPPING+NUMPED*PERIODEN
'Y/ERROR=BINOMIAL,LINK=LOGIT' STOPPING
'FIT/PRINT=ACF,INT=N' NUMPED
'ADD/PRINT=ACF,INT=Y'
    : PERIODEN
    : NUMPED.PERIODEN ; FVAL=FITTED
'HEADING' H=''LP''
'GRAPH' FITTED,STOPPING ; NUMPED $ H
'RUN'
'DROP' NUMPED*PERIODEN
'GRAPH' PHAT ; NUMPED
'RUN'
'CLOSE'
'STOP'
```

Genstat for Flexible Summaries of Data or... When is a Genstat User not a Genstat User?

P J Colman
Pfizer Central Research
Sandwich
Kent
United Kingdom *CT13 9NJ*

1. Background

Pfizer Central Research (in Sandwich, UK) conducts research into veterinary medicinal products as well as human medicinals. Clinical trials in animals typically result in high-quality data (no problems with patient non-appearance!) for a number of treatment groups. The types of data encountered are as varied as the research areas involved; faecal egg counts (from parasitology trials) and temperatures (from antibacterial studies) being just two examples. Weight and (to a lesser degree) feed consumption are frequently measured, as changes in them can be sensitive indicators of the animal's general welfare. A common feature of the trials is the repeated measures nature of much of the data; for example, weights may be taken weekly over a six-month grazing season or temperatures may be taken daily over a two-week treatment period. A natural requirement of the investigator running the trial is to see his data summarised as it becomes available and, from time to time, to present the up-to-date situation to his management. To fulfill these repetitive requests used to result in the statisticians having sufficent time to provide only basic analyses of data from completed trials. It was in order to try and improve this situation that this Genstat program was created.

2. Designing the Program

The basic rules of system design are as follows:

(i) analyse the needs of the system user;

(ii) construct a structured specification of the system which is expected to meet the users' needs;

(iii) walk through the structured specification with the users to ensure suitability;

(iv) implement the system.

There is, of course, the opportunity for iteration between the first three steps so that the specification that is eventually implemented is definitely that which will meet the users' needs.

Unfortunately, I am unable to report that this is the strategy that I followed. My approach was more that of the ongoing prototype! However, I did endeavour to analyse the actions required within the program to produce the output that I believed to be what was wanted.

3. Designing the Program – Needs

The driving force of program design was the output requirements; it clearly makes no sense to produce a program that does not meet the needs for which it was written. The main output requirements were as follows:

(i) protocol summary, including:

- key dates (e.g. the start and end of the treatment period),
- people (investigator, statistician),
- location,
- objective (of experiment or trial),
- treatments (including description of treatment structure);

(ii) raw data listings by treatment group;

(iii) summaries by treatment group (which must be flexible, to allow specification by investigator);

(iv) flat-file of data in a format suitable for local plotting software.

Secondary requirements for output were as follows:

(i) detect and mark abnormal values;

(ii) standardise units (e.g. weight may have been recorded in pounds or kilograms);

(iii) neat format to enable direct incorporation of the output into the investigator's trial report;

(iv) sensible English-language descriptions of summaries and sensible numerical formats for data and summary output;

(v) ability to produce just plots, just tables or both.

The data input required to fulfill these needs is then:

(i) control data (scalars);

(ii) protocol data (read in as a header and printed straight out again);

(iii) dates of observations and corresponding days (since there is no date function in Genstat 4.04);

(iv) raw data to be listed and summarised.

4. Designing the Program – Structure

A somewhat simplified version of the structure of the program follows.

Read control information (scalars), protocol information (header), dates and day numbers (variates);
Calculate the number of days for which data are available using the NVAL function.

Read raw data (one variate for each day/date);
Calculate number of animals.

Set number of units;
Standardise missing values as * (other codes acceptable prior to this step);
Standardise units of measurement;
Calculate and tabulate variates to achieve desired summaries and changes from previous value;
If plots required then output plot data to file;
If tables not required then exit.

Set up headings for tables and descriptions of summary statistics.

For each day:
    For each variable to be summarised and for the change from previous value:
        Calculate the pooled estimate of the standard deviation;
        Flag value as + or – if the value is more than 2 s.d. away from the treatment group mean. (The value 2 is easily changed if desired.)
    Next variable;
    Join this day's data and bits of table to the existing list;
Next day.

```
Print protocol details etc;
For each treatment group:
    For each page of output for the treatment group:
        Calculate page number;
        Calculate number of days' data on the page;
        Set print formats accordingly;
        Print table header;
        Restrict to treatment group;
        Print raw data and flags;
        Derestrict data;
        For each summary statistic:
            Print summary statistic description and values;
        Next summary statistic;
    Next Page;
Next Treatment Group.
Finish.
```

5. Using the Program

The summary program is usually accessed through the Animal Health Database; the user is generally unaware of the software he is using (hence the alternative title for this talk). The user (who may be an investigator or a statistician) enters the database system and, having selected the trial on which to work, he requests the report options available for that trial. Let us assume that one of the report options available to him is this summary program; in fact several versions may be available – one for each of a number of different data-types. By selecting a summary report, a database query language program is invoked which extracts the data required and then sets up a subprocess (using the Vax VMS facility to spawn a subprocess) to run the Genstat job. Finally, the user may choose to see the report at the terminal, to file it for future reference or to print it on the printer of his choice. When the printer selected is a laser printer (such as the DEC LN01 or LN03), we find that the quality of output is sufficient to allow direct incorporation of the summary table into trial reports and other documents. Alternatively, the file may be incorporated into a word-processing package.

An example of the output of the program is given in Table 1.

The example concerns animal weights which were recorded throughout a parasitology trial. The first output consists of the protocol information which usually occupies the first page of the summary. This is useful for checking and identifying the summary. In the body of the table (on the next page), note the flagged extreme values; recall that these may be due to an abnormally high or low value for a particular time or they may refer to an abnormally high or low change from the previous time. In this example, we produce simple summary statistics only, but other, more complex calculations have been performed with other variables.

6. Future Developments

As has already been hinted, there are several different versions of the program at large which results in the process of bug-correction and improvement being extremely difficult. The next major step (apart perhaps from conversion to Genstat 5!) will be to put the variable- and trial-specific information into macros. This would then allow there to be just one copy of the driving Genstat program, while the provision of the specific macros would still allow flexibility in individual situations.

B O D Y W E I G H T S to day 168 — Page : 1

Trial type :Cattle parasitology trials
Trial number :5231E-03-85-002
Location code : - Not specified
1st Investigator :022 - Dr I.A.M. Investigator
2nd Investigator :024 - Dr M.E. Too
Statistician :051 - Mr I.M. Numerate
Projected start :01-APR-85
Projected end :01-OCT-85

To assess the efficacy of the treatment administered at spring turnout in the control of gastrointestinal parasites throughout the grazing season.

Treatment Details

TREATMENT CODE	COMPOUND	DOSE	PRINC/TRACER
T1	CONTROL	---	PRINCIPAL
T2	WONDER_DRUG	1 only	PRINCIPAL

B O D Y W E I G H T S to day 168 — Page : 2

Trial Number : 5231E-03-85-002
Treatment Group : T1
Units : Kg

Animal Code / Date Day	16MAY85 0	30MAY85 14	13JUN85 28	27JUN85 42	11JUL85 56	25JUL85 70	08AUG85 84	22AUG85 98	05SEP85 112	19SEP85 126
806	162	161	182	190	200	204	214	204 -	193	*
809	149	148	164	170	187	196	200	204	192	187
814	143	143	152	162	171	173	170	167	173 +	176
819	158	151	165	183	189	198	198	204	192	190
820	145	145	149	161	168	170	181	182	167	162
824	155	165	172	177	170 -	175	181	179	177	170
826	176	171	183	193	205	200 -	199	200	198	*
830	164	156	170	185	192	198	198	202	200	195
832	164	155	165	174	182	185	190	184	163 -	165
834	168	143	166	180	193	199	205	206	194	*
836	160	160	174	184	201	198	206	207	210	*
837	147	133	147	155	166	168	161 --	167	162	*
849	155	166	177	190	202	204	212	211	188 -	187
853	162	160	170	184	194	202	202	205	205	195
855	180	164	184	195	193 -	198	196	197	195	183
857	165	153	170	182	195	195	201	195	192	181
861	174	178	183	205 +	209	213	217	216	202	198
863	168	158	175	181	195	197	199	190 -	189	182
864	166	157	176	185	197	204	210	209	190	185
869	195 +	172	162 -	181	193	196	200	206	203	189
Total Weight	3256.0	3139.0	3386.0	3617.0	3802.0	3873.0	3940.0	3935.0	3785.0	2745.0
Number of Animals	20	20	20	20	20	20	20	20	20	15
Mean Weight	162.8	156.9	169.3	180.8	190.1	193.7	197.0	196.8	189.3	183.0
Standard Deviation	12.6	11.1	10.9	12.0	12.5	12.6	14.3	14.1	13.9	10.8
Mean Weight Gain	0.0	-5.9	12.4	11.6	9.3	3.5	3.4	-0.3	-7.5	-5.5
Mean Cumulative Weight Gain	0.0	-5.9	6.5	18.0	27.3	30.8	34.2	34.0	26.5	20.1

Table 1

Following on from this development, it should then be possible to set up default summaries for specific data-types. This would allow users to summarise their data in a standard manner, without having to get further macros set up for each new trial.

Large data sets currently cause problems which depend on the nature of the largeness:

(i) if we have many time points, performance suffers quite markedly because of the single-day handling of each time;

(ii) if we have many treatment groups (no matter what the size of each group), performance suffers because of the looping through treatment groups;

(iii) if we have many (i.e. greater than about 20) animals in a treatment group, the output format currently spills over onto the next page in a controlled but untidy manner.

7. Conclusion

A flexible structure for a Genstat program has been developed, to cope with the common requirement of summarising repeated measures data. This allows the statistician to spend more time on design and analysis of trials to the benefit of all. Whilst some elementary form of systems analysis took place before production of the Genstat code, the author intends to undertake a more thorough analysis before embarking on the next stages of development.

Cumulative Count Data

P Brain and R Butler
Department of Agricultural Sciences
University of Bristol
AFRC Institute of Arable Crops Research
Long Ashton Research Station
Bristol
United Kingdom *BS18 9AF*

1. Introduction

Suppose a process is observed and cumulative counts are recorded at various times after the start of the experiment. Typical examples of this include studies on cumulative mortality of insects after innoculation with a virus [5], cumulative counts of insects caught in traps [2] and data from germination tests [6]. In all cases the obvious way to present the data is as a plot of cumulative counts versus time. This has led to a variety of inappropriate analyses in the literature. Glen and Brain [2] used a more appropriate method for analysing cumulative trap catch data. Hunter, Glasbey and Naylor [3] presented this method formally, but they wrote their own program in Fortran. In this article the theory of their method is outlined and a procedure which analyses this type of data using Genstat 5 is presented. Possible extensions to the original method are also discussed.

2. Analysis Using Maximum Likelihood

Cumulative count data are not amenable to ordinary least-squares methods for two reasons. Firstly, there is a serial correlation between the value at a given time and the values at all previous times, and secondly the data are in the form of counts which are not Normally distributed. In the literature both these points have largely been ignored and typical analyses have included fitting the logit transformation of percentage count against time [1], and nonlinear regression of cumulative count against time directly [4]. Other previous methods are noted by Hunter, Glasbey and Naylor.

However, if we look more closely at what is being observed in an investigation of this kind we note that we are recording how many individuals from a population respond in a given time interval, where the response may be, for example, seed germination, insect death or number of insects trapped. In the first two examples we are crudely measuring an underlying continuous variable, the time an individual from the population takes to respond; in the third, we are measuring a composite time variable which is made up of the time until the insect hatches and the subsequent time until it is trapped. In the first two examples we also know the size of the population being sampled whereas in the third we do not. However in all three cases the underlying variate is the time to an event. The plot of cumulative count against time is thus proportional to an underlying cumulative distribution function corresponding to the time to an event. The method of analysis presented here is a more general form of that presented by Hunter, Glasbey and Naylor.

2.1. Maximum-likelihood Analysis for a Single Subpopulation

We define the probability distribution function of the time to the event for an individual that is capable of responding by $g(t)$, and the proportion of individuals that are capable of responding by p (in the three cases mentioned above, the individuals not capable of responding are the proportion of seeds that are non-viable, the proportion of insects that are either resistant or not infected, and the insects that will not be trapped during their lifetime). We denote the number that respond in (t_{i-1},t_i) by n_i, for $i = 1...k$, where t_0 is the start of the experiment and the final measurement is made at time t_k. All individuals that respond after this time are thus not recorded. Let us denote the cumulative distribution function by $G(t)$. Then the set of counts

$(n_1,n_2...n_k)$ are multinomially distributed with probabilities $\frac{G(t_i)-G(t_{i-1})}{G(t_k)}$ $(i = 1...k)$ for the k cells. (Note that we are dealing with a censored population so that the divisor is required to ensure that the probabilities sum to unity.)

If we know the size of the population from which we are sampling (N) then the maximum-likelihood estimate of p, the proportion capable of responding, is

$$\hat{p} = \frac{\sum_i n_i}{NG(t_k)}$$

This type of estimation problem is readily solved by Genstat 5 using a combination of the MODEL directive with a multinomial distribution, and the FITNONLINEAR directive with the relevant calculation to define the cumulative distribution function, $G(t)$. Note that $G(t)$ will contain parameters that need to be estimated. In the example presented (Table 2, Hunter, Glasbey and Naylor) an appropriate formulation for $G(t)$ is

$$G(t) = \text{NORMAL}\{b(\ln(t-lag)-m)\}$$

where *lag* is the lag time, m is the mean of $\ln(T-lag)$, and $1/b$ is the variance of $\ln(T-lag)$ (T is the time to germination). This is a three-parameter model and estimates can be made of all three parameters using FITNONLINEAR.

2.2. Maximum-likelihood Analysis for Several Subpopulations

The population under investigation may be made up of subpopulations which behave differently. The response time for individuals in the different subpopulations will have different cumulative distribution functions (CDFs), so that if we have s populations we have a set of CDFs, $G_1(t),G_2(t)...G_s(t)$.

Each of these will have some different parameters. Again we have a proportion of the population that are not capable of responding. We denote the proportion of the responding population in subpopulation i by p_i, so that $p_1 + ... + p_s = 1$.

Then the joint distribution of the n_i is still multinomial, but the i^{th} cell probabilities is now of the form

$$\sum_j \left[p_j \frac{G_j(t_i)-G_j(t_{i-1})}{G(t_k)} \right]$$

Again, provided the number of parameters is less than the number allowed by Genstat it is possible to fit this model for a given set of CDFs $G_i(t)$, estimating both the proportions p_i and the parameters implicit in the formula for the CDF.

2.3. Practical Problems: Detecting Lack of Fit and Multiple Populations

With this method there is a temptation to use the whole set of measurements, as we would do if we were fitting a logistic curve to the growth of a plant with time. However, the time intervals in which very few events are predicted may need pooling to ensure that they contain a reasonable number of predicted events.

In practice it may be difficult to fit the model with multiple subpopulations, and the choice of initial estimates will generally be crucial.

As we are fitting the model using the multinomial distribution and maximum-likelihood estimation we can use the deviance directly to detect whether there is significant lack of fit. Lack of fit can be caused by an inappropriate model for $G(t)$ (in which case a plot of the predicted CDF and the actual CDF against time may suggest an improved formula), or by multiple subpopulations. (Alternatively it may be caused by bad data!) If multiple subpopulations are suspected it may be possible in the first instant to assume that there are two

populations and refit; the deviance may then be non-significant. Extra subpopulations may be added sequentially and the decrease in deviance tested against the residual mean deviance using an approximate F-test if the residual deviance is significant, or a χ^2-test if it is not.

3. A Genstat Procedure to Analyse Cumulative Count Data

A general procedure has been developed to deal with this type of data using the statistical theory outlined above. The procedure fits models with any combination of the following attributes:

a choice of link functions;

a choice of distribution functions;

single or multiple subpopulations of individuals capable of responding.

The attributes are discussed in more detail below.

3.1. Link Functions

The procedure uses time as its explanatory variate. In practice transformations of time have been used to 'Normalise' the time to response in some way. For example, Weaver, Tan and Brain [6] used $Z = \ln(T-lag)$. We define Z to be the transformed time, i.e. in general $Z = f(t;a)$, where a is an unknown parameter such as *lag*. For ease of use and flexibility the procedure uses this type of transformation as a link function; the following links are available:

Identity:	$z = t$
Log:	$z = \ln(t)$
Shifted log:	$z = \ln(t-lag)$, where *lag* is the lag minus the time
Own:	$z = f(t;\ ownp)$, where *ownp* is an unknown parameter and the user is allowed to define his own link

For the case where a lag is present it is estimated by the procedure.

3.2. Distribution Functions

A choice of distributions of Z (the transformed time, defined above) is allowed; they are as follows:

Normal:	CDF = NORMAL($b(z-m)$)
Complementary log-log:	CDF = exp(–exp($b(z-m)$)
Logistic:	CDF = 1/(1+exp(–$b(z-m)$)

Note that for all the above distributions the formulation is in terms of inverse standard error; this parameterization is used to enable comparability with the usual growth curves to which this process is analogous.

3.3. Number of Subpopulations

As noted above it is theoretically possible to subdivide a population of individuals into subpopulations. If the number of subpopulations is given the procedure attempts to fit a mixture of distributions. The distributions of each subpopulation are assumed to be of the same family, but with different values of b and m. If a lag is used in the link function it is assumed to be the same for all subpopulations.

3.4. Output

The procedure gives the 'standard' range of output options allowed by FITNONLINEAR, but also includes an option to produce a graph of the cumulative counts and the cumulative fitted curve. This is included to allow a visual inspection of how well the model describes a given set of data.

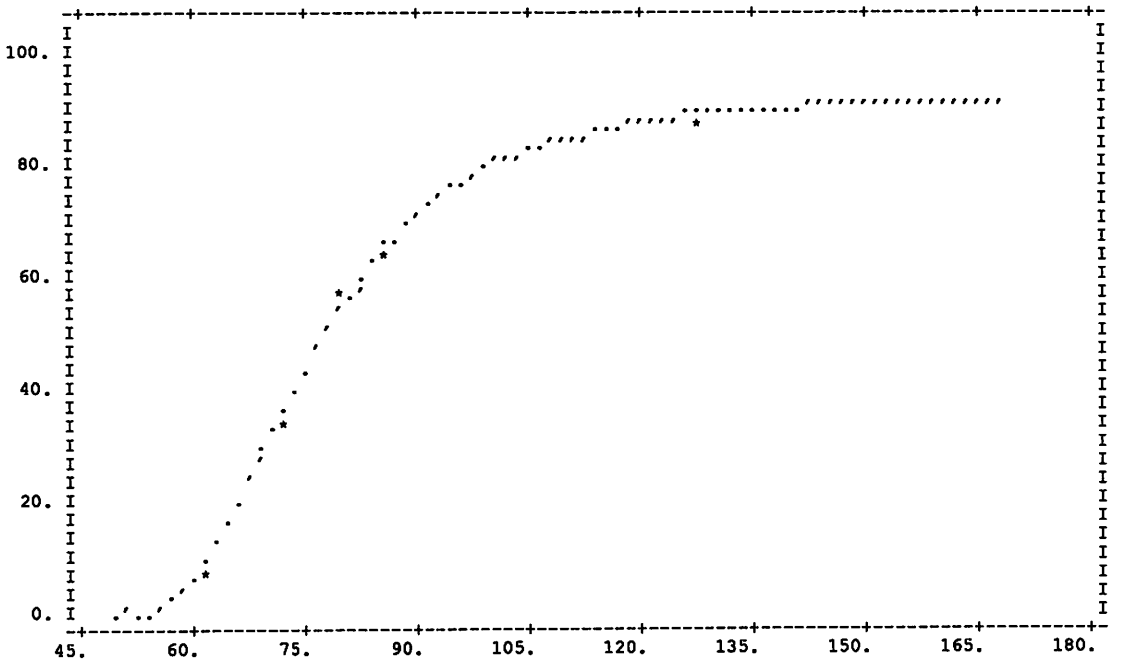
4. Example

In their paper, Hunter, Glasbey and Naylor [3] present a 'typical' data set, and this is analysed using the procedure to illustrate its use and output (for further details of the data see the paper). They assume that the link is ln(*time*–48); in our example we assume that the lag value of 48 is known. We then re-analyse estimating the lag as one of the parameters. Finally, we compare the results against that obtained by the inappropriate probit analysis, with the count being the number that germinated by the last measurement time.

```
  1  JOB 'TESTING HUNTER, GLASBEY AND NAYLOR DATA'
  2
  3  "
 -4  Trial analysis of data presented in Table 2 from Hunter, Glasbey and
 -5  Naylor. They derived the Mean and Variance, rather than the Mean and
 -6  B-parameter we have used. However variance = 1/B**2, and their values
 -7  compare with the results from this program in all cases.
 -8
 -9  Read in the time values and the counts prior to 49, 55, etc. Note
-10  that the first count must be zero. This gives an indication of the
-11  the length of the lag.
-12  "
 13  READ [PRINT=data,errors,summary; SETNVALUES=yes] Time,COUNT

 14   49  0
 15   55  1
 16   62  7
 17   72 27
 18   79 22
 19   86  8
 20   96 13
 21  103  3
 22  120  6
 23  127  1
 24  144  1
 25  151  1
 26  168  1
 27  :

  Identifier   Minimum      Mean   Maximum    Values   Missing
        Time      49.0     100.9     168.0        13         0
       COUNT     0.000     7.000    27.000        13         0

 31  "
-32  Calculate ln(Time-48), as used by Hunter, Glasbey and Naylor,
-33     also the cumulative counts as required by the procedure.
-34  "
 35  CALCULATE LTIME = LOG(Time-48)
 36  & COUNT = CUMULATE(COUNT)
 37  "
-38  Use the procedure, assuming that ln(time to germination -48) is
-39  Normally distributed.
-40  "
 41  FITDIST [PRINT=m,s,e,f; MODEL=normal; LINK=identity] \
 42     DATA=COUNT; TIME=LTIME; INITIAL=!(3,3); SUBPOP=1
```

```
***** Cumulative Count Analysis *****

 Response variate:  COUNT
 Explanatory variate:  LTIME
 Distribution : Multinomial
 Link :     Identity, z[]=b[](X-m[])
 Model :    Normal, c[i]=prop[i]*normal(z[])
  Note : Fitted curve passes through cumulative count at last time point


***** Nonlinear regression analysis *****

*** Summary of analysis ***

             d.f.      deviance    mean deviance
Regression      2             *                *
Residual        9         6.885           0.7650
Total          11             *                *


***** Nonlinear regression analysis *****

*** Estimates of parameters ***

                 estimate         s.e.      Correlations
b[1]                1.814        0.138        1.000
m[1]               3.3301       0.0559      -0.011  1.000


 Number of Units capable of responding    91.38


***** Nonlinear regression analysis *****

*** Fitted values and residuals ***

                                          Standardized
        Unit     Response Fitted value      residual
           1         0.00            *             *
           2         1.00         0.55          0.65
           3         7.00         9.05         -0.85
           4        27.00        26.16          0.20
           5        22.00        16.75          1.47
           6         8.00        12.49         -1.64
           7        13.00        11.45          0.54
           8         3.00         4.89         -1.11
           9         6.00         6.09         -0.04
          10         1.00         1.21         -0.24
          11         1.00         1.56         -0.58
          12         1.00         0.33          1.13
          13         1.00         0.45          0.86

Mean                 7.00         7.58          0.03


 Note - Response is the difference of the Cumulative Counts

  43  "
 -44  Refit assuming that the lag-time is unknown, when it will be
      estimated. Note that in this case the estimated lag is very close
      to 48, as used by Hunter, Glasbey and Naylor (1984).
 -45  "
  46  FITDIST [PRINT=m,s,e,f,g; MODEL=normal; LINK=shift] \
  47     DATA=COUNT; TIME=Time; INITIAL=!(48,3,3); SUBPOP=1
```

```
***** Cumulative Count Analysis *****

  Response variate:  COUNT
  Explanatory variate:  Time
  Distribution : Multinomial
  Link :     Log with Lag, z[]=b[](log(X-lag)-m[])
  Model :    Normal, c[i]=prop[i]*normal(z[])
   Note : Fitted curve passes through cumulative count at last time point


***** Nonlinear regression analysis *****

*** Summary of analysis ***

             d.f.      deviance     mean deviance
Regression      3             *                 *
Residual        8         6.885            0.8606
Total          11             *                 *


***** Nonlinear regression analysis *****

*** Estimates of parameters ***

                 estimate         s.e.      Correlations
lag                 47.99         4.16         1.000
b[1]                1.814        0.362        -0.913  1.000
m[1]                3.330        0.182        -0.945  0.861  1.000


  Number of Units capable of responding    91.37


***** Nonlinear regression analysis *****

*** Fitted values and residuals ***

                                        Standardized
         Unit     Response Fitted value residual
            1         0.00            *         *
            2         1.00         0.55      0.62
            3         7.00         9.05     -0.81
            4        27.00        26.16      0.19
            5        22.00        16.75      1.40
            6         8.00        12.50     -1.56
            7        13.00        11.45      0.51
            8         3.00         4.89     -1.05
            9         6.00         6.09     -0.04
           10         1.00         1.22     -0.23
           11         1.00         1.56     -0.55
           12         1.00         0.33      1.07
           13         1.00         0.45      0.81

Mean                  7.00         7.58      0.03


 Note - Response is the difference of the Cumulative Counts
```



COUNT v. Time using symbol *
Fitted v. Time using symbol .

```
 48  "
-49  Carrying out a pseudo-Probit analysis assuming that at each time point
-50  the cumulative counts are bionomially distributed with a total for
-51  each equal to the total number germinated by the end of the experiment.
-52  "
 53  SCALAR NGERM
 54  CALCULATE NGERM = MAX(COUNT)
 55  & VNGERM = NGERM*(COUNT/COUNT)
 56  MODEL [DISTRIBUTION=binomial; LINK=probit] COUNT; NBINOMIAL=VNGERM
 57  FIT LTIME

57.....................................................................

***** Regression Analysis *****

 Response variate: COUNT
  Binomial totals: VNGERM
     Distribution: Binomial
    Link function: Probit
     Fitted terms: Constant, LTIME


*** Summary of analysis ***
    Dispersion parameter is 1

              d.f.      deviance     mean deviance
Regression       1             *                *
Residual        10         3.136           0.3136
Total           11             *                *

* MESSAGE: The following units have high leverage:
                              3                0.40


*** Estimates of regression coefficients ***

                 estimate         s.e.            t
Constant           -5.981        0.366       -16.36
LTIME               1.803        0.101        17.86
* MESSAGE: s.e.s are based on dispersion parameter with value 1

  58  "
 -59  Note the very low Residual Deviance, so that this method (as well
 -60  as being incorrect) is very insensitive to lack of fit.
 -61  "
  62  STOP
```

5. Acknowledgement

Long Ashton Research Station is financed through the Agricultural and Food Research Council.

6. References

[1] Brown, R.F. and Myer, D.G.
Problems in Applying Thornley's Model of Germination.
Annals of Botany, **57**, pp. 49-53, 1986.

[2] Glen, D.M. and Brain, P.
Pheromone Trap Catch in Relation to the Phenology of Codling Moth (*Cydia pomonella*).
Ann. Appl. Biol., **107**, pp. 429-440, 1982.

[3] Hunter, E.A., Glasbey, C.A. and Naylor, R.E.L.
The Analysis of Data from Germination Tests.
J. Ag. Sci., **102**, pp. 207-213, 1984.

[4] Hsu, F.H., Nelson, C.J. and Chew, W.S.
A Mathematical Model to Utilitise the Logistic Function in Germination and Seedling Growth.
J. Exp. Bot., **35**, pp. 1629-1640, 1984.

[5] Payne, C.C.
The Susceptibility of the Pea Moth, *Cydia nigricana*, to Infection by the Granulosis Virus of the Codling Moth, *Cydia pomonella*.
J. Inv. Path., **37**, pp. 71-77, 1981.

[6] Weaver, S.E., Tan, C.S. and Brain, P.
Effects of Temperature and Moisture on Time of Emergence of Tomatoes and Four Weed Species.
Can. J. Plant Sci., **68**, pp. 877-886, 1988.

Appendix: FITDIST, a Procedure for Analysing Cumulative Count Data

A procedure has been written to assist in fitting models to this type of data; a full listing can be obtained from the authors on request. A description of the four options and six parameters of the procedure is given below.

Options:

PRINT — What output to print:

m	model, including link
s	summary analysis of deviance
e	parameter estimates, including their correlations, estimates of the total number of units capable of responding, and the number in each subpopulation
f	fitted values, including residuals, and the variate of responses (the differences of the original accumulated counts in DATA variate)
mon	monitoring of the fitting process
g	graph of accumulated fitted values and accumulated counts against time
*	no output
default:	m, s, e, g

MODEL — Which CDF to use to fit to the DATA variate:

normal	Normal: link(TIME)
comp	complementary log-log: exp(–exp(–link(TIME))
logistic	logistic: 1/(1+exp(–link(TIME))
default:	normal

LINK — Which transformation of the TIME variate to use

log	b(log(TIME)–m)
shift	b(log(TIME–lag)–m)
identity	b(TIME–m)
own	link defined by the user, specified in OWN option: b(own(TIME)–m)

b, m and lag are the parameters to be estimated. Separate values of b and m are calculated for each SUBPOP

default: log

OWN — an expression for use when LINK is set to own. The expression must be declared beforehand, and must include a single parameter which is named by setting OWNP. The expression should be the right-hand side of an equation only, and should be a function of the TIME variate; for example:

```
EXPRESSION own; VALUE=!e(LOG(2*TIME-param))
```

OWN must be set with LINK=own, but does not need to be included otherwise. OWNP must be set.

OWNP — This option must be set if LINK is set to own. It must be set to the name of the parameter used in the expression set in the OWN option; for example:

```
FITDIST [LINK=own; OWN=!e(TIME-param); OWNP=param] \
   DATA=data; TIME=time
```

Parameters:

DATA A variate containing accumulated counts, the first of which must be zero. DATA should not contain any missing values. This must be set.

TIME A variate of the same length as DATA containing the time each count was recorded. TIME should not contain missing values. This must be set.

INITIAL A variate of initial parameter estimates in the order

lag, *b*[1...SUBPOP], *m*[1...SUBPOP], *prop*[1...(SUBPOP−1)]

lag only needs inclusion with LINK = shift. If LINK = own then the initial value of the parameter used in the expression OWN must be put in first position. The number of estimates for *b* and *m* must equal the number of subpopulations given by the parameter SUBPOP and the number of *prop*[] (proportions) should be one less than this.

prop[] only needs including with SUBPOP greater than 1; these are the proportions of the total population of individuals capable of responding in each subpopulation.

b should be non-negative, and *prop*[] should lie between 0 and 1. This parameter must be set.

SUBPOP Number of subpopulations. With SUBPOP greater than one, the proportion of the population in each subpopulation is estimated with parameters *prop*[]. Default 1.

STEP A variate of step-lenghths for the fitting process. Need not be set.

SAVEPOP A variate; if SUBPOP = 1, this holds the estimated number of units capable of responding in the population; if SUBPOP is not 1, it should be of length SUBPOP + 1, and will also hold the number in each subpopulation.

When calling the procedure, the chosen options should be specified exactly as specified; for example, MOD=normal not MOD=NORM.

If any of the products of the regression procedure are required at a later stage, they can be accessed using the RKEEP directive, in the usual manner. However, this does not apply to the number of units capable of responding, or the number in each subpopulation, which can only be saved by setting SAVEPOP.

Because the calculations in the procedure involve differencing the DATA vector, and of the requirement that the first element in DATA is zero, TIME and DATA should not be restricted.

Use of Genstat for Bootstrap Estimation of Parameters

P M E Altham
Statistics Laboratory
University of Cambridge
Cambridge
United Kingdom *CB2 1SB*

The following consultancy problem prompted me to consider the topic of bootstrap estimation in Genstat. Suppose we have data (x_i, y_i), $i = 1...n$ for which we wish to fit the model

$$y_i = f(\underline{\beta}; x_i) + \varepsilon_i, \qquad i = 1...n,$$

where $\varepsilon_1 ... \varepsilon_n$ are residuals, here assumed independent with zero mean, and variance σ^2, f is a known function, and $\underline{\beta}$ the unknown parameter of interest. We can use Gentat to obtain an estimate $\underline{\hat{\beta}}$ say of $\underline{\beta}$, by using the linear or nonlinear regression facilities of Genstat. In a particular scientific context, it may be important to look closely at the sampling properties of $\underline{\hat{\beta}}$, or functions of $\underline{\hat{\beta}}$. This is especially true in a nonlinear regression problem, for then $\underline{\hat{\beta}}$ is not a linear function of the observation $\underline{y}$ and so we cannot obtain its exact distribution.

For ease of exposition, bootstrap estimation is illustrated just for the slope in the ordinary linear regression of y or x, and is applied in the program below to some data kindly supplied by the Welding Institute, Abington, near Cambridge. The simulations could doubtless, with some extra effort, be programmed more efficiently in a language other than Genstat. Often, however, the statistician's time is more valuable than CPU time, and therefore it is very useful to be able to keep the whole program in Genstat.

The bootstrap procedure is as follows:

In the usual linear regression $y_i = \alpha + \beta x_i + \varepsilon_i$, $1 \le i \le n$, let $\hat{\alpha}$ and $\hat{\beta}$ be the least squares estimators of α and β and define the fitted values as

$$f_i = \hat{\alpha} + \hat{\beta} x_i$$

and the residuals as

$$r_i = y_i - f_i.$$

A single *simulation* consists of taking a random sample *with* replacement from $r_1 ... r_n$. Call this sample $r'_1 ... r'_n$ and define 'new' y–values, say $newy_i$, by

$$newy_i = f_i + r'_i \qquad i = 1...n.$$

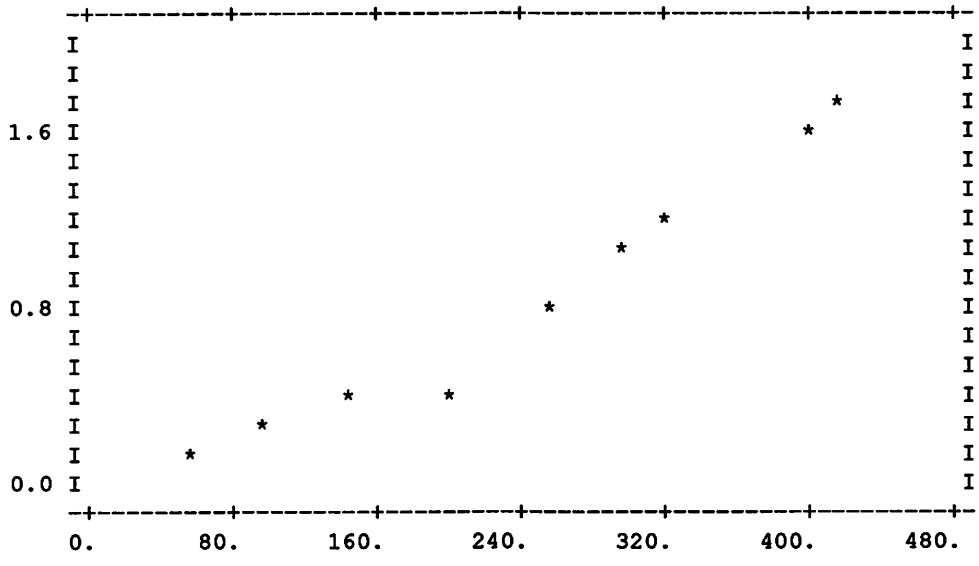
Now find the least-squares estimate of the slope in the regression of $newy_i$ on x_i. This estimate is then stored as an element of the variate *slope*.

Repeating this simulation say 100 times yields $slope_1 ... slope_{100}$, which we can use to study the precision of our estimate, for example by constructing the histogram of the 100 slope values and hence obtaining a 95% confidence interval for the parameter β. Efron and Tibshirani [1] give a comprehensive review of bootstrap methods.

Example

```
 1  "Bootstrap estimation of the slope in linear regression."
 2  READ [PRINT=data,summary; SETNVALUES=yes] y,x

 3   .18  52.8
 4   .22  97.4
 5   .42 200.1
 6   .34 142.9
 7   .74 253.7
 8  1.18 323.7
 9  1.06 294.6
10  1.76 412.0
11  1.54 398.7 :
```

```
  Identifier   Minimum      Mean   Maximum    Values   Missing
           y    0.1800    0.8267    1.7600         9         0
           x      52.8     241.8     412.0         9         0
  12  GRAPH y; x
```



y v. x using symbol *

```
  13  "Fit linear regression."
  14  MODEL y
  15  FIT x

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

***** Regression Analysis *****

 Response variate: y
     Fitted terms: Constant, x

*** Summary of analysis ***

             d.f.         s.s.         m.s.
Regression      1       2.6143      2.61428
Residual        7       0.1409      0.02013
Total           8       2.7552      0.34440

Percentage variance accounted for 94.2


*** Estimates of regression coefficients ***

                estimate         s.e.            t
Constant          -0.252        0.106        -2.38
x               0.004462     0.000392        11.40

  16  "Form the unstandardized residuals."
  17  RKEEP FITTEDVALUES=f
  18  CALCULATE r = y-f
```

```
 19  PRINT y,f,r

          y           f           r
     0.1800     -0.0165     0.19648
     0.2200      0.1825     0.03748
     0.4200      0.6408    -0.22075
     0.3400      0.3855    -0.04553
     0.7400      0.8799    -0.13991
     1.1800      1.1922    -0.01224
     1.0600      1.0624    -0.00240
     1.7600      1.5862     0.17377
     1.5400      1.5269     0.01311

 20  "Bootstrap: take many samples from the residuals."
 21  SCALAR nsample,count; VALUE=100,0
 22  VARIATE [NVALUES=nsample] slope
 23  CALCULATE size = NVALUES(y)
 24  "Initialize the random number generator."
 25  & init = URAND(87736; 1)
 26  FOR [NTIMES=nsample]
 27     CALCULATE count = count+1
 28     "Generate a set of random integers in the range (1,size)."
 29     & index = 1 + INTEGER(size * URAND(0; size))
 30     & newy = f + r$[index]
 31     MODEL newy
 32     FIT [PRINT=*] x
 33     RKEEP ESTIMATES=beta
 34     CALCULATE slope$[count] = beta$[2]
 35  ENDFOR
 36  "Display the sample distribution of the estimates of slope."
 37  HISTOGRAM slope

Histogram of slope


          - 0.0038   3 ***
     0.0038 - 0.0040   7 *******
     0.0040 - 0.0042   8 ********
     0.0042 - 0.0044  19 *******************
     0.0044 - 0.0046  22 **********************
     0.0046 - 0.0048  17 *****************
     0.0048 - 0.0050  14 **************
     0.0050 - 0.0052   6 ******
     0.0052 - 0.0054   4 ****
     0.0054 -          0

Scale:  1 asterisk represents 1 unit.

 38  STOP
```

Acknowledgements

I am grateful to the Welding Institute, Abington, Cambridge for suggesting this problem to me, and to my colleague Dr. G A Young for helpful discussions.

Reference

[1] Efron, B. and Tibshirani, R.
The Bootstrap.
Statistical Science, **1**, pp. 54-77, 1986.

Features of the Genstat 5 Language: 1

S A Harding
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Hertfordshire
United Kingdom *AL5 2JQ*

K I Trinder
NAG Ltd
Wilkinson House
Jordan Hill Road
Oxford
United Kingdom *OX2 8DR*

1. Introduction

The purpose of this article is to gather together items of practical information about using Genstat 5 that might be particularly valuable to users. Most of the items can be found in the Reference Manual, though the practical effects of some aspects of the language are not always laid out there. The title (perhaps presumptuously) has been given the number 1 since it is anticipated that there will be similar articles as more features of the Genstat 5 language become apparent. However, this depends on users, and the authors would welcome suggestions or contributions for later articles of the same nature. Note that individual items are intended to be short and that the information might well be about Genstat 5 on specific computer systems, (the PC version of Genstat 5 is a likely candidate, given the memory limitations).

2. Suffixed Identifiers and Pointers

Pointers provide a powerful means of referencing groups of identifiers. The elements of a pointer structure may be referred to either by the explicit identifier name or by the subscript equivalent. The penalty of such a facility is that the data space requirements when using pointers are high. This is not likely to pose a problem on most computer systems with virtual memory. However, on systems where the data space is limited, such as PCs for example, it may be necessary to use pointers with caution or avoid them altogether when handling large data sets.

Users who know Genstat 4 may find themselves using pointers in Genstat 5 without realising it. In Genstat 4,

```
'VARIATE' V(1...20)
```

declares 20 variates named `V(1)`, `V(2)` through to `V(20)`. However, the Genstat 5 statement

```
VARIATE V[1...20]
```

although superficially similar, is actually quite different since it declares a pointer named `V` pointing to 20 unnamed variates which may be referred to as the subscripted elements of V; that is, `V[1]`, `V[2]` through to `V[20]`. This may seem to be unnecessarily pedantic since `V[`*n*`]` in Genstat 5 is used in very much the same way as `V(`*n*`)` in Genstat 4; however, there are some important points that should be made.

(a) In Genstat 4, the whole list of variates may be referenced as `V(1...20)`. Similarly, in Genstat 5, `V[1...20]` may be used. In addition, in Genstat 5 the notations of `V[]` and `#V` may also be used in the same way. Note that *pointer*`[]` and `#`*pointer* are not the same if any of the elements of *pointer* are themselves pointers.

(b) In Genstat 4 it is quite permissable to have a data structure named `V` which has nothing to do with `V(`*n*`)`. This is not the case in Genstat 5, since `V` is a pointer structure.

(c) Chapter 3 of the Genstat 5 Reference Manual (page 70) describes how the first use of a suffixed identifier results in the implicit declaration of a pointer. For example, the statement

```
VARIATE V[1...10]
```

will result in the pointer V being set up with 10 values, each of which is then declared as a variate. This will not normally cause any problem; however, it does allow a large number of new structures to be formed within one statement and in exceptional cases this may cause the directory to overflow, generating an SP-4 diagnostic. The exact number of new structures that can be formed will vary according to how much space is available in the directory and how much workspace is available during the compilation of a statement, but as a rough guide anything above 200 (or about 50 in the case of the PC version) may cause an overflow, as in the implicit declaration of `V[1...200]` or `X[1...100],Y[1...100]`.

The problem can easily be avoided by declaring large pointers in advance of their use in suffixed identifiers: this will reserve sufficient space for the required structures. For example:

```
POINTER [NVALUES=200] V
READ V[1...200]
```

This is particularly important within procedures, where the number of suffixed identifiers declared within the procedure may be dependent on an input parameter, as in the following example:

```
PROCEDURE 'SUFFIX'
PARAMETER 'DATA'
  CALCULATE Ndata = NVALUES(DATA)
  POINTER [NVALUES=Ndata] Xval
  SCALAR Xval[1...Ndata]
  ...
  ...
ENDPROCEDURE
```

In normal use, `Ndata` may not be very large, but in some exceptional cases the procedure would fail if `Xval` were not first declared as a pointer.

(d) A further problem may arise with suffixed identifiers in a procedure. Consider the following statements:

```
PROCEDURE 'FRED'
PARAMETER 'DATA'
  ...
  ...
  CALCULATE Sum[1] = SUM(DATA[1])
  ...
  ...
ENDPROCEDURE
...
...
READ [CHANNEL=2] Var[1...5]
FRED Var
```

The pointer `Var` to some variates is passed as a parameter to the procedure and `Sum[1]` is assigned the sum of the values of `DATA[1]`. `DATA` is a dummy structure pointing to `Var`, so `DATA[1]` refers to `Var[1]`, as required. A problem may arise however if the pointer that is specified as the procedure parameter does not have suffixes that start at 1, as in the following case:

```
READ [CHANNEL=2] Yields[1980...1985]
FRED Yields
```

Now the reference to DATA[1] within the procedure will add a new value to the suffix list of Yields, which will now point to Yields[1,1980,1981...1985]. Yields[1] will be a new structure, of undefined type and without values, so the CALCULATE statement will fail.

To protect procedures against this potential problem a local copy should be made of parameters that are known to be pointers:

```
POINTER Copy; VALUES=DATA
```

The suffixes of Copy will be 1 upwards regardless of the actual suffixes specified for the pointer that DATA refers to. Within the procedure further references to DATA can be made using Copy[1] and so on, without altering the original structure. This approach does have one minor side effect, in that the names of the elements of Yields will now be printed as Copy[1], Copy[2], and so on; this can be dealt with at exit from the procedure by redeclaring the DATA parameter, as follows:

```
POINTER DATA; VALUES=DATA
```

The best solution is to set up dummy structures to point to the elements of the DATA pointer, thus avoiding any problems with names. For example the following statements could be inserted at the head of the procedure given above:

```
CALCULATE Nval = NVALUES(DATA)
POINTER [NVALUES=Nval] Copy
DUMMY Copy[1...Nval]; VALUE=DATA[]
```

A further note: this problem with suffixes does not arise if use of the pointer is restricted to the null suffix list, i.e. DATA[], throughout the procedure.

3. Loops

(a) FOR loops allow sequences of statements to be repeated, perhaps with one or more index structures changing with each pass through the loop. For example,

```
FOR I=A,B,C
  statements
ENDFOR
```

will execute the statements in the loop three times with the dummy I being successively set to the structures A, B and C. If a group of statements is to be repeated a number of times (12, say) without reference to any indexing structures, then it might be tempting to use

```
FOR I=1...12
  statements
ENDFOR
```

While this is valid, it is inefficient because it involves setting up unnecessarily the dummy I and twelve unnamed scalars storing 1 up to 12. It is better to use the NTIMES option of FOR, as in the following:

```
FOR [NTIMES=12]
  statements
ENDFOR
```

Setting up unnamed structures, as needed for 1...12 in the above example, requires a large amount of temporary data space, as with pointers. The following illustrates an alternative solution which takes less data space:

```
SCALAR I; VALUE=0
FOR [NTIMES=100]
  CALCULATE I = I+1
  statements
ENDFOR
```

This example assumes that I is to take the values 1 through to 100. It could easily be adapted so that the values are a different arithmetic progression or any other sequence of numbers.

(b) FOR loops along with IF blocks, CASE constructions and procedures make Genstat 5 a fully structured language. It is worth noting that other structured languages offer repeat-until loops, which take the form

repeat
  statements
until condition is true

and while-do loops, which take the form

while condition is true do
  statements
end do

Both of these constructions can be achieved in Genstat 5 by using FOR with the NTIMES option set to a large number together with the EXIT directive. The following example

```
FOR [NTIMES=9999]
  statements–1
  EXIT logical-expression
  statements–2
ENDFOR
```

is equivalent to a repeat-until loop (when *statements*–2 is empty) or a while-do loop (when *statements*–1 is empty). It may seem inelegant to set NTIMES to 9999 or any other large number, but this does ensure that an infinite loop cannot accidently be set up in Genstat.

4. The Language-Definition Files

An important feature of Genstat 5 is that the command language is defined externally. That is to say, information about directives (directive names, option names, parameter names, permitted option settings, default option settings and function names) are held in a file which is known as the Binary Language-Definition File. When Genstat is executed, this file is retrieved and each directive and function is defined. The file is not in an ASCII character format and cannot therefore easily be changed. However, another version of the file, called simply the Language-Definition File, is in character form and is supplied to sites specifically so that it may be altered using, say, an editor. A new binary language-definition file can then be formed from the edited language-definition file: the means of doing this are described in Chapter 12 of the Manual, and in the appropriate Installers Note for each implementation of Genstat 5.

The main reason for wanting to change the bootstrap file is to increase the internal data space. When directive definitions have been retrieved from the binary file, they are stored in the numeric workspace where all numeric data structures created by users are stored. Therefore, if certain directives are not required, their definitions can be deleted (or commented out) thus making more of the workspace available for other use. This is likely to be particularly useful for the PC version of Genstat, where space is seriously limited by the constraints of PCs and the MS-DOS and PC-DOS operating systems. As an example, W. Slob of RIVM in the Netherlands has informed us that deleting all directive definitions not required in a specific program resulted in a reduction in the binary language-definition file by a factor of 2.31 and an increase in the available data space by a factor of 1.44.

Having made this point, it should be stressed that any changes to the definitions render the Genstat language at a particular site incompatible with the standard language as given in the documentation (the Reference Manual, the Reference Summary, etc.) and the on-line help facility. It will also mean that any procedures which use undefined directives will not work.

There can also be unexpected problems in deleting directive definitions, because some directives assume that others have been defined. Here is a list of directives that are needed by others:

JOB	always needed
PRINT	needed by TABULATE, PREDICT and directives for multivariate analysis
GRAPH	needed by CORRELATE
SSPM	needed by FIT and TERMS
FIT	needed by TERMS
TERMS	needed by FIT
CALCULATE	needed by FITNONLINEAR
OWN	needed by FITNONLINEAR
INPUT	needed by BREAK

Note that one other use of the Language-Definition File is to define the graphics environment. This would normally be set up by the installer of Genstat at each site and should not be changed without consulting that person.