NAG®

GENSTAT

Newsletter

Issue No. 25

Editors

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

G W Morgan
NAG Limited
Wilkinson House
Jordan Hill Road
OXFORD
United Kingdom OX2 8DR

Printed and produced by the Numerical Algorithms Group Limited

ISSN 0269-0764

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

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

Genstat Newsletter
Issue No. 25

NP2047

March 1990

Contents

			Page
1.	Editorial		3
2.	New Facilities in Genstat 5 Release 2	*Genstat 5 Committee*	4
3.	The Genstat Menu System	*P Lane*	23
4.	SASS Experience of Teaching Statistics using Genstat	*C A Glasbey and G W Horgan*	36
5.	Detecting Parallelism in Nonlinear Models	*R Butler and P Brain*	40
6.	Fitting a General Growth Model to Data	*A Keen*	47
7.	The Inefficiency of the Recovery of Inter-block Information about Non-orthogonal Treatments from Genstat ANOVA	*K J Worsley*	51
8.	Using the EDIT Directive to Deal with Awkward Textual Data	*T R Butler-Stoney*	57
9.	Combining Tables with Variates	*Dr R Sackville Hamilton*	59

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

Editorial

Keith Trinder left NAG in July, and has relinquished his co-editorship of the Newsletter. He has been replaced by Geoff Morgan, who leads the Statistics Group at NAG. Both the current co-editors would like to record their thanks to Keith for all the work he has done to promote Genstat. We wish him well in his new post at Southampton University.

This issue of the Newsletter contains three more articles from the sixth Genstat Conference at Edinburgh. One describes the use of Genstat in teaching statistics to scientists, and the others concern procedures for fitting nonlinear models, both of which have since been submitted to the Genstat Procedure Library.

As promised in the previous issue, there is an article describing the new facilities in Release 2.1. This is a precis of the Genstat 5 Release 2 Manual Supplement, available from NAG. (For news about the progress of Release 2, see the News Section.) Information about the Genstat Menu System has been separated into another article, and this also updates the material presented at the sixth Genstat Conference and later at the one-day Conference on Interactive Statistical Modelling in April.

One article in this issue deals with problems of recovering inter-block information when analysing some designs. Finally, there are two short articles describing useful techniques that users may not be aware of: combining tables with variates, and using the `EDIT` directive. The editors are very keen to see more short articles of this nature, to complement the longer and more specialized ones that have dominated recent issues.

Genstat News

Progress on Release 2

Implementations of Release 2.1 started in August 1990, after extensive testing of the base version of the Fortran source for portability using the Toolpack routines available from NAG. The guides for implementors and the material supplied to help them have also been considerably redesigned in the light of experience gained from Release 1. It is hoped, therefore, that implementors will have a much easier task with Release 2 than was possible with Release 1,which was the first release to make use of Fortran 77 and to include the changes consequent on redesigning the Genstat language.

The first completed versions should be distributed during September. These will be accompanied by the Genstat 5, Release 2, Reference Manual Supplement and Reference Summary (both printed at NAG). At present, there are versions of Genstat Release 1.3 for 17 operating systems, with four more in progress. These systems range from MS-DOS for XT-compatible PCs to the Cray X-MP. Three further systems have versions for Release 1.2, with work still in progress on Release 1.3.

Seventh Genstat Conference

The Seventh International Genstat Conference will be held at the Pappendal Conference Centre near Arnhem, The Netherlands, from Monday 23rd to Friday 27th September 1991. A.A.M. Jansen is chairman of the local Organizing Committee, and Paul W. Goedhart is the Secretary; both are members of the Agricultural Mathematics Group, Wageningen. The two other members of this committee are Valentijn Van den Berg and Margriet Stapel.

New Facilities in Genstat 5 Release 2

Genstat 5 Committee
Statistics Department
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Herts AL5 2JQ

The full documentation of Release 2 is published by NAG as the 'Genstat 5 Release 2 Manual Supplement'. This article summarizes the new facilities, but includes neither the examples nor the detailed description of options and parameters that are to be found in the Supplement. As in the Supplement, the sections of this article are numbered to correspond to the equivalent chapters of the Genstat 5 Reference Manual (1987), for ease of comparison.

1. Introduction

Release 2 is the first major upgrade to Genstat 5; apart from the few exceptions listed in the Appendix, the changes are all compatible with Release 1. Thus, nearly all programs written for Genstat 5 Release 1 should be able to run in exactly the same way with Release 2. In many cases the only differences to notice will be that Genstat has required less execution time, and that there was more workspace to spare.

The extra workspace has been obtained by a change in the way in which the details of the Genstat syntax are stored [12]; as a result, Genstat starts up more quickly, and the tidying of workspace is less time-consuming. This is just one of the ways in which efficiency has been enhanced in Release 2. Also, for example, procedures are stored very much more compactly, and the contents of procedure libraries are easier to ascertain so that Genstat can recognize an invalid statement more quickly.

There are other improvements that have not required any extensions to the syntax. For example, the READ directive allows easier recovery from errors during interactive use [4], and estimated regression coefficients, saved by RKEEP, now have unit labels like those in the printed output. Another unseen addition is that Genstat automatically opens and executes a start-up file at the outset of any job. This would allow you, for example, to open a file to contain a transcript of input and output (using the OPEN and COPY directives), initialize for high-resolution graphics (directives AXES, DEVICE, FRAME and PEN), and so on. There is also a standard Start-up File supplied with Release 2, which is used to set up the Genstat Menu System [2].

Many directives have been extended, by adding new options or parameters, or by allowing further settings of options or parameters. In most cases, the new options or parameters have been placed at the end of the existing list, so that statements in Release 1 that made use of Genstat's abbreviation rules would not need to be changed for Release 2. For example, the statement

```
    SET [DIAGNOSTIC=faults]
```

can be abbreviated in Release 1 to:

```
    SET [D=faults]
```

In Release 2, there are extra options, DSAVE and PROMPT. However, these occur after all the other options, so the abbreviation of DIAGNOSTIC to D is still valid. For a few directives, however, placing the new options or parameters at the end would have meant that options of the same name might have occurred in different orders in different directives. For example, in a language-definition file, the OPTION directive has a VALUES parameter which occurs before the DEFAULT parameter (see Genstat 5 Reference Manual, page 612). To facilitate the automatic checking of options and parameters of procedures, there was the need also to include an NVALUES parameter (amongst other new parameters). In the directives that declare data structures, NVALUES always precedes VALUES. Consequently, NVALUES has been inserted before VALUES in the OPTION directive. All the directives that have been affected by such changes are listed in the Appendix.

There are also 12 completely new directives. In the input/output section, the QUESTION directive provides the basic tool for building menu-driven conversational interfaces to Genstat.

QUESTION	obtains a response using a Genstat menu [4.1.1].

There is now a directive for interpreting and manipulating formulae; the other new directive in the calculations section provides monotonic regression.

FCLASSIFICATION forms a classification set for each term in a formula, breaks a formula up into separate formulae (one for each term) and applies a limit to the number of factors and variates in the terms of a formula [5.5].

MONOTONIC fits an increasing monotonic regression of *y* on *x* [5.6].

For high-resolution graphics, there is one extra type of display and two new directives to facilitate interactive graphics.

COLOUR enables the red, green and blue intensities of the Genstat colours to be modified (where permitted by the graphics device) [7.2.1].

DREAD reads the locations of points from an interactive graphical device [7.3.1].

DSURFACE produces perspective views of two-way arrays of numbers [7.3.2].

Nonlinear regression is enhanced by the ability to produce estimates of functions of the parameters, together with likelihood-based standard errors and covariances.

RFUNCTION estimates functions of parameters of a nonlinear model [8.2.1].

The facilities for multivariate analysis have been extended by the addition of non-metric multidimensional scaling.

MDS performs non-metric multidimensional scaling [10.1].

Finally, the REML algorithm is now available for estimating variance components and for analysing unbalanced designs.

REML fits a variance-components model by residual (or restricted) maximum likelihood [13.2].

VCOMPONENTS defines the variance-components model for REML [13.1].

VDISPLAY displays further output from a variance-components analysis [13.3].

VKEEP copies information from a variance-components analysis into Genstat data structures [13.4].

2. The Environment of a Genstat Program

The Genstat system now includes menu facilities. The standard Genstat Menu System allows people to use many standard techniques in Genstat without having to learn the command language. However, the Menu System is designed to be extended, so that knowledgeable users can modify the standard system, or construct completely new systems, for the convenience of themselves or their colleagues. These new facilities are described in a separate article in this Newsletter, so no further details are given here.

Another major addition is that Genstat automatically opens and executes a start-up file at the beginning of any job. The standard Start-up File affects only interactive use: it prints a message and arranges for a copy of statements to be kept in a file called G21COMND. You can copy the standard Start-up File, and add extra statements – perhaps a SET statement to specify that the case of identifiers is to be ignored, or OPEN statements to allow access to files that you need regularly, or TEXT statements to define macros storing statements or parts of statements that you use frequently.

Other changes to the directives concerned with the environment of Genstat programs are designed to improve interactive working, and to provide extra information – particularly for procedure writers.

2.1. Information About the System

The information available on-line from the HELP directive has, of course, been updated to include information about the new directives and options. In addition, there are new codes: ampersand (&) to repeat a section of text, less-than (<) as a synonym of <RETURN> to go back a level in the hierarchy of information, and <RETURN> at the top level as a synonym of colon (:) to exit from HELP. All the main subjects of information now include a DESCRIBE keyword which provides general information about that subject.

The HELP directive and the DISPLAY directive now have a CHANNEL option. Both these directives can send output either to an auxiliary file or to a text structure as, indeed, can DUMP [2.3], RDISPLAY [8.1] and ADISPLAY [9] in Release 2.

2.2. Setting and Accessing Details of the Environment

If you do not like the prompt that Genstat issues when expecting a statement, you can reset it with the PROMPT option in either the JOB or SET directives; the default prompt is the greater-than symbol (>). The current setting of the prompt can be extracted using the corresponding option in the GET directive.

The COPY directive has been modified to make it easier to keep records of an interactive session. It is now possible to keep separate records of input and output. For example,

```
OPEN 'GEN.REC','GEN.OUT'; CHANNEL=2,3; FILETYPE=output
COPY [PRINT=statements] 2
COPY [PRINT=output] 3
```

will keep a record of all statements in the file GEN.REC and of all output in the file GEN.OUT. A later statement

```
COPY [PRINT=statements,output] 2
```

will stop output from being directed to GEN.OUT (because information can be copied to only one file at a time), sending it instead to GEN.REC together with the statements.

The standard version of the Start-up File arranges to copy statements, given in interactive mode, to a file called G21COMND, attached on Channel 5. The Genstat Menu System [4.4] generates Genstat statements that would duplicate the work done in response to menus, and stores these in the file G21COMND; it also copies output to the file G21RESLT.

2.3. Accessing Details of Data Structures

The DUMP directive has also been extended to show some extra information. This is mostly designed for the developers of Genstat, but may be of help to other users. If the option INFORMATION is set to 'full', then DUMP will provide information about all structures that are associated with the structures in the primary parameter list. Thus, if you ask for information about a factor that has labels defined for its levels, the text structure that stores the labels will also be displayed. You can also ask for information about an unnamed structure, by giving its (negative) reference number (as displayed by DUMP when indicating its association with another structure) in the parameter list.

The COMMON option of the DUMP directive has an additional setting, 'chrs', to display the contents of the Fortran common /CHRS/, which stores information about the local codes used for Genstat's character set.

The settings of the ATTRIBUTE option of the GETATTRIBUTE directive have been extended so that any attribute of a data structure can be saved; some were unavailable in Release 1.

3. Data Structures

In Release 2, there are no changes to the directives concerned with declaration of data structures.

4. Input and Output

One major change to input and output is the addition of the QUESTION directive and the Genstat Menu System, described separately. Another important change in the VAX/VMS version of Genstat is that Fortran carriage-control characters are no longer used in output files. Instead, all output files from Genstat are standard ASCII files, using the form-feed character (ASCII code 12) to indicate page throws where necessary. We hope that other implementations of Genstat will also use this method, but this depends on the availability of non-standard facilities of Fortran.

The READ directive has been enhanced to improve interactive working, as described in [4.1]. The PRINT directive can now deal properly with formula and expression structures [4.2], and small improvements have been made to the OPEN directive [4.3], the OUTPUT directive [4.4] and backing-store facilities [4.5].

4.1. Reading Data

The READ directive is now friendlier when you type data directly at the terminal, allowing recovery from some types of error. In common with the HELP and EDIT directives, READ now issues a prompt that makes it clear that you are expected to type information specific to that directive rather than to type another Genstat statement.

4.1.1. The QUESTION directive

This directive is designed to prompt at the terminal for information, as required in the Menu System. It is described in a separate article in this Newsletter.

4.2. Printing Data

The PRINT directive is now able to display formula and expression structures properly. In Release 1.3, only the internal coded form of these data structures could be displayed, which was uninterpretable by most Genstat users.

4.3. Getting Access to External Files

The OPEN directive has an enhancement that may be particularly useful when working interactively. The statement

```
OPEN 'FRUIT.DAT'; CHANNEL=*
```

will open the file FRUIT.DAT for input on the next available input channel. If all input channels are open, then you will see a diagnostic message, and will have to CLOSE one of the channels; but otherwise you do not have to recall precisely which channels you have used already in the session.

4.4. Transferring Input and Output Control

All output can now be indented automatically. This extension has been made particularly to help people who need to put printed Genstat output into binders, punching holes at the edge of the paper. For example, assume that your implementation of Genstat allows 132 characters per line when printing to a file (as opposed to a terminal), and that your printer displays 10 characters to the inch. Then the statement

```
OUTPUT [INDENTATION=10; WIDTH=127] 1
```

at the beginning of a program (or in your start-up file) will ensure that there is a one-inch margin to the left and a half-inch margin to the right of all output. Of course, only 117 characters can then be printed on any line, but Genstat will handle the arrangement of data display within these limits automatically.

4.5. Storing and Retrieving Structures

The RETRIEVE directive has a new option FILETYPE to allow information to be retrieved from procedure libraries, as well as from backing-store files. This is likely to be useful mainly to people who construct their own procedure libraries, and want to include other information in the library alongside the procedures. The help information for the procedures in the Genstat Procedure Library can already be accessed using the LIBEXAMPLE procedure.

5. Data Handling

There are two new directives, for manipulating formulae [5.5] and for performing monotonic regression [5.6]. There are also 16 new functions [5.1]; these include several statistical transformations, functions for inserting missing values, for calculating correlations and covariances, and for character handling. There are also major improvements to tabulation, to cater for weights, to calculate medians, and to cope with complicated data input such as occurs in hierarchical surveys [5.4].

5.1. Functions For Use in Expressions

The new MVINSERT function is the converse of MVREPLACE. You can use this function to place missing values into appropriate positions in the first argument as determined by true values in the corresponding locations in the second argument.

The POSITION function finds the unit where each value of one vector first occurs within another vector, or returns the value zero if it does not occur.

Two new scalar functions, COVARIANCE and CORRELATION, form covariances and correlations from variates. Correspondingly, there are two new variate functions, VCOVARIANCE and VCORRELATION; they expect pairs of pointers as arguments, and give as results the variate of covariances and correlations across the sets of values of the variates in the pointers.

There are five new statistical functions providing transformations of percentages and the corresponding back transformations, or inverses. The IANGULAR function is the inverse of the existing ANGULAR function, and turns degrees into percentages. The other new functions are LOGIT and CLOGLOG with their corresponding inverses, ILOGIT and ICLOGLOG. The logit transformation is given by

$$\text{logit}(P) = \log(P/(100-P)) \qquad 0 < P < 100$$

while the complementary loglog transformation is given by

$$\text{cloglog}(P) = \log(-\log((100-P)/100)) \qquad 0 < P < 100$$

New to CALCULATE are functions that allow information to be obtained about text structures. The existing NVALUES function works with a text structure, simply giving the number of lines. The new CHARACTER function gives you a variate containing the length of each line of the text. The remaining functions, GETFIRST, GETLAST, and GETPOSITION, also return variates. GETFIRST and GETLAST find the position of the first or last non-space character in each line respectively. GETPOSITION lets you find the position, in each line of a text, of the corresponding line from the text in the second argument.

The CONSTANTS function (synonym C) allows you to use the numeric value of certain special mathematical constants without having to remember their actual value or a particular way in which to calculate them. Three constants are available through the string-type argument, which is case-insensitive and allows abbreviation; they are as follows:

'pi'	pi, the ratio of the circumference of a circle to its diameter;
'e'	Napier's e or the base of natural logarithms;
'missingvalue' or '*'	missing value.

5.2. Operations on Text

The other extension for character handling is that the CONCATENATE directive can now change the case of letters, as specified by the new option CASE. By default, CASE=given leaves the case of each letter as given in the existing text. To change all letters to upper case (or capitals) you can put CASE=upper, or CASE=lower to change all letters to lower case. Alternatively, CASE=changed puts lower-case letters into upper case, and upper-case letters into lower case.

5.3. Operations on Pointers

In a Genstat program, you may sometimes have dummies pointing to other dummies, in a chain. This can often happen when one procedure calls another, passing one of its own arguments as the argument to the procedure that it calls. In Release 1 there was no way of assigning a value to any except the last dummy of the chain; thus, for example, it was impossible to write a procedure to assign defaults to the unset options or parameters of other procedures. In Release 2, the ASSIGN directive has been extended with an NSUBSTITUTE option to allow dummies to be substituted a set number of times in order to determine which dummy in a chain is to be assigned a value.

5.4. Operations on Tables

The TABULATE directive has three major additions: first, tables of medians, and of any other quantiles, can be formed; second, any kind of table can be weighted with respect to a variate of weights; and third, the input of data that are to be tabulated can be controlled explicitly at the Fortran level.

Weighting is provided by a new option called WEIGHT. Weighting is already available for regression and analysis of variance, and now it has been added to TABULATE. You can, in general, think of weights as a set of multipliers which are applied to the data before any operations are performed. This is not quite what happens in the case of variances, where weights are assumed to be integers representing numbers of replicates.

Quantiles can now be tabulated, using the QUANTILES parameter or the 'quantiles' setting of the PRINT option. The simplest quantile you can produce, and the one you get by default, is the median (50% quantile); but by using the PERCENTQUANTILES option you can obtain any percentage point (between 0 and 100, of course).

A new facility, which requires a knowledge of Fortran, allows you to tabulate data that may not be easily handled by Genstat. For instance, hierarchical data or data requiring different operations on different types of units before tabulation, can be handled easily. OWNTAB is a Fortran subroutine, supplied by the user, which is called from within TABULATE for each unit to be tabulated. It contains switches to tell TABULATE when a data error occurs or when all the data have been read. To use it you have to link your own version of Genstat, as when using the OWN directive described in Chapter 12 of the Genstat 5 Reference Manual. Then your version of OWNTAB will be used instead of the standard version supplied as part of Genstat. The subroutine can be as simple or as complicated as you like (or need), provided it obeys a few simple rules. A very simple version, reading two variates and two factors, is supplied with Genstat. To make worthwhile use of OWNTAB you would have to write your own version, and link it into your own private version of Genstat. Five options have been added to TABULATE for the benefit of OWNTAB. These are OWN, OWNFACTORS, OWNVARIATES, INCHANNEL, and INFILETYPE.

5.5. The FCLASSIFICATION Directive

The new directive, FCLASSIFICATION, allows the manipulation of formulae. As explained in Section 1.5.3 of the Genstat 5 Reference Manual, when Genstat uses a formula in a statistical analysis, it is expanded into a series of model terms linked by the operator plus (+). FCLASSIFICATION allows you to do this expansion yourself, and save the result in another formula using the OUTFORMULA option; at the same time you can use the FACTORIAL option to apply a limit to the number of factors and variates in the resulting terms. The NTERMS option allows you to find out how many terms the expanded formula will contain. Also, you can obtain the classification sets of the terms, or save the terms individually in separate formulae.

5.6. The MONOTONIC Directive

Monotonic regression plays a key role in non-metric multidimensional scaling, for which the MDS directive is now available 10.1. However, it can be useful in its own right, so the MONOTONIC directive has also been introduced in Release 2. A monotonic regression through a set of points is simply the line that best fits the points subject to the constraint that it never decreases: of course the line need not be straight, and in fact it rarely will be. If you need a monotonically decreasing line, you can simply subtract all the y-values from their maximum, find the monotonically increasing regression, and then back-transform the data and fitted line, and change the sign of the residuals.

The MONOTONIC directive has no options. Its four parameters specify, in order, the y-values, x-values, and variates to save the residuals and fitted values from the regression. The x-values need not be supplied, in which case the directive assumes that the y-values are in increasing order of the x-values. In common with the other regression directives, the variates to save the residuals and fitted values need not have been declared in advance.

6. Program Control

6.1. Procedures

One important aim for Release 2 has been to improve the efficiency of procedures. The way in which Genstat stores procedures internally has been redesigned so that they use far less space and take less time to execute. This also means that less time and space are required to store procedures on backing store. Procedure sub-files of backing-store files have been restructured to make it easier for Genstat to determine their contents. Retrieval is thus very much faster; this is particularly noticeable when procedures are retrieved automatically from large libraries. A desirable side-effect of this improvement is that, if you mistype a directive name, there is no longer a long wait while the procedure libraries are searched. However, as a result, procedure libraries formed with Release 1 cannot be used with Release 2.

Although much less space is required by procedures it may still sometimes be useful to delete some of the procedures stored within a job. The DELETE directive now has an option called PROCEDURE to indicate whether the structures to be deleted are procedures or ordinary data structures. This operates similarly to the PROCEDURE option of the STORE directive. Thus, to delete the procedure MyProc, you would put

```
DELETE [PROCEDURE=yes] MyProc
```

The default setting, PROCEDURE=no, causes ordinary data structures to be deleted, as with Release 1. Note that the use of DELETE above deletes MyProc only from Genstat's internal workspace, and not from any of the libraries attached to the job.

The LIST option can be used to delete all the procedures currently stored within Genstat's internal workspace, exactly as for ordinary data structures:

```
DELETE [PROCEDURE=yes; LIST=all]
```

For writers of procedures, the main improvement is that the OPTION and PARAMETER directives have been extended to allow option and parameter settings of procedures to be checked automatically when the procedure is executed. These directives now have extra parameters NVALUES, VALUES, DEFAULT, SET, DECLARED, TYPE, COMPATIBLE, and PRESENT to define the various aspects of the options or parameters that must be checked. There was a DEFAULT parameter of OPTION in Release 1, but this was available only within a language-definition file (see page 612 of the Genstat 5 Reference Manual), and not within procedures.

When a procedure is being defined interactively, it may be useful to store the statements that it contains so that they can be modified later. The PROCEDURE directive now has a SAVE option which allows you to specify the identifier of a text structure to save the subsequent statements, up to and including ENDPROCEDURE. The saved version is a modified form of the original input. Each line of the text contains a single statement. Thus, where a statement spans several lines of input, these are concatenated into a single line in the text (deleting the continuation characters). Also, any line that contains several statements is split. Comments are removed, and any occurrence of several contiguous spaces is replaced by a single space. Also, a colon is placed at the end of each line.

6.2. Breaks and Exits

The BREAK directive now has an unnamed first parameter. This can be set to a scalar logical expression, as with the IF and EXIT directives. If the expression yields a 'true' (i.e. non-zero) result, there will be a break in execution; if 'false' (zero) there will be no break.

In procedures (or other control structures), there is often the need to test a condition like the validity of an option or parameter, print a warning message, and then exit. To simplify this process in Release 2, the EXIT directive has been extended with an EXPLANATION option which can be set to a text to be printed if the exit occurs.

7. Graphical Display

There have been a number of minor internal changes and changes to defaults that improve the output from high-resolution graphics. Extensions to the existing syntax simplify some tasks and add new facilities. There are new directives to provide surface plotting, extra interactive facilities, and control of the colours associated with Genstat pens.

A major change for Release 2 is the inclusion of a basic graphical interface that does not require additional graphics software (e.g. GKS). This will provide HPGL and PostScript output to files; in addition VAX installations will also have full interactive facilities on ReGIS terminals (VT125/240/340). As before, additional interfaces will be provided for the graphics packages GKS, Ghost, and Gino, to allow access to a greater number of devices where possible.

7.1. Line-printer Graphics

There are no changes to the directives that produce line-printer graphics.

7.2. The Environment for High-Resolution Graphics

Graphics procedures may be made more self-contained through extensions to the GET and SET directives which allow the graphics environment to be restored to its original state on exit from a procedure. The special structures obtained by GET now include a 'dsave' structure. This is a special pointer that stores all the current settings of the graphics environment. The dsave structure can then be used in the DSAVE option of SET to reset the graphics environment.

The AXES directive has three new parameters: PENTITLE, PENAXES, and PENGRID. These specify which pens are to be used for the various components of a set of axes: the overall title, axes and annotation, and grid respectively. Thus, you now have control over which pens are used for drawing axes in each window, and the attributes of these pens.

The PEN directive has a new setting for the METHOD parameter: 'fill' allows area filling to be performed by DGRAPH [7.3]. The number of pens has been increased to 20. The COLOUR parameter can now be set to an integer in the range 0...15; where possible, colour 0 is interpreted as the background colour, allowing points to be erased from a plot. This will generally work only on colour graphics terminals. The number of BRUSH settings has been increased to 32: brushes 1...16 will produce the same patterns as before and brushes 17...32 are defined individually for each device. Where possible, hardware fill will be used for brushes 17...32; this will be much more efficient (i.e. faster) for some terminals, but may produce different results on different devices. The FONT parameter can be set to an integer between 1 and 10 to select different fonts for text appearing as titles, axis annotation, plotting symbols, and key information. The THICKNESS parameter controls the thickness of plotted lines. The standard thickness is defined to be 1.0; setting the THICKNESS of a pen to other values will multiply the thickness by the specified amount. The SIZE parameter is used similarly to change the size of plotted symbols and text.

The DEVICE directive has a new parameter ENDACTION with settings 'continue', 'pause', and 'unchanged'. This controls the action taken at the end of a plot. When using a graphics terminal interactively it may be convenient to pause at the end of a plot to examine the screen. When you are ready to continue, pressing the carriage-return key (<RETURN>) will switch the terminal back to text mode and the Genstat prompt will appear.

The default window definitions used in the FRAME directive have been changed. Windows 1...4 retain their original settings, but 5,6,7,8 are the top-left, top-right, bottom-left, and bottom-right quarters of the frame respectively.

7.2.1. The COLOUR directive

The COLOUR directive allows you to redefine the colour map stored internally. Genstat uses the RGB colour system to define each colour (0...15) in terms of its red, green, and blue components. These are specified as values in the range [0,1]. Thus black is represented by (0,0,0), white by (1,1,1), red by (1,0,0) and so on. The COLOUR directive can be used in three ways: firstly to define a colour in RGB terms; for example, to define colour 1 as yellow:

```
COLOUR 1; RED=0.5; BLUE=0.5; GREEN=0.0
```

Points plotted as colour 1 will then appear yellow. The MATCH parameter allows a colour to take its RGB values from the current settings of another colour; for example,

```
COLOUR 2; MATCH=1
```

will set colour 2 to be yellow also. Note that if colour 1 is changed again, colour 2 will not be altered. Finally a colour can be returned to its default settings by specifying only the colour number; for example,

```
COLOUR 1,2
```

will set colours 1 and 2 back to their original values. The background colour may be altered by changing the definition of COLOUR 0.

7.3. High-resolution Graphics

Histograms no longer have horizontal grid lines, and the AXES directive can be used to provide more control over histogram axes and annotation. The STYLE parameter settings allow you to obtain the x-axis only (by specifying 'x' or 'none'), *x*- and *y*-axes ('y','xy','grid'), or *x*- and *y*-axes with a box ('box'). The YTITLE, YUPPER, YMARKS, and YLABELS parameters allow annotation to be specified for the *y*-axis.

When plotting graphs, the axis bounds will by default be extended by 5% of the data range at each end, and the axis origins will by default be at the bottom-left corner of the plot. These defaults can easily be overridden with the AXES directive, using the YLOWER, YUPPER, XLOWER, XUPPER, YORIGIN, and XORIGIN parameters. The *y*-axis title is now produced with rotated text, where permitted by the underlying graphical software, and the graph title is no longer repeated in the key window.

The PEN parameter of DGRAPH can now be set to a variate or factor, which allows different pens to be used for different subsets of the units. All units which are at the first level of the factor are drawn first, using pen 1 with associated symbol, method, colour etc. Then all units at the second level of the factor are drawn separately, with pen 2, and so on. If PEN is set to a variate, its values are used similarly to define the pen for each unit.

A new method of plotting points is available using METHOD=fill in the PEN directive. The Y and X variates define a set of points which are joined by straight lines to form one or more polygons which are then filled using the brush pattern specified for the pen. It is important to remember that the JOIN setting of PEN is used to determine in which order the points should be joined; the settings 'given' and 'ascending' can produce quite different results.

In the DCONTOUR directive, the PEN parameter can now be set to a variate containing a list of pen numbers. This is to allow highlighting of particular contours. The first value specifies the pen for the first (lowest) contour, second value for the second contour, and so on. The list is recycled if there are too few values for the number of contours to be plotted. For example,

```
DCONTOUR Matrix; PEN=!(1,1,1,2)
```

will produce a contour plot where every fourth contour is drawn by pen 2. It is now possible to plot an irregular grid of contour heights by defining the matrix to be contoured with variates to define its rows and columns.

7.3.1. The DREAD directive

The DREAD directive provides graphical input on interactive terminals, where this facility is supported by the underlying graphics package. The exact implementation details will vary according to the underlying graphics package and terminal, but this should be explained in the local documentation. In general, a cursor will appear on the graphics screen or window. This can be positioned using cursor keys or a mouse, then pressing a key or mouse button will read the *(x,y)* coordinates of the cursor. The cursor can then be moved to another position and another pair of values read. The values obtained will be in the scale of the data that has been plotted in that window. You can use graphical input within any window that contains a graph, but you cannot input data from histograms, contours, etc. You can identify particular points on an existing graph and can mark the points you have read.

The PRINT option of DREAD is similar to the PRINT option of READ. The CHANNEL and WINDOW options are used to specify which device and window to read from; the default is to read from window 1 of the current device. The values read are stored in a pair of variates specified in the Y and X parameters. Any number of points may be read in one DREAD statement. If the number of points is known in advance, the Y and X variates can be declared with the appropriate length. Alternatively the option SETNVALUES can be set to 'yes', in which case points are read until you terminate the input. This can be done in two ways, either by attempting to read a point lying outside the current axes or by pressing a key or mouse button that has been specifically set up for this purpose.

Several types of cursor may be available. You can select which cursor to use by setting CURSORTYPE to an integer between 1 and 10. Normally cursors 1, 2, and 3 are different graphics cursors; for example, large cross-hair, arrow and small cross. Cursors 4 and 5 may be set up to provide special functions called rubber-band and rubber-rectangle. A rubber-band cursor works by reading one point in the normal way (as if CURSORTYPE was set to 1). This defines an anchoring point for a line whose other end is attached to the cursor. As you move the cursor about the line will change direction and contract or expand, but always linking the fixed point to the current cursor position: hence the term 'rubber-band'. When you read the next point this will become the anchor point for a new rubber-band segment which you use whilst locating a third point, and so on until the required number of points have been read. The rubber-rectangle works in a similar way, with the first point being read with a normal cursor. This defines the fixed point and the cursor is now regarded as being attached to the diagonally opposite corner of a rectangle that contracts and expands as you move the cursor around the screen. Reading the second point terminates the input; using a rubber-rectangle cursor will always read exactly two values, ignoring the SETNVALUES option and any predefined length for Y and X.

The PEN parameter of DREAD can be used to specify a pen which will be used to plot each point as its position is read. The various attributes of the specified pen, as defined by a PEN statement, will determine how the points are plotted. If the pen method is set to 'line', 'monotonic', 'open', or 'closed', then straight line segments will be drawn between the points; otherwise just the points themselves will be plotted. If the points are to be joined by lines and a rubber-rectangle cursor is being used, then the rectangle will be drawn in rather than the diagonal.

The YGIVEN, XGIVEN, and SAVESET parameters can be used together to enable DREAD to identify points that form part of an existing plot. The YGIVEN and XGIVEN parameters should be set to the *y*- and *x*-variates that were plotted on the graph. The identifier associated with the SAVESET parameter is set up as a variate having the same length as the Y and X structures output by DREAD. Each value of this variate will be set to the unit number of the point in YGIVEN and XGIVEN that is nearest to the corresponding point in the Y and X variates. The variate thus defined could then, for example, be used in CALCULATE or RESTRICT statements to obtain the actual coordinates of the plotted points that were selected by DREAD; whereas the Y and X structures output by DREAD would contain the coordinates of the exact position of the cursor.

7.3.2. The DSURFACE directive

The DSURFACE directive produces a perspective plot of a surface. The data are supplied as a grid of values in a matrix, two-way table, or set of variates; as in the DCONTOUR directive, the x- and y-axes are represented by the rows and columns of the data structure and the data values are heights, or z-values, at each (x,y) position. The display that is plotted represents a view of the surface from a particular viewpoint. The position of this viewpoint is defined by the options ELEVATION and DISTANCE which define the angle of elevation (in degrees) and distance from the surface. Rotation of the surface in the horizontal (i.e. x,y) plane is controlled by the AZIMUTH option; the default angle is 225 (degrees) which ensures that the element in the first row and column of the data structure is nearest the viewpoint. The default settings of ELEVATION, DISTANCE, and AZIMUTH have been chosen to produce a reasonable display of most surfaces; but if, for example, some parts of the surface are obscured by high points these options may be set to other values to get a better view.

As in the DCONTOUR directive, there are LOWERCUTOFF and UPPERCUTOFF options that specify lower and upper bounds for the grid values, but in addition the ZORIGIN option defines the origin of the z-axis. By default, LOWERCUTOFF and UPPERCUTOFF will be the minimum and maximum grid values and ZORIGIN defaults to the LOWERCUTOFF setting.

The TITLE, WINDOW, and SCREEN options are used to specify a title, which window should be used for the plot, and whether the screen should be cleared first; as in other graphics directives.

The PEN parameter allows a scalar to specify which pen should be used for the plot; this pen will be used for plotting all parts of the surface – the colour and linestyle settings of the pen will be used but other attributes will be ignored. However, you cannot give a variate of pens to highlight some parts of the plot, as is now possible with DCONTOUR. A key cannot be provided for a surface plot so there is no KEYWINDOW option or DESCRIPTION parameter.

8. Regression Analysis

Most of the improvements in the regression section concern the display of results [8.1] and the provision of new summaries of the results. The latter involve new parameters for the RKEEP directive and a new option for the PREDICT directive [8.2], and a new directive RFUNCTION to allow estimation of functions of parameters from nonlinear models [8.2.1]. The algorithms used in nonlinear modelling have also been improved [8.3].

8.1. Improvements to Output

All directives that can display the fit of a regression model, such as RDISPLAY and FIT, have new options to give extra information. The FPROBABILITY option is like that in the ANOVA and ADISPLAY directives, allowing you to request that F-probabilities be displayed in the tables produced by the option settings PRINT=summary and PRINT=accumulated. The option has no effect if the distribution is not Normal, since deviance ratios are only asymptotically distributed as F-statistics. The variance or deviance ratio for regression is now automatically displayed in the summary table.

The TPROBABILITY option allows you to request that t-probabilities be displayed with the t-statistics produced by the option setting PRINT=estimates. Again, this has no effect for non-Normal distributions.

The RDISPLAY directive has been enhanced in line with the other main display directives in Genstat: the CHANNEL option can now be set to a text identifier. This allows sections of output to be stored in texts, so that they can be edited to satisfy a specific layout requirement, or output later together with other results – particularly in procedures.

When displaying correlation matrices of parameters in nonlinear models, the correlations are now separated from the estimates themselves. The display therefore now looks more like the display following the fit of a linear model.

8.2. Additions to Results that can be Stored

The RKEEP directive is now able to store two extra results. The parameter DESIGNMATRIX will form and store the design matrix as a rectangular matrix following a linear or generalized linear analysis. This is the matrix X in the model equation:

$$y = X\beta + \varepsilon$$

The columns correspond to the parameters in the fitted model, in the same order as the estimates stored by the ESTIMATES parameter. Thus, if a constant term is estimated, the first column will consist of ones; if a factor is in the model, there will be columns of zeros and ones representing each level except the first; and so on.

The second new parameter in RKEEP is PEARSONCHI. It provides the Pearson chi-squared statistic for dispersion in a linear or generalized linear model. For the Normal distribution, this statistic is the same as the deviance: that is, the residual sum of squares.

One other improvement to the RKEEP directive is that all structures that store information about the parameters of a model are now automatically assigned labels that can be displayed during printing, or used within a qualified identifier to simplify the accessing of information for an individual parameter. The structures concerned are those associated with the ESTIMATES, SE, INVERSE, VCOVARIANCE, and DESIGNMATRIX parameters.

The PREDICT directive has a new option to allow formation of predictions on the scale of the linear predictor in a generalized linear model, rather than on the scale of the fitted values. The option setting BACKTRANSFORM=none should be used to specify that the predictions should not be transformed back to the scale of the fitted values using the link function.

8.2.1. The RFUNCTION directive

The new directive RFUNCTION provides estimates of functions of parameters in nonlinear models, together with approximate standard errors and correlations. The directive can be used after any model has been fitted by FITCURVE or FITNONLINEAR. However, if there are any linear parameters in the model for which standard errors have not already been estimated, it is not possible to estimate standard errors or correlations for functions that depend on those parameters. In addition, it is not possible to use the RFUNCTION directive after using the FITCURVE directive with the option NONLINEAR=separate.

The parameter of the RFUNCTION directive must be set to a list of scalars which are to hold the estimated values of the functions. They will be declared implicitly to be scalars if necessary. The CALCULATION option must be set to specify the calculations that form the values of the functions from the values of the parameters in the fitted model. As with the CALCULATION option of FITNONLINEAR, the setting of the CALCULATION option can be either a single expression structure, or a pointer to several expression structures.

The PRINT option controls output, as usual. By default, the estimates of the function values are formed – as could be done simply by a CALCULATE statement using the expressions. In addition, approximate standard errors are calculated, using a first-order approximation based on difference estimates of the derivatives of each function with respect to each parameter. The 'correlations' setting of the PRINT option can be set to request approximate correlations between function values, if there is more than one function.

There are three further options. The SE and VCOVARIANCE options allow the standard errors and the variance-covariance matrix of the functions to be stored in a variate and a symmetric matrix, respectively. (The estimates of the functions themselves are automatically available in the scalars listed in the parameter of the RFUNCTION directive.) The SAVE option specifies which fitted model is to be referred to; it is set in the same way as the SAVE option of RDISPLAY.

When there are linear parameters in the model, not all parameters have explicit names. In this case, there are no named scalars to use in the expressions that form the function values. Instead, the labels of these parameters may be used: but note that the labels must be exactly the same, including case, as those displayed, for example, by the PRINT=estimates option of FITCURVE or FITNONLINEAR. The labels are inserted in the expressions as quoted strings.

8.3. Improvements to Optimization

Both of the algorithms available for fitting nonlinear models have been overhauled for Release 2, to improve performance with a range of test problems – many reported by users of Release 1. It is still possible to construct problems that defeat the algorithms, particularly if they involve inappropriate scaling, unstable parameterization, poor initial values, or discontinuous functions. However, well formulated problems should be solved more reliably in Release 2.

The modified Newton-Raphson algorithm has been extended to remove any formal limit on the number of nonlinear parameters. This means that any function minimization problem can at least be attempted using Genstat – previously, there was a limit of six nonlinear parameters. In addition, there is a new setting of the METHOD option of the RCYCLE directive. If METHOD is set to 'FletcherPowell', each iteration of the search process will consist of one step supplied by the modified Newton-Raphson algorithm, and one step by the Fletcher-Powell conjugate-gradient algorithm. Initial experience suggests that this does not usually improve the search process, except in occasional awkward problems. So the option has been provided to give an extra tool to try on difficult cases.

9. Analysis of Designed Experiments

Several extensions have been made to the directives for analysis of variance in Release 2. Messages are now produced in the output to draw attention to large residuals or to point out the consequences of non-orthogonality; an extra option, NOMESSAGE, has thus been added to the ADISPLAY and ANOVA directives to control which of these are to be printed. The AKEEP directive has also been extended, with four extra parameters to allow information to be saved about contrasts. However, the BLOCKSTRUCTURE, COVARIATE, and TREATMENTSTRUCTURE directives are unchanged.

There are also two improvements that have required no changes of syntax. Firstly, the design structure (as specified by the DESIGN option of ANOVA) has been extended to store the block and treatment formulae used for the analysis. The second, less esoteric, change is that variance ratios are now printed for block terms, provided there is an appropriate term lower in the hierarchy of strata with which to compare them.

9.1. Extensions to the Design Structure

As explained on pages 398-400 of the Genstat 5 Reference Manual, the design structure is a pointer which contains all the information about the design, the model to be fitted, and so forth, necessary to perform a further analysis. In Release 2, the pointer has an additional 10th element to store a copy of the formula specified by the previous BLOCKSTRUCTURE directive (if any), and an 11th element to store the formula specified by the previous TREATMENTSTRUCTURE directive.

9.2. Variance Ratios for Block Terms

In the analysis-of-variance table, variance ratios are now printed for block terms, provided there is an appropriate term lower in the hierarchy of strata with which to compare them. In the split-plot design described in Section 9.2 of the Genstat 5 Reference Manual, Blocks can be compared with Blocks.Wplots, and Blocks.Wplots with Blocks.Wplots.Subplots. However, F-probabilities are not produced for variances ratios of block terms. Also, where there is no single appropriate term for the comparison, no variance ratio is calculated. An example would be given by the block formula for replicated Latin squares (see page 417 of the Genstat 5 Reference Manual)

```
Squares / (Rows * Columns)
```

which expands to

```
Squares + Squares.Rows + Squares.Columns + Squares.Rows.Columns
```

The term Squares could equally well be compared with either Squares.Rows or Squares.Columns. The ratio of most interest would depend on the exact layout of the trial; for example, if the squares were alongside each other, it might be interesting to see whether the squares were more variable that columns within squares. Genstat has no information about layout, and so leaves you to make these comparisons yourself.

9.3. Messages

The analysis-of-variance output now includes messages about large residuals, like those produced with regression (see page 311 of the Genstat 5 Reference Manual). Checking is done for the residuals of every stratum. The messages are part of the information summary, produced by default by ANOVA, but not by ADISPLAY. The other message from ANOVA and ADISPLAY occurs when there is non-orthogonality between treatment terms or between treatments and covariates; the message is a reminder that the sums of squares and estimated effects for each treatment term are for that term eliminating the terms that precede it in the treatment formula and any covariates (see page 445 of the Genstat 5 Reference Manual), and that those for covariates are after eliminating treatments (page 422 of the Genstat 5 Reference Manual). The ANOVA and ADISPLAY directives now have an extra option, NOMESSAGE, to allow messages to be suppressed. The CHANNEL option of ADISPLAY has also been extended to allow output to be saved in a text structure, as mentioned in [2.1].

9.4. Saving of Contrasts

The AKEEP directive has four extra parameters to allow information to be saved about contrasts; the parameters are CONTRASTS, XCONTRASTS, SECONTRASTS, and DFCONTRASTS. For each treatment term there will generally be several contrasts, so the information is stored in pointers with one element for each contrast.

10. Multivariate and Cluster Analysis

The existing directives for multivariate analysis are mainly unchanged in Release 2. The PCP and PCO directives have been extended to accept a rectangular matrix as their input. This is taken to represent a data matrix, where the rows correspond to the units and the columns to variates. The main improvement is that a new directive, MDS, has been introduced for multidimensional scaling [10.1].

10.1. The MDS Directive

The MDS directive carries out iterative scaling, including metric and non-metric scaling. The input data consists of a symmetric matrix whose values may be interpreted, in a general sense, as distances between a set of objects. The matrix is specified by the DATA option; thus only one matrix can be analysed each time the MDS directive is invoked. The objective of the MDS directive is to find a set of coordinates whose interpoint distances match, as closely as possible, those of the input data matrix. When plotted, the coordinates provide a display which can be interpreted in the same way as a map: for example, if points in the display are close together, their distance apart in the data matrix was small. The algorithm invoked by the MDS directive uses the method of steepest descent to guide the algorithm from an initial configuration of points to the final matrix of coordinates that has the minimum stress of all configurations examined.

Printed output is controlled by the PRINT option; by default, nothing is printed. Output includes the solution coordinates, rotated to principal coordinates; the latent roots of the solution coordinates; the inter-unit distances, computed from the solution configuration; the fitted distances; the stress of the solution coordinates; and a summary of the results at each iteration.

The METHOD option determines whether metric or non-metric scaling is used. The algorithm involves regression of the distances, calculated from the solution coordinates, against the dissimilarities in the symmetric matrix specified by the DATA option. With the default setting, METHOD=nonmetric, monotonic regression is used; if METHOD=linear, the algorithm uses linear regression through the origin.

The stress function to be minimized can be selected using the STRESS option. There are three possibilities: ls (least squares), lss (least-squares-squared), and logstress.

The TIES option allows you to vary the way in which tied data values in the input data matrix are to be treated. By default, the treatment of ties is primary, and no restrictions are placed on the distances corresponding to tied dissimilarities in the input data matrix. In the secondary treatment of ties, the distances corresponding to tied dissimilarities are required to be as nearly equal as possible. Kendall (1977) describes a compromise between the primary and secondary approaches to ties: the block of ties corresponding to the smallest dissimilarity are handled by the secondary treatment, the remaining blocks of ties are handled by the primary treatment.

The WEIGHT option can be used to specify a symmetric matrix of weights. Each element of the matrix gives the weight to be attached to the corresponding element of the input data matrix. If the option is not set, the elements of the data matrix are weighted equally. The most important use of the option occurs when the matrix of weights contains only zeros and ones; the zeros then correspond to missing values in the input data matrix, allowing incomplete data matrices to be scaled.

The MAXCYCLE option determines the maximum number of iterations of the algorithm. The default of 30 should usually be sufficient. However, it may be necessary to set a larger value for very large data matrices or when using the logstress setting of the SCALING option. The monitoring setting of the PRINT option may be used to see how convergence is progressing. The NSTARTS option allows you to specify how many starting configurations are to be used, generated by perturbing the initial configuration.

The NDIMENSIONS parameter must be set to a scalar (or scalars) to indicate the number(s) of dimensions in which the multidimensional scaling is to be performed on the data matrix. An MDS statement with a list of scalars will carry out a series of scaling operations, all based on the same matrix of dissimilarities, but with different numbers of dimensions.

The remaining parameters of the MDS directive allow output to be saved in Genstat data structures. These must all have been declared in advance. The COORDINATES parameter can list matrices to store the minimum stress coordinates in each of the dimensions given by the NDIMENSIONS parameter, and the STRESS parameter can specify scalars to store the associated minimum stresses. The parameters DISTANCES and FITTEDDISTANCES can specify symmetric matrices to store the distances computed from the coordinates matrix and the fitted distances computed from the monotonic or linear regressions, respectively.

11. Analysis of Time Series

There are two types of change to the directives for the analysis of time series in Release 2. There are facilities to save more of the information generated, and to make the ESTIMATE directive easier to use for initialization.

11.1. Correlation

The CORRELATE directive has a new option, CORRELATIONS, which allows you to specify a symmetric matrix to store correlations between the variates in the parameter list. Previously, the CORRELATE directive could only print the correlations.

The COEFFICIENTS parameter of the CORRELATE directive now allows you to save the prediction coefficients for all lags up to the maximum lag. You can still save just the coefficients for the maximum lag; this will happen if you give a variate or undeclared structure. You will get the full set of coefficients only if you specify a matrix. The coefficients for any selected lag less than the maximum can then be obtained from this matrix. The matrix is also useful in methods for simulating autoregressive time series.

11.2. ARIMA Modelling

There is a new option for ESTIMATE. Option METHOD has four possible settings. The default setting is 'full', which gives the usual estimation to convergence, or until the maximum number of iterations has been reached. You can specify 'initialize' to tell the program to initialize for forecasting. This replaces the artificial method of setting PRINT=* and MAXCYCLE=0 in order to initialize for forecasting, although this will still work. The other settings, 'zerostep' and 'onestep', both override MAXCYCLE to give zero or one iteration of the estimation process only.

With the setting 'initialize', ESTIMATE merely carries out the residual regeneration steps necessary for the further construction of forecasts. None of the parameters are changed. No standard errors of parameters are available after this step, although the deviance is available. This setting may also be useful for calculating the deviance values efficiently if it is desired to plot the shape of the deviance surface over a grid of parameter values.

With the setting 'zerostep', the same calculations are carried out as for 'initialize', but the standard errors of the parameters are also calculated, and are available using the TKEEP directive. The scores are also available from TKEEP [11.3]. The parameter values remain unchanged except that the innovation variance in the ARIMA model is replaced by its estimate conditional on all other parameters. This setting may be useful for constructing tests of parameter values using the scores and the inverse matrix from TKEEP. For example, if a model has been fully estimated, it is possible to test whether the orders might be increased. This is done by introducing new parameters with their values set to zero, then using ESTIMATE with option METHOD=zerostep.

With the setting 'onestep', just one iteration of the estimation procedure is performed. This may also be useful for testing parameter values. Again if a previously estimated model is extended by increasing its orders and setting the new parameters to zero, the parameter changes after the 'onestep' estimation may be assessed using their standard errors.

The foregoing tests are asymptotically valid as tests of the null hypothesis that the new parameter values are zero. Their value is that they may be used when a relatively large number of new parameters are being assessed without going to full estimation of the extended model.

11.3. Saving Results of ARIMA Modelling

There is a new parameter SCORES for TKEEP. This allows you to save the scores, which are the first derivatives of the deviance with respect to the model parameters, in a variate. These may be used to check on the convergence of the estimation – they should ideally be zero. They may also be used to construct tests of parameter constraints, by using ESTIMATE with the METHOD=zerostep option described above. In this case the scores are evaluated for the parameters given in the time series models supplied to ESTIMATE.

The length of the scores vector is the number of parameters estimated, and the elements correspond to the parameters in the order in which they are printed by ESTIMATE, taking into account equality constraints as indicated by the parameter reference numbers.

12. Extending Genstat

12.1. The Genstat Procedure Library

The new facilities in Release 2, for example for checking settings of options and parameters of procedures automatically [6.1], have resulted in many changes to the procedures in the Library. In most cases, these will not be visible to users – other than that speed of execution will be much improved. However, the ability to manipulate formulae (by the FCLASSIFICATION directive [5.5]) has allowed the syntax of several procedures to be simplified by deducing information, such as the number of model terms in a formula, within the procedure instead of requiring it to be specified by the user. Options of several Library procedures have thus been deleted; details are given in Section 2 of the Appendix.

Some procedures have also been extended. For example, procedures CONCORD, KRUSKAL, and SPEARMANN have a new parameter DF to allow the number of degrees of freedom to be saved for the large-sample test statistics. Further information, including a list of new procedures, can be obtained from within Genstat, by the statement

```
NOTICE [PRINT=library]
```

12.2. Extensions Using Fortran

There have been few changes to the facilities for extending Genstat in Fortran, apart from the new facilities for tabulation described in [5.4]. However, if you have developed a version of either the OWN or the EXTRAD subroutines to work with Release 1.3, it will be necessary to modify it slightly to work with Release 2.1. The only changes necessary are to update the Fortran commons that are included in these subroutines: the new versions can be extracted from the new versions of OWN and EXTRAD that are distributed with Release 2.1.

If you have developed a version of the GNPASS program to work with Release 1.3, this too will require modification. The names of the files that transfer information between Genstat and GNPASS may have changed – depending on the implementation on your operating system. Also, some extra information has been added, both to the Fortran common /DATA/ within GNPASS, and to the transfer files. Therefore, it will be necessary to remake your version of GNPASS, using the new standard version distributed with Release 2.1 – but this should merely mean inserting a call to your own routines in the new standard version.

If you want to write a new application, using either the OWN or the PASS directive, you may now find it easier to see what to do. The example in Chapter 12 of the Genstat 5 Reference Manual is now included with the OWN subroutine and the GNPASS program, showing the Fortran code that is needed for a simple extension.

13. Variance Components Estimation

Release 2 contains new directives for estimating variance components and analysing unbalanced designs. The algorithm is taken from the REML program of the Scottish Agricultural Statistics Service (Robinson, Thompson, and Digby 1982); we are very grateful for their permission to adapt their code for Genstat. The algorithm uses the method of residual (or restricted) maximum likelihood devised by Patterson and Thompson (1971).

The REML method is applicable to a wide variety of situations. It can be used to obtain information on sources and size of variability in experimental material. This can be of interest in its own right, as in animal breeding experiments or where relative sizes of variability need to be assessed in order to design more effective experiments. It can also be used to combine information on treatment effects in experiments where estimates are available in more than one stratum, or over similar experiments conducted at different times or in different places, as with variety trials.

The directives are analagous to those for analysing balanced experiments. The VCOMPONENTS directive specifies the various aspects of the model to be fitted. This has a parameter RANDOM to define the random terms, similarly to the BLOCKSTRUCTURE directive, except that the specified model formula can contain variates as well as factors. Likewise the FIXED option, which is analagous to the TREATMENTSTRUCTURE directive, can also have variates as well as factors. There is thus no option to correspond to the COVARIATES directive; covariates (and their interactions with factors) should be specified in either the FIXED or RANDOM formulae, according to whether or not they are to be regarded as fixed or random terms. The REML directive requests the analysis of one or more y-variates, allowing residuals, fitted values, and save structures to be saved, as with the ANOVA directive for balanced experiments. Further output can be displayed using the VDISPLAY directive, and components of output can be placed into Genstat data structures using VKEEP.

13.1. The VCOMPONENTS Directive

The VCOMPONENTS directive specifies the variance-components model to be fitted by subsequent REML statements.

The FIXED option specifies a model formula to define the parameters for the fixed model. The rules for this formula are similar to those for the TREATMENTSTRUCTURE directive, except that there is no need to use pseudo-factors with REML and that you can also include covariates and their interactions. It is possible at this stage to omit the constant term from the fixed model by putting option CONSTANT=omit; the default setting, CONSTANT=estimate, automatically includes the constant term.

The random part of the model is defined by the formula specified using the RANDOM parameter. As the error variance is considered separately from the rest of the random part of the model, there is no need to define the bottom stratum explicitly, as it is included automatically. So for a completely randomized design, where the only variance component is σ^2, there is no need to specify a RANDOM model. Again, the rules for the RANDOM model are similar to those for the BLOCKSTRUCTURE directive, except that covariates and their interactions are allowed in the formula. However, random covariates do not seem to occur very frequently.

The estimation of the variance components is carried out in terms of the gamma parameters, which represent variance ratios. If an approximate value of the ratio is known for any of the components, then it is wise to use this as a starting point in order to save computing time. This is done by listing a set of initial gamma values with the INITIAL parameter. If a gamma is known well enough for there to be no point in further estimation, it can be constrained to be fixed to its initial value. Constraints are given in parallel to initial values using parameter CONSTRAINTS. Putting CONSTRAINT=fixed fixes the gamma to its initial value, which is 1 if no other initial value is given; by default CONSTRAINT=positive, so the gamma is estimated subject only to the constraint that it remains positive.

The other use of the VCOMPONENTS directive is to specify an absorbing factor. This factor is used internally to save computing time and space by splitting the FIXED and RANDOM models into two parts: those terms that contain the absorbing factor, and those that do not. The parameters in the part not containing the absorbing factor are estimated as usual, while the parameters in the other part (that is, the absorbing factor model) are estimated sequentially for each level of the absorbing factor in turn. This reduces the space required for this part of the model by an order given by the number of levels of the absorbing factor; it can also reduce computing time, by decreasing the order of the matrices that have to be inverted at each iteration. However, the information needed to calculate standard errors for estimates from the absorbing factor modèl is not stored. So a good choice of absorbing factor is either a factor with a large number of levels, or one where you are not interested in estimates of its effects.

13.2. The REML Directive

Once you have defined a variance-components model using VCOMPONENTS, you can then analyse the variates containing the data (the *y*-variates) using REML.

You can use the FACTORIAL option to set a limit on the number of factors and variates allowed in each fixed term; any term containing more than that number is deleted from the model.

Options WEIGHTS, MVINCLUDE, TOLERANCES, and MAXCYCLE all control aspects of the analysis. You can use the WEIGHTS option to specify a weight for each unit in the analysis. The MVINCLUDE option allows you to include units in the analysis that would normally be excluded. Units for which the *y*-variate has a missing value will always be excluded. However, setting MVINCLUDE=yes allows you to include units for which there are missing values in the factors or variates defining the model terms. The TOLERANCES option controls the tolerances for matrix inversion, and you can change the maximum number of iterations from the default of 10 using the MAXCYCLE option.

The three remaining options PRINT, PTERMS, and PSE all control the printed output. The PRINT option selects the output to be displayed: estimates of variance components; tables of effects; tables of means; approximate stratum variances; monitoring information at each iteration; the variance-covariance matrix of the estimated components; and the deviance of the fitted model. The default of PRINT=c,e gives the estimates of the variance components and the approximate stratum variances. Options PTERMS and PSE control the tables of means and effects that are to be printed, and their accompanying standard errors.

The first parameter of REML, Y, lists the variates that are to be analysed. If any of these *y*-variates is restricted, then the restrictions must all be the same, and the analysis will use only those units not excluded by the restriction.

The parameters FITTEDVALUES and RESIDUALS allow you to store the fitted values and residuals. Parameter SAVE can be used to name the REML save structure for use with later VKEEP and VDISPLAY directives.

13.3. The VDISPLAY Directive

The VDISPLAY directive allows further output to be produced from one or more REML analyses without having to repeat all the calculations. You can store the information from a REML analysis using the parameter SAVE in the REML statement, and then specify the same structure with the SAVE parameter of VDISPLAY. Several SAVE structures can be specified, corresponding to the analyses of several different variates. These need not have been analysed using the same REML statement, or even from the same model (as defined by VCOMPONENTS). Alternatively, if you just want to display output from the latest *y*-variate that was analysed, then there is no need to use the SAVE parameter in either REML or VDISPLAY: the save structure for the latest *y*-variate analysed is saved automatically, and provides the default for VDISPLAY.

The options of VDISPLAY are the same as those that control output from REML: PRINT, PTERMS, and PSE.

13.4. The VKEEP Directive

VKEEP allows you to copy information from a REML analysis into Genstat data structures. As for VDISPLAY, you can save the information from a REML analysis in a save structure using the SAVE parameter in the REML statement, then access the information by specifying the same structure in the SAVE option of VKEEP. Alternatively, Genstat automatically stores the save structure for the latest *y*-variate that was analysed using REML, and this save structure provides the default for VKEEP if no other save structure is specified.

Overall information from the analysis is saved using the options of VDISPLAY, while the parameters allow you to save information for specific model terms similarly to AKEEP. The terms (fixed or random or a mixture) for which you require information are defined by the formula specified with the TERMS parameter. The other parameters can then be used to specify a series of structures, running in parallel with the model terms, for saving information.

The information that can be saved by the options is fairly self-explanatory. The options RESIDUALS and FITTEDVALUES can specify variates to save the residuals and fitted values. The residual variance can be saved in a scalar using SIGMA2. The VCOVARIANCE option can specify a symmetric matrix to save the variance-covariance matrix for the estimates of variance components. Finally, the FULLVCOVARIANCE option can be used to store the variance-covariance matrix for the full set of fixed and random effects, excluding those in the absorbing factor model. This matrix will often be very large, and is useful only for looking at covariances between effects associated with different model terms; you can save variance-covariance matrices for individual model terms using the EFFECTS parameter.

The formula given in the TERMS parameter will be interpreted as explained in Section 9.1.1 of the Genstat 5 Reference Manual. The other parameters of VKEEP are taken in parallel with the set of expanded model terms.

The COMPONENTS parameter allows you to save the estimated variance component for each random term in the model. If you try to save a variance component for a term in the fixed model, it will be given a missing value.

The EFFECTS parameter is used to save tables of effects, as described for VDISPLAY. A symmetric matrix of the standard errors of differences between the effects of each term can be saved using parameter SEDEFFECTS.

Tables of means (as for VDISPLAY) are saved by the MEANS parameter, and standard errors of differences between the means of each term can be saved by SEDMEANS. You can also save the variance covariance matrix for the means of each term using VARMEANS.

Appendix: Incompatibilities with Release 1

Release 2.1 is fully compatible with Release 1.3 except for the changes listed below.

A.1. New Options and Parameters for Existing Directives

In five directives, extra options or parameters have been added between existing options or parameters. This will cause problems only if you set options or parameters without specifying their names.

Directive	New options	Position	Next option
FITNONLINEAR	FPROBABILITY	9	NGRIDLINES
PREDICT	BACKTRANSFORM	8	PREDICTIONS
RDISPLAY	FPROBABILITY; TPROBABILITY	5; 6	SAVE
STEP	FPROBABILITY; TPROBABILITY	6; 7	INRATIO

Directive	New parameter	Position	Next parameter
OPTION	NVALUES; VALUES	3; 4	DEFAULT

A.2. Changed Options in Library Procedures

Use of the FCLASSIFICATION directive has allowed the syntax of the following procedures to be simplified by deducing information, such as the number of model terms in a formula, within the procedure instead of requiring it to be specified by the user. The deleted options are as follows:

Procedure	Options
ANTORDER	FINALSTRATUM
ANTTEST	NTREATTERMS; FINALSTRATUM
MANCOVA	NTREATTERMS; BLOCKTERMS
MANOVA	NTREATTERMS; BLOCKTERMS

A.3. Modification to Action of Directives

The changes to the COPY directive may cause existing programs to behave differently with Release 2.1 than with 1.3. In Release 1.3, only one channel could be used at a time for keeping a transcript, which could be either of the statements or of the output or both. In Release 2.1, it is possible to keep a transcript of the statements in one file, and of the output in another. Thus the following statements behave differently:

```
OPEN 'FILE1.REC','FILE2.REC'; CHANNEL=2,3; FILETYPE=output
COPY [PRINT=statements] 2
COPY [PRINT=output] 3
```

In Release 1.3, FILE1 would receive a copy of the second COPY statement only. In Release 2.1, FILE2 would receive copies of all following statements as well, until another COPY statement redirects the transcripting of statements.

The interpretation of suffixed identifiers has been slightly modified in Release 2.1. This is only likely to affect complicated programs and procedures. In Release 1.3, the statement

```
SCALAR A,B,C; VALUE=11,12,13
VARIATE [VALUE=2] Single
POINTER [VALUES=A,B,C] P
PRINT P[Single]
```

would print the scalar B, with value 12.00, exactly as with the statement

```
PRINT P[2]
```

By contrast, the statements

```
VARIATE [VALUES=2,3] Double
PRINT P[Double]
```

printed the sub-pointer with values B and C. In Release 2.1, the first PRINT statement above would print the sub-pointer with value B. In both releases, the statement

```
PRINT P[2,3]
```

would print the scalars B and C.

References

Patterson. H.D. and Thompson, R.
Recovery of inter-block information when block sizes are unequal.
Biometrika, **58**, pp. 545-554, 1971.

Kendall, M.G.
On the tertiary treatment of ties.
Proc R. Soc Lond. A, **354**, pp. 407-423, 1977.

Robinson, D.L., Thompson, R. and Digby, P.G.N.
'REML – a program for the analysis of non-orthogonal data by restricted maximum likelihood.'
In: COMPSTAT 1982, part II (supplement), pp. 231-232.
Wien: Physica-Verlag, 1982.

Payne, R.W., Lane, P.W., Ainsley, A.E., Bicknell, K.E., Digby, P.G.N., Harding, S.A., Leech, P.K., Simpson, H.R., Todd, A.D., Verrier, P.J., White, R.P., Gower, J.C., Tunnicliffe-Wilson, G. and Paterson, L.J.
Genstat 5 Reference Manual.
Oxford University Press, 1987.

The Genstat Menu System

Peter Lane
Statistics Department
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Herts AL5 5QU

This article summarizes information about the Menu System which has been presented at two recent Genstat conferences. A skeleton version of the System was described and demonstrated at the sixth Genstat Conference in Edinburgh, in September 1989. After incorporating a number of suggestions made at the conference, a virtually complete version was demonstrated at the Genstat One-day Conference on 'Interactive Statistical Modelling' at Rothamsted, in April 1990. This article is a modified version of Section 2.4 of the Genstat 5 Release 2 Manual Supplement, published by NAG. The material is included in this Newsletter to help disseminate information about the System to users as quickly as possible, because of its introduction of a wholly new mode of using Genstat.

The Genstat Menu System is designed to provide access to some of Genstat's standard facilities without the need to learn the command language. An illustrative session is given in Section 1. The System cannot be used with versions before Release 2, because it makes use of the new QUESTION directive. This directive, described in Sections 3 and 4, has been designed primarily to allow experienced users to construct their own menu systems, either for their own use, or for that of their colleagues or clients. In this context, a menu system consists of one or more files of Genstat statements, likely to include many QUESTION statements, that ask the user for information and carry out work from a specific repertoire. The standard System, which covers many common methods of analysis, is described in Section 4, whereas the method of setting up alternative systems is described in Section 5. The files that make up the System can be copied and modified as required, or translated into other languages, or just referred to as examples when constructing more specialized or more complex systems.

1. An Illustrative Session Using the Genstat Menu System

The following example shows the use of the Menu System to analyse a balanced experiment. To save space, some of the repeated material has been omitted and comments are given instead. Also, no use has been made of the special response codes, which are described in Section 3. The answers to the questions are printed below in bold type to distinguish these from the text displayed by Genstat itself.

```
> GENSTAT

Genstat 5  Release 2.1  (Vax/VMS5)                12-FEB-1990 16:09:16.89
Copyright 1990, Lawes Agricultural Trust (Rothamsted Experimental Station)

**************************************************************************
* You can use Genstat interactively in command-mode or in menu-mode.     *
* You are now in command-mode:                                           *
*    type HELP for on-line help about the command language of Genstat;   *
*    type STOP to finish;                                                *
*    type MENU to enter menu-mode - Genstat will prompt for information. *
* The standard menu system covers some of the standard analyses that can *
* be done in command-mode, and is designed so that you can extend it.    *
**************************************************************************

> MENU

 Welcome to Release 2.0[2] of the Genstat menu system.
 The system will use three files to record the work you do:
       G21RESLT:  results produced by the operations you choose;
       G21COMND:  copy of Genstat commands that do the operations;
       G21STORE:  quick-access binary storage of your data.
```

```
BASE menu
What would you like to do next ?
          (null) chooses default response, if any;                  |
?         help;  ?? lists current structures of allowed type;       |  Special
                 ?code gives specific help, if any;                 |  responses
&         repeat the question;                                      |  to any
<         return to previous layer of menus;                        |  question
>>        exit to command mode.                                     |

i         input data
c         calculate new data, or define groups
t         display or summarize data in tables
p         display data in pictures
a         analyse data by standard statistical methods
q         quit using menu system

Code (i,c,t,p,a,q; Default:  i) > i


INPUT menu

Where are the data values ?

b         in a binary file previously set up using Genstat
s         in a character-type file, with values separated by spaces
t         to be typed at the terminal

Code (b,s,t; Default:  b) > s


FILENAME menu (from INPUT menu)

What is the file called ?
   Enclose filename in single quotes ('); or just press RETURN
   to enter operating system temporarily to find filename.

Identifier (Default: enter operating-system) > 'anova_3frb.dat'


END-OF-DATA menu (from INPUT menu)

How is the end of data marked ?

c         Colon (:) after the last value
e         Data continue to the end of the file
s         Special symbol (to be specified later) after the last value
n         No marker: read a given number of values (to be specified later)

Code (c,e,s,n; Default:  c) > c


The first two records of the file are:
1 1 1  0  1
1 1 2  5  1


IDENTIFIER menu (from INPUT menu)

What identifiers will you use for the variables to be input ?
   If more than one, separate the identifiers with commas;
   end a line with backslash (\) to continue on the next line.

Identifiers > n,p,k,y,block


STRUCTURE-TYPE menu (from INPUT menu)

What type of variable is  n

v         Variate: for quantitative measurements (integers or real numbers)
t         Text: for textual data (simple or quoted strings of characters)
f         Factor: for group codes (numbers, to be supplied)

Code (v,t,f; Default:  v) > f


LEVELS menu (from INPUT menu)

What are the numeric level-codes for  n
```

```
   If more than one, separate the numbers with commas;
   end a line with backslash (\) to continue on the next line.

Numbers > 1,2,3


(Comment: STRUCTURE-TYPE and LEVELS menus for p and k are as for n
          and so are omitted here)


STRUCTURE-TYPE menu (from INPUT menu)

What type of variable is  y

v         Variate: for quantitative measurements (integers or real numbers)
t         Text: for textual data (simple or quoted strings of characters)
f         Factor: for group codes (numbers, to be supplied)

Code (v,t,f; Default:  v) > v


STRUCTURE-TYPE menu (from INPUT menu)

What type of variable is  block

v         Variate: for quantitative measurements (integers or real numbers)
t         Text: for textual data (simple or quoted strings of characters)
f         Factor: for group codes (numbers, to be supplied)

Code (v,t,f; Default:  v) > f


LEVELS menu (from INPUT menu)

What are the numeric level-codes for  block
   If more than one, separate the numbers with commas;
   end a line with backslash (\) to continue on the next line.


Numbers > 1,2,3,4


    Identifier   Minimum      Mean   Maximum    Values   Missing
             y      0.00     38.85    109.00       108         0


STORE menu (from INPUT menu)

Do you want to store the data in the binary file:
G21STORE
for later access ?

y         Yes
n         No

Code (y,n; Default:  y) > y


***** catalogue *****

catalogue of structures in the subfile SUBFILE

    entry  identifier     type        points to

    1      n              factor      6
    2      p              factor      7
    3      k              factor      8
    4      y              variate
    5      block          factor      9
    6                     variate
    7                     variate
    8                     variate
    9                     variate

 *Press RETURN to continue*
```

```
BASE menu

(Comment: BASE menu as above, so omitted here)

Code (i,c,t,p,a,q; Default:  i) > a


ANALYSIS menu (from BASE menu)

What standard method of analysis do you want ?

r          regression analysis, linear or nonlinear
a          analysis of designed experiments
ms         multivariate analysis based on sums of squares and products
md         multivariate analysis based on distances
t          analysis of time series
h          tests to compare two samples (t-tests or non-parametric tests)

Code (r,a,ms,md,t,h; Default:  r) > a


ANOVA Menu

What kind of experimental design have you used ?

one        Completely randomized one-way design
cr         Completely randomized design
rb         Randomized block design
sp         Split-plot design
latin      Latin square design
lattice    Lattice design
bib        Balanced incomplete-block design

Code (one,cr,rb,sp,latin,lattice,bib; Default:  one) > rb


Y-VARIATE Menu (from ANOVA Menu)

What is the variate to be analysed ?

Identifier > y


BLOCKS Menu (from ANOVA Menu)

What factor specifies which unit is in which block ?

Identifier > block


TREATMENTS Menu (from ANOVA Menu)

What factors specify which unit received which treatments ?
   If more than one, separate the identifiers with commas;
   end a line with backslash (\) to continue on the next line.

Identifiers > n,p,k


INTERACTION Menu (from ANOVA Menu)

What interactions do you want to estimate ?

m          main effects only, i.e.: A + B + C + ... + N
m2         main effects and two-factor interactions
m23        main effects, two-factor interactions, and three-factor interactions
a          all interactions that can be estimated

Code (m,m2,m23,a; Default:  m) > a


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

All terms orthogonal, none aliased.
```

```
* MESSAGE: the following units have large residuals.

block 1.00  *units* 2           3.41   s.e. 1.20
block 1.00  *units* 4           3.41   s.e. 1.20
block 1.00  *units* 5           3.41   s.e. 1.20
block 1.00  *units* 6           3.41   s.e. 1.20
block 4.00  *units* 2          -4.26   s.e. 1.20
block 4.00  *units* 4          -4.26   s.e. 1.20
block 4.00  *units* 5          -4.26   s.e. 1.20
block 4.00  *units* 6          -4.26   s.e. 1.20


DISPLAY Menu (from ANOVA Menu)

What results would you like to see ?
a         Analysis of Variance Table
i         Information summary
e         Effects: tables of estimated treatment parameters
r         Tables of estimated residuals
m         Tables of predicted means for treatment terms
%         Coefficients of variation and standard error of individual units
mv        Estimates of missing values
q         Quit to Base Menu

Code (a,i,e,r,m,%,mv,q; Default:  m) > a


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

Variate: y

Source of variation     d.f.       s.s.       m.s.    v.r.  F pr.

block stratum             3    901.259    300.420  149.50

block.*Units* stratum
n                         2  32373.629  16186.814 8055.16  <.001
p                         2  17422.520   8711.260 4335.05  <.001
k                         2  20541.631  10270.815 5111.14  <.001
n.p                       4   1982.815    495.704  246.68  <.001
n.k                       4   3341.038    835.259  415.66  <.001
p.k                       4   1945.482    486.370  242.04  <.001
n.p.k                     8    406.519     50.815   25.29  <.001
Residual                 78    156.741      2.009

Total                   107  79071.633

 *Press RETURN to continue*


DISPLAY Menu (from ANOVA Menu)

(Comment: DISPLAY menu as above, so omitted here)

Code (a,i,e,r,m,%,mv,q; Default:  m) > m


***** Tables of means *****

Variate: y

Grand mean  38.85

        n     1.00     2.00     3.00
             15.22    45.11    56.22

        p     1.00     2.00     3.00
             24.11    37.33    55.11

        k     1.00     2.00     3.00
             22.11    38.56    55.89

        n        p     1.00     2.00     3.00
     1.00              7.67    14.00    24.00
     2.00             27.33    44.00    64.00
     3.00             37.33    54.00    77.33

 *Press RETURN to continue*
```

```
        n        k    1.00      2.00      3.00
     1.00             8.33     14.33     23.00
     2.00            24.00     44.00     67.33
     3.00        .   34.00     57.33     77.33

        p        k    1.00      2.00      3.00
     1.00            14.00     21.00     37.33
     2.00            21.67     37.33     53.00
     3.00            30.67     57.33     77.33

        n        p        k    1.00      2.00      3.00
     1.00     1.00             4.00      5.00     14.00
              2.00             7.00     14.00     21.00
              3.00            14.00     24.00     34.00
     2.00     1.00            14.00     24.00     44.00
              2.00            24.00     44.00     64.00
              3.00            34.00     64.00     94.00
     3.00     1.00            24.00     34.00     54.00
              2.00            34.00     54.00     74.00
              3.00            44.00     84.00    104.00

 *Press RETURN to continue*

*** Standard errors of differences of means ***

Table                   n           p           k           n
                                                            p
rep.                   36          36          36          12
s.e.d.              0.334       0.334       0.334       0.579

Table                   n           p           n
                        k           k           p
                                                k
rep.                   12          12           4
s.e.d.              0.579       0.579       1.002

 *Press RETURN to continue*

DISPLAY Menu (from ANOVA Menu)

(Comment: DISPLAY menu as above, so omitted here)

Code (a,i,e,r,m,%,mv,q; Default:  m) > q


BASE menu

(Comment: BASE menu as above, so omitted here)

Code (i,c,t,p,a,q; Default:  i) > q


Quit from menu system.
   Type STOP to leave Genstat;
   type MENU to re-enter menu system;
   or type Genstat commands.

> STOP
```

Following this session, three standard files are available from the operating system. G21STORE is a Genstat backing-store file that contains the data used in the analysis. It will automatically be available in the next session, if the defaults of the INPUT menus are chosen.

G21RESLT contains a copy of all the reports produced during the session; that is, all the output in the above example excluding the menus themselves.

G21COMND contains Genstat statements that would have produced the results. The intention of this is two-fold: to record the work that has been done, and to help people if they want to learn the command language. Here are the contents of the file formed during the above session.

```
MENU
OPEN 'G21STORE'; CHANNEL=5; FILETYPE=backing
OPEN 'anova_3frb.dat'; CHANNEL=2
READ [SETNVALUES=yes] _levels
1.000
2.000
3.000
:
FACTOR [LEVELS=!(#_levels)] n
READ [SETNVALUES=yes] _levels
1.000
2.000
3.000
:
FACTOR [LEVELS=!(#_levels)] p
READ [SETNVALUES=yes] _levels
1.000
2.000
3.000
:
FACTOR [LEVELS=!(#_levels)] k
VARIATE y
READ [SETNVALUES=yes] _levels
1.000
2.000
3.000
4.000
:
FACTOR [LEVELS=!(#_levels)] block
POINTER [NVALUES=5] _data
READ _data
    n
    p
    k
    y
block
:
READ [CHANNEL=2; SETNVALUES=yes; END=':'] _data[]
BSUPDATE _data
BLOCKS block
TREATMENTS n*p*k
ANOVA [PRINT=*; FACTORIAL=9] y
ADISPLAY [PRINT=inform; FPROB=yes]
ADISPLAY [PRINT=aovtable; FPROB=yes]
ADISPLAY [PRINT=means; FPROB=yes]
STOP
```

The BSUPDATE procedure is in the Genstat Procedure Library, and so is available to all Genstat jobs. It allows structures to be added to an existing subfile of a Genstat backing-store file.

2. The QUESTION directive

The QUESTION directive displays a Genstat menu and obtains a response in interactive mode. In batch mode, the directive does nothing. Here is a simple example that asks the user to provide the identifier of a variate, as used in the example above to request the variate to analyse.

```
QUESTION [PREAMBLE=!t('Y-VARIATE Menu (from ANOVA Menu)',*, \
  'What is the variate to be analysed ?'; RESPONSE=_yvar; \
  DECLARED=yes; TYPE=variate; PRESENT=yes]
```

This statement displays the following Genstat menu:

```
Y-VARIATE Menu (from ANOVA Menu)

What is the variate to be analysed ?

Identifier >
```

The PREAMBLE option specifies a text structure, whose contents are printed at the beginning of the menu. Following this is the prompt: by default, this consists of a reminder of what type of answer is expected, followed by the greater-than symbol (>). However, there is a PROMPT option that allows any text to be printed instead, before the greater-than symbol.

The RESPONSE option specifies a dummy identifier that will point to the answer given by the user. Note that the identifiers used in the standard System all begin with the underline symbol (_) to reduce the chance of a clash with the user's own identifiers. Menus can request information in one of five modes; the default is Mode 'p' (pointer), as here, and expects a response to consist of an identifier; but the MODE option can also be set to 'v' (variate), 't' (text), 'e' (expression), or 'f' (formula). When a correct answer has been received, an unnamed structure of the relevant type (pointer, variate, or whatever, but see later for text mode) is set up, and the dummy in the RESPONSE option is set to point at this unnamed structure.

Thus, if the user gives the identifier Y in response to the above question, the dummy _yvar will store the identifier of a pointer containing the single identifier Y after the QUESTION statement above has been executed. So in the standard System, the statement following this QUESTION statement is

```
    ANOVA #_yvar
```

the hash (#) being needed to substitute the values of the unnamed pointer that is stored in the dummy structure _yvar. This may seem unnecessarily complex for a simple question expecting a single answer, but it deals with multiple answers as well, and has proved to be very convenient in practice when constructing the standard System.

By default, a question will expect to receive a single item of the specified mode: identifier, number, string, expression, or formula. However, if the option LIST is set to 'yes' for modes 'p', 'v', or 't', then a list of items is expected. The unnamed structure set up to store the answer will then contain as many values as there are items in the list.

The other three options in the example above specify restrictions on the answer that will be accepted. The DECLARED option specifies that the identifier must be of a structure that has already been declared. If a previously unused identifier is given, the QUESTION statement will print a warning, and issue the prompt again. Similarly, the TYPE option specifies what type of structure is acceptable; the setting may be a list of types that are to be allowed. The PRESENT option specifies that the structure must already have values. Two further options, LOWER and UPPER, can be used to specify limits for numbers given to questions of mode 'v'.

Most menus in the standard System are of mode 't', and resemble more closely what most people think of as a menu than does the simple display above. Such menus require extra information to be specified using parameters of the QUESTION directive. The VALUES parameter should be set to a list of text structures, each of which stores a single string, usually a simple letter-code, that is to be accepted as an answer to the question. The CHOICE parameter should be set to another list of text structures, each storing a single string that is to be displayed by the side of the corresponding code in the menu to explain it. Here is an example, again taken from the standard System, showing first the statement and then its output.

```
   QUESTION [PREAMBLE=!t('INPUT menu',*,'Where are the data values ?'); \
     RESPONSE=_cdsourc; MODE=t; DEFAULT='b'] VALUES='b','s','t'; CHOICE= \
     'in a binary file previously set up using Genstat', \
     'in a character-type file, with values separated by spaces', \
     'to be typed at the terminal'

INPUT menu

Where are the data values ?

b         in a binary file previously set up using Genstat
s         in a character-type file, with values separated by spaces
t         to be typed at the terminal

Code (b,s,t; Default:  b) >
```

The codes must obey the rules for unquoted strings (Genstat Reference Manual, page 11): that is, they must start with a letter and consist of letters and digits only. Only the first eight characters will be displayed, and only the first eight characters of the answer will be checked – all eight must match. Usually, of course, it is convenient to use single-letter codes.

Note that mode 't' cannot be used to ask the user for an arbitrary string, for example to use as a label for output. To request such information, you must use mode 'p', and set TYPE=text; the user must then supply the string in quotes, or supply the identifier of a text structure that already stores the string.

The response to a question of mode 't' is stored not as a text, but as a variate, each value being the number of the corresponding code as listed in the VALUES parameter. Usually, of course, a menu of mode 't' will be set with LIST=no, the default, and so the variate will contain only a single number. This can be used to control subsequent action in the menu system, in particular in conjunction with a CASE statement. For example, the statements in the standard System following the above QUESTION statement could have the following structure.

```
CASE _cdsourc
  " Statements to deal with code 'b' "
OR
  " Statements to deal with code 's' "
OR
  " Statements to deal with code 't' "
ENDCASE
```

The DEFAULT option is used here to specify a default answer if the user just types RETURN (see Section 3); it can be set for any mode of question. The HELP option and parameter of the QUESTION directive allows you to provide help text to guide the person answering the question; these are described in Section 3. The SAVE option allows you to declare a menu without necessarily issuing it, and also to issue a menu that has already been declared.

3. Special Responses Allowed by the QUESTION Directive

There are a number of special responses that can be given to any question posed by a QUESTION statement. These are provided to allow extra information to be passed to the user before answering the question, and to allow escape from a question when it cannot be answered for some reason. Here is a list of the special responses, as listed in the preamble of the BASE menu in the standard System.

```
         (null) chooses default response, if any;          |
?        help;   ?? lists current structures of allowed type; |  Special
                 ?code gives specific help, if any;        |  responses
&        repeat the question;                              |  to any
<        return to previous layer of menus;                |  question
>>       exit to command mode.                             |
```

If no answer is given – that is, the user just types RETURN – then this is deemed to accept the default response if one has been specified by the DEFAULT option. If DEFAULT is not set, the user will be told there is no default and be prompted again for an answer.

The query character (?) can be used to ask for help. If the user types a single query character, then the text specified in the HELP option of the QUESTION statement will be displayed. If the text is long, it will automatically be interrupted by pauses as if the PAUSE option of the SET directive had been set. If the HELP option was not set, the user will be informed that there is no help available.

For menus of mode 't', the user can respond with a query followed by one of the allowed codes. This will display the help-text associated with that code in the setting of the HELP parameter, if any.

The double query response (??) can be used for menus of mode 'p'. It will cause a dump (as produced by the DUMP directive) showing all current structures in the job of the correct type (according to the setting of the TYPE option). Thus the user can be reminded of an identifier that may have been defined much earlier in a session. The dump is likely to include unnamed structures, and possibly named structures that have been set up internally by the menu system. However, it should be easy to pick out the names of the user's own structures if a simple convention is used in the naming of structures within the menu system.

The ampersand (&) can be given to redisplay a question. This is particularly useful after reading help information, when the original question may well have disappeared from a terminal screen.

The two remaining codes use the less-than and greater-than symbols to cause the current question to be abandoned. Less-than (<) has the same meaning as in the HELP directive, and causes abandonment of the current question and return to the previous layer of the menu system. The precise result of this will depend on how the system has been set up; however, in the standard System, if less-than is typed in answer to the BASE Menu, it will cause exit from the Menu System. If it is typed in response to any other menu shown in the above example, then it will cause return to the BASE Menu. This facility allows a user to make mistakes in moving around the System. For example, if the ANOVA Menu is chosen without entering any data first, it will not be possible to answer the Y-VARIATE Menu with the identifier of a variate. By typing less-than, the user can return to BASE and then select the INPUT Menu to define a variate.

The double-greater-than response (>>) causes exit from the menu system altogether, putting the session into command mode. Someone familiar with the command language can then continue the session, using all the structures that have been set up by previous menus. The system can be reentered at any time, in the same way as it was first started: by the statement

```
MENU
```

in the case of the standard System.

4. The Repertoire of the Genstat Menu System

The Genstat Menu System consists of eleven files of Genstat statements, together with some definitions in the standard Start-up File, plus the Library procedure MENU which enters the System. When the statement

```
MENU
```

is given, the procedure changes input channel to the file G2MNBASE. This contains the QUESTION statement that displays the BASE menu. There is than a CASE statement which results in a further change of input channel to one of the files G2MNINPU (input), G2MNCALC (calculations), G2MNTABU (tabulation), or G2MNPICT (pictures), or poses a further question to discover what type of statistical analysis is required. The latter question can change input channel to one of the files G2MNREGR (regression), G2MNANOV (analysis of designed experiments), G2MNMULS (multivariate analysis based on SSPs), G2MNMULD (multivariate analysis base on distances), G2MNTIME (time series), or G2MNTEST (simple statistical tests).

The definitions in the Start-up File are designed to allow modifications – both for extension to the System, and for changes necessary for some implementations of Genstat. For example, the filenames used in the Genstat OPEN statements that pass control to the various menu files above are intended for operating systems like Vax/VMS that have the concept of a logical or environmental name. In operating systems without this concept, the Start-up File can be amended to supply specific names for the files, to be used throughout the menu system.

To give some indication of the scope of the Genstat Menu System, the choices available in the main menus in all ten files (other than the BASE Menu File) are listed below.

```
INPUT menu

Where are the data values ?

b         in a binary file previously set up using Genstat
s         in a character-type file, with values separated by spaces
t         to be typed at the terminal

CALCULATIONS menu

What type of calculations do you want do do ?

t         Transformation of all numbers stored in a variate
e         Edit individual values of a variate
f         Form groups from the numbers in a variate or the strings in a text
g         General calculation, using expression involving existing data

TABULATION menu

How do you want the data to be displayed ?

l         List the values of one or more variates, texts and factors
s         Summary statistics of the values in a variate
g         Grouped summary of a variate, for groups defined by factors

PICTURE menu

What type of picture do you want ?

g         Graph of one variate against another
h         Histogram of the values in a variate
```

```
REGRESSION menu

What kind of regression model do you want to fit ?
   All (except 1) can be fitted with or without one grouping factor

s          Simple linear regression (one explanatory variable)
m          Multiple linear regression (several explanatory variables)
l          Log-linear analysis of a contingency table (classified counts)
q          Probit or logit analysis of quantal data (counts of successes)
c          Standard nonlinear curve (one explanatory variable)

ANOVA Menu

What kind of experimental design have you used ?

one        Completely randomized one-way design
cr         Completely randomized design
rb         Randomized block design
sp         Split-plot design
latin      Latin square design
lattice    Lattice design
bib        Balanced incomplete-block design

MULTIVARIATE (SSPM) Menu

What kind of multivariate analysis do you need ?

p          Principal component analysis (PCP)
c          Canonical variate analysis (CVA)
f          Factor rotation (after PCP or CVA)
q          Quit to BASE menu

MULTIVARIATE (DISTANCES) Menu

What kind of multivariate analysis do you need ?

f          Forms a similarity matrix
p          Principal coordinate analysis (after forming similarities)
h          Hierarchical cluster analysis (after forming similarities)
q          Quit to BASE menu

TIME-SERIES menu

What operation would you like to perform on a single time series ?

d          Display statistics to help with ARIMA model selection
e          Estimate parameters, fitting an ARIMA model to a time series
f          Forecast, based on a previously fitted ARIMA model
q          Quit to BASE menu

TEST menu

What type of test do you want to carry out ?

t          T-test of the difference between the means of two samples
pt         Paired t-test of the difference between the means of two samples
m          Mann-Whitney U-test of location difference between two samples
w          Wilcoxon matched-pairs test of location difference between two samples
k          Kolmogorov-Smirnov test of difference in distribution of two samples
```

5. Setting Up a Menu System

The provision of the Genstat Menu System involves no additional overheads to the command-mode operation of Genstat beyond the extra Fortran source code required to interpret the QUESTION directive. Extension of the system, or addition of alternative systems, will have no effect on the operation of Genstat in command mode. The only overhead is the time spent in designing the extensions or additions, and the storage-space required for the resulting files of Genstat statements.

To extend the standard System, someone familiar with the command language and this document should have no difficulty in modifying the existing QUESTION statements to provide extra branches. For example, consider the task of adding the ability to produce a PostScript graphics file representing a high-resolution graph of one variate against another. The PICTURE menu in the file G2MNPICT needs to be given one extra text in the VALUE parameter, say 'p', and one extra text in the CHOICE parameter, say 'PostScript high-resolution graph of one variate against another'. Then the CASE structure needs to be extended with an additional OR statement and the statements to draw the graph. Here is what it could look like (excluding help-text), assuming that high-resolution graphics have been implemented with Genstat Device 3 corresponding to PostScript.

```
QUESTION [PREAMBLE=!t('PICTURE menu',*, \
  'What type of picture do you want ?'); \
  RESPONSE=_cdpict; MODE=t; DEFAULT='g'; HELP=_hppict] \
  VALUES='g','h','p'; CHOICE= \
  'graph of one variate against another', \
  'histogram of the values in a variate', \
  'PostScript high-resolution graph of one variate against another'; \
  HELP=_hpgraph,_hphist,_hppost

CASE _cdpict

  " ******(2) Graphs "  ... as in standard file

OR

  " ******(3) Histograms "  ... as in standard file

OR

  " ******(4) PostScript graphs "

  " Get the y-variate "
   QUESTION [PREAMBLE=!t('Y-VARIATE menu (from PICTURE menu)',*, \
    'What variate do you want to plot vertically ?'); \
    RESPONSE=_ptyvar; DECLARED=yes; TYPE=variate; PRESENT=yes]

  " Get the x-variate "
  QUESTION [PREAMBLE=!t('X-VARIATE menu (from PICTURE menu)',*, \
    'What variate do you want to plot horizontally ?'); \
    RESPONSE=_ptxvar; DECLARED=yes; TYPE=variate; PRESENT=yes]

  " Draw the graph "
  OPEN 'G21GRAPH'; CHANNEL=3; FILETYPE=graphics
  DEVICE 3
  DGRAPH _ptyvar[1]; _ptxvar[1]
  CLOSE 3; FILETYPE=graphics

  " Put statements in the record file that would draw the graph "
  PRINT [CHANNEL=_chcomnd; SQUASH=yes; IPRINT=*] !t( \
    'OPEN ''G21GRAPH''; CHANNEL=1; FILETYPE=graphics', \
    'DEVICE 3'); FIELD=1; SKIP=0; JUSTIFICATION=left
  &  'DGRAPH ',_ptyvar,'; ',_ptxvar; FIELD=1; SKIP=0
  &  'CLOSE 1; FILETYPE=graphics'; FIELD=1; SKIP=0

ENDCASE

" Return to basic menu. "
DELETE _hpgraph,_hphist,_hppost
RETURN
```

To add an additional menu system requires the provision of files of statements constructed in the same way as the standard System. There should be a base file, that will be entered from the MENU procedure, and this should contain statements to switch input to other files as necessary. The construction of the standard BASE Menu File should be closely followed. For a simple addition, the alternative base file may be enough. For a complicated system, it may be preferable to have several levels of files in the hierarchy – though the number of levels is limited by the number of input files that can be open simultaneously (usually 5).

The MENU procedure is defined in the Genstat Procedure Library in such a way as to allow additional systems to be added without the need for modifications. For example, assume that the file G2MNBAS1 is a new base menu file. Then the user needs to give the statement

```
MENU ['G2MNBAS1']
```

to enter the new system. The statement

```
MENU
```

will enter the standard System as usual. For reference, the definition of the MENU procedure is given below, with the definitions of file names and channel numbers as in the Start-up File.

```
" Set up texts containing logical names of standard menu & record files. "
TEXT _flmnbas,_flmnsub,_flcomnd,_flreslt,_flstore; VALUES='G2MNBASE',\
  !t(G2MNINPU,G2MNCALC,G2MNTABU,G2MNPICT,G2MNREGR,G2MNANOV,G2MNMULS,\
    G2MNMULD,G2MNTIME,G2MNTEST),'G21COMND','G21RESLT','G21STORE'
" Set up scalars with channel numbers for input, output and backing-store. "
SCALAR _chmnbas,_chmnsub,_chcomnd,_chreslt,_chstore; VALUE=4,3,5,4,5

" Define MENU procedure, for entering menu system from command mode."
PROCEDURE 'MENU'
  OPTION 'FILENAME','INCHANNEL','OUTCHANNEL'; \
    DEFAULT=_flmnbas,_chmnbas,_chcomnd
  GET [ENV=_envirp]
  EXIT [CONTROL=proc] _envirp['run'] .EQS. 'batch'
  " Turn off automatic logging of statements. "
  COPY [*] OUTCHANNEL
  SET [DIAG=f]
  CLOSE INCHANNEL
  SET [DIAG=w]
  " Use parameters to provide alternative menus and/or channels. "
  OPEN FILENAME; INCHANNEL
  " Input channel will not actually change until procedure has finished."
  INPUT [*] INCHANNEL ENDPROCEDURE
```

SASS Experience of Teaching Statistics using Genstat

C A Glasbey and G W Horgan
Scottish Agricultural Statistics Service
JCMB
The King's Buildings
Edinburgh EH9 3JZ

1. Introduction

The Scottish Agricultural Statistics Service (SASS) provides statistical and mathematical support for agricultural, environmental and food research and development in Scotland. SASS receives funding for 16 statisticians from the £28 million research and development budget of the Department of Agriculture and Fisheries for Scotland. For every 60 scientists, there is one statistician to cover statistical aspects of £2 million of research. Therefore, it is a strategic use of SASS resources to allocate one (shared) post to teaching most scientists to cope with their own data, freeing the other 15 to collaborate with the scientists where the most effective statistical inputs can be made.

The teaching objectives we set are that scientists should be able to use basic statistical methods, and recognise when they should consult a statistician. The challenge is to put statistical ideas across in more interesting ways, and make statistics more understandable, than has typically been achieved in university statistics courses. To this end, intuitive arguments are used instead of mathematical ones, and theory is taught by way of examples. Exploratory and graphical methods are emphasised as being as important as formal analysis. A key component is a simple-to-use, interactive computer package.

2. Plan

Five two-day modules have been prepared, each to be taught by two presenters. A bound set of notes is given to each participant.

The first module covers basic statistics and is built on Minitab. Emphasis is placed on exploratory methods for examining data prior to analysis, using graphs, tables and summary statistics. The course also includes elementary aspects of estimation, confidence intervals and hypothesis tests, and an introduction to analysis of variance and linear regression. Because of its ease of use and simple structure, Minitab syntax can be learnt very quickly.

Scientists whose statistical needs have outgrown what Minitab can provide are directed to Genstat. The second module teaches the basics of the Genstat 5 language: the audience is assumed to be familiar with the basic statistical methods covered in the first module. Although many attending the course have used Genstat 4, this is not assumed.

The three final modules build on the first two, and cover more advanced statistical and data analytic techniques:

(a) 'Experimental design and analysis of variance' covers randomisation, replication, blocking and factorial treatment structures. Analysis of variance is used to interpret experimental results.

(b) 'Regression and curve fitting' progresses from simple linear regression to multiple regression, and then to identifying and fitting nonlinear functions to data. Generalized linear models are briefly introduced.

(c) 'Graphical methods for multivariate data' presents methods for exploring and identifying structure in multivariate data. These include principal components, multidimensional scaling and classification techniques.

The Appendix gives an example of teaching material from the regression module. It is simply a Genstat interactive session. The lecturer uses these as a basis for explaining the concepts and methods of regression, and the scientists can, if they wish, reproduce the Genstat session themselves. This helps ensure they never feel the material is 'beyond them'. In each course about one third of the time is spent on exercises, in which the students practise analysing some supplied data. These serve to reinforce the teaching, and provide time for sorting out misunderstandings on a one-to-one basis.

3. Discussion

In the last year, the modules have been presented a total of 18 times to 240 scientists: an average of 13 participants per course. By basing courses around computer packages, and travelling to the various institutes and colleges (in Aberdeen, Ayr, Dundee and Edinburgh) to give them, we created problems for ourselves. It meant that we had to find a computer laboratory at each site, adapt to using a different computer and different terminals, and limit numbers of participants.

We found a very wide range of computer 'literacy'; even for a language as transparent as Minitab, more possibilities for misunderstanding were encountered than we ever imagined! There is nothing like giving a course for realising how to give it better. In all modules we found that we were too ambitious in how much we tried to cover in two days, and our illustrations of computer coding had to be more robust to possible misunderstandings.

Most people agree that statistics is a hard subject to teach. However, we found that the combination of the computer and a motivated audience goes a long way towards overcoming these problems. Finally, we recommend teaching experience for consultant statisticians: it helps to develop an overview of a subject, improves communication skills, and gives job variety.

Appendix: Example of teaching material from the Regression Module

```
>"
>   EXP:    The standard exponential     A + B*(R**X)
>   ==================================================
>
>  The standard exponential is monotonic and, because it has no point of
>  inflexion,  is either convex everywhere or concave everywhere.  It
>  approaches an asymptote  of A as X increases, and goes to +/- infinity
>  as X decreases (unless SENSE=L,  in which case it asymptotes for X
>  decreasing and goes to infinity as X  increases).   In particular:
>
>  if R < 1 (set by SENSE=R) and B  > 0 then it is decreasing & concave
>                         and if B  < 0 then it is increasing & convex
>  if R > 1 (set by SENSE=L) and B  > 0 then it is increasing & concave
>                         and if B  < 0 then it is decreasing & convex
>
>  We will illustrate these properties by considering some 'FROG' data.
>  Observations are the electrical current flowing through the end-plate
>  membrane of a frog's muscle fibre following a jump in voltage.   We
>  wish to study how the current changes with time.
>"
>OPEN 'FROG'; CHAN=2
>UNITS [124]
>READ [CHAN=2] TIME, CURRENT
     Identifier   Minimum      Mean   Maximum      Values   Missing
           TIME      1.25     16.62     32.00         124         0
        CURRENT     11.13     26.40     77.78         124         0     Skew
>GRAPH CURRENT; TIME
              -
              I
              I
          80. I  *
              I   3
              I    2*
              I     22
              I       22
              I        *22
          40. I          *222
              I             *2322
              I                  323222
              I                       *232232322*
              I                                 223223232232322323223
              I
           0. I
              -+---------+---------+---------+---------+---------+---------+-
              0.        6.       12.       18.       24.       30.       36.
                                CURRENT  v. TIME using symbol *
```

```
>"
>  The data exhibits the characteristic shape of an exponential (with
>  R < 1 and B > 0), although this does not mean that it has the
>  functional form of the standard exponential.  We can assess this
>  further by plotting the log-transformed currents, after subtracting
>  a guess at the asymptotic value (A).
>"
>CALC LOGCUR = LOG(CURRENT-10)
>GRAPH LOGCUR; TIME
              -
              I
              I  **
           4. I   2232
              I       2322
              I          *2232
              I               32232
              I                    32232
              I                         3222
           2. I                            *23223
              I                                  23222*
              I                                      **32*3*
              I                                          * *32*2*
              I                                               ******2
              I                                                  2  *
           0. I                                                   **
              -+---------+---------+---------+---------+---------+---------+-
              0.        6.       12.       18.       24.       30.       36.
                              LOGCUR  v. TIME using symbol *
>
>"
>  The linearity shows that the standard exponential looks like a
>  reasonable function to try!
>"
>MODEL CURRENT; RESID=R; FIT=F
>FITCURVE [CURVE=EXP] TIME
 ***** Nonlinear regression analysis *****
  Response variate: CURRENT
       Explanatory: TIME
      Fitted Curve: A + B*R**X
       Constraints: R < 1
 *** Summary of analysis ***
               d.f.         s.s.         m.s.
 Regression       2     34701.69     1.735E+04
 Residual       121        41.18     0.340E+00
 Total          123     34742.87     2.825E+02
 Percentage variance accounted for 99.9
 * MESSAGE: The following units have large residuals:
                             1                  2.77
                             2                  2.76
 * MESSAGE: The following units have high leverage:
                             1               0.153
                             2               0.130
                             3               0.110
                             4               0.094
                             5               0.080
                             6               0.069
 *** Estimates of parameters ***
                   estimate        s.e.
 R                 0.870912    0.000957
 B                   77.448       0.300
 A                   11.137       0.121
```

```
>GRAPH R; TIME
              -
              I
              I  **
          2.5 I
              I                    *
              I   *              * ****
              I   *               *  * 2**     2*
              I    *            ***   2  * 2 **    *    *  *
              I     *        *  *   *   * **2** 2 3*222   * *
          0.0 I    * *         * *      * *            *  2**
              I     * ** *  * 2*   *         *   *   * ***  *2*2*  *2
              I     **                                 * *     ****
              I       *2*******                  *            *  *  *
              I          * *                                     * *
              I         *                                         *
         -2.5 I           * *
              -+---------+---------+---------+---------+---------+---------+-
              0.        6.       12.       18.       24.       30.       36.
                              R  v. TIME using symbol *
>"
>  The lack of fit revealed by the residual plot indicates that we
>  must consider a more complicated function.
>
>   DEXP:    The double exponential     A + B*(R**X) + C*(S**X)
>   ===========================================================
>
>  The double exponential is the sum of two standard exponentials.   In
>  Genstat, the fit is constrained (by  R & S < 1 or R & S > 1) so that
>  both exponentials approach asymptotes in the same direction of X.
>  If the two parameters B & C have the same sign, then the double
>  exponential has a similar shape to the standard exponential.   If
>  they have opposite signs, then it has one turning point (a minimum
>  or maximum) and one point of inflexion (and is therefore convex for
>  one interval of X, and concave for another).
>"
```

49

Detecting Parallelism in Nonlinear Models

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

1. Introduction

A nonlinear model is one which cannot be written in the form

$$y = a + \Sigma b_k X_k + e$$

where a and b_k are parameters to be estimated, and X_k are explanatory variates or functions of explanatory variates with known parameters. (Ross [2]). In general, the random errors e need not be Normally distributed, but work with nonlinear models concentrates on models in which they are. The general form of a nonlinear model can be written as follows:

$$y = a + \Sigma b_k f_k(X;\theta_k)$$

which cannot be re-written so as to make θ_k linear with respect to X_k. The parameters θ_k are the nonlinear parameters. If k is 1, then the equation can be written more simply:

$$y = a + bf(X;\theta).$$

An example of this type of curve is the logistic curve:

$$y = a + c/(1+\exp(b(x-m))$$

where a and c are the linear parameters and b and m are the nonlinear parameters (θ).

This type of model is often used with data that is grouped, such as that from a herbicide trial using different spraying mechanisms, with the aim being to compare these mechanisms (groups) across a range of doses (values of X). A nonlinear model can be used to describe the relationship of the response to the dose, so the comparison between mechanisms can be made on the basis of deciding which, if any, of the parameters of this model need to be different between the groups. The basis for such a decision is often known as the analysis of parallelism (Ross [2]), and the rest of this paper discusses how such analyses can be carried out in Genstat. An example using real data and a procedure written to carry out the analysis of parallelism is also presented.

Four types of model are considered which differ according to which of the parameters vary between groups: there are several other possibilities which are not discussed here.

In the model descriptions below, the case of $k = 1$ above is illustrated, with y_{ij} the jth observation for group i. Similar equations apply when $k > 1$.

(a) Single Curve

All parameters are common between groups.

$$y_{ij} = a + b\ f(x_{ij};\theta)$$

(b) Parallel Curves, or Separate Constants Model

Constant parameter a varies between groups.

$$y_{ij} = a_i + bf(x_{ij};\theta)$$

This model is also referred to by Ross [2] as a vertical displacement model, since it describes curves of the same shape shifted on the vertical axis.

(c) Separate Linear, Common Nonlinear Parameters Model

All Linear Parameters vary between groups, but nonlinear parameters do not.

$$y_{ij} = a_i + b_i f(x_{ij};\theta)$$

This model can be said to have nonlinear parallelism.

(d) Separate Curves

All parameters vary between groups.

$$y_{ij} = a_i + b_i f(x_{ij};\theta_i)$$
$$i = 1...ng \text{ (number of groups);}$$
$$j = 1...n_i \text{ (number in group } i\text{)}$$

There is also a fifth possibility with common linear and some separate nonlinear parameters, described by Ross [2] as a horizontal displacement model, since it describes curves of the same shape shifted on the horizontal axis. This important model will not be considered here because, firstly, it is not a model that can be fitted using the standard nonlinear directive FITCURVE, and also because it does not easily fit into the hierarchy of the first four models described. This is because horizontal displacement is achieved by keeping the linear parameters constant between groups and allowing a subset of the nonlinear parameters to vary between groups, the subset being dependent on the particular model in use.

The first four of the above models can easily be fitted for the nonlinear curves provided with the FITCURVE directive. For instance, a logistic model can be tested with the following commands (cf Genstat 5 Reference Manual, page 371):

```
MODEL y
TERMS x * factor
FITCURVE [CURVE=logistic] x
ADD factor
ADD factor.x
ADD [NONLINEAR=separate]
```

This produces an accumulated ANOVA which can be used to choose between models:

```
Change
+ x                             {single curve}
+ factor                        {separate constants}
+ factor.x                      {separate linear}
+ Separate Nonlinear            {separate curves}
```

If the model to be fitted is not one of the standard curves, or else is a sum of nonlinear functions, the commands for producing such an analysis within Genstat are not obvious (in contrast with MLP, Ross [2], where such analyses are relatively straightforward). For example, the required model might be a sum of three exponentials (FITCURVE provides a double exponential):

$$y = a + b_1 e^{k1x} + b_2 e^{k2x} + b_3 \; e^{k3x}$$

(Note: This model must be treated with caution, since estimates of its parameters are often unstable, Ross [3].)

This is a nonlinear model with $k = 3$ and $f_l(x;\theta_l) = e^{-klx}$.

For any nonlinear model, the single curve and separate constants models are easily fitted in a manner similar to that for FITCURVE.

```
EXPRESSION e; VALUE=!e( f[1...3] = EXP( x*k1,k2,k3 ) )
MODEL y
RCYCLE k1,k2,k3
FITNONLINEAR [CALCULATION=e] f[]          {single curve}
FITNONLINEAR [CALCULATION=e] f[],factor   {separate constants}
```

The equivalent for the other two models does not work. The separate curves model can be fitted relatively easily, although some effort is required to calculate a pooled residual variance, and the appropriate standard errors of the parameters. The model can be fitted to each level of the grouping factor in turn, after the EXPRESSION, MODEL and RCYCLE statements have been made:

```
FOR i=1...ng
   RESTRICT y,x; factor.EQ.i
    FITNONLINEAR [CALCULATION=e] f[]
ENDFOR
```

The remaining model, Separate Linear Parameters, is not easy to fit. The equation required to be fitted is:

$$y_{ij} = a_i + b_{ik} \; f_k(x_{ij};\theta_k)$$

Two different approaches for fitting this model are described below.

2. Implicit Dummy-variable Method

If the nonlinear parameters were known, then estimates of the linear parameters a and b could be found using multiple regression. These can be written in terms of the nonlinear functions f_k and their parameters $\boldsymbol{\theta}_k$, substituted back into the original nonlinear equation. This gives an equation with only nonlinear parameters to be estimated explicitly.

Maximum Likelihood equations for a and b:

$$\hat{a}_i = \bar{y}_i - \Sigma b_{ik} \underset{j}{\text{mean}}(f_k(x_{ij};\boldsymbol{\theta}_k))$$

$$\hat{\boldsymbol{b}}_i = (X_i^T X_i)^{-1} X_i^T (y_i - \bar{y}_i)$$

where $\boldsymbol{y}$ is the vector of all observations,
$\boldsymbol{y}_i$ is a vector of observations for group i,
$\boldsymbol{x}_i$ is the explanatory variable for group i,
$\boldsymbol{X}_i$ is a matrix with nf columns $f_{ki} - \bar{f}_{ki}$,
and $f_{ki} = f_k(\boldsymbol{x}_i;\boldsymbol{\theta}_k)$;
$k = 1...nf$, the number of functions,
$i = 1...ng$, the number of groups,
$j = 1...ni$, the number of observations for group i.

After substituting for the linear parameters a and $\boldsymbol{b}$, the predicted values for y_i, $\hat{y}_i$ are given by:

$$\hat{y}_i = \bar{y}_i - X_i(X_i^T X_i)^{-1} X_i^T (y_i - \bar{y}) \qquad i = 1...ng$$

This method can be thought of as a profile-likelihood method for the nonlinear parameters. This equation can be programmed as a Genstat expression to be fitted with FITNONLINEAR which can then minimise the differences between $\hat{y}$ and the observed values y. The linear parameters a and b can then be found by substituting the estimates for the nonlinear parameters back into in the maximum-likelihood equations.

3. Explicit Dummy-variable Method

This method is more straightforward in its approach, and involves all parameters being estimated at the time of fitting. The same initial equation can be re-written in matrix form, incorporating dummy variables to impose the required restrictions on the parameters.

$$\hat{y} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ 0 & 1 & & \vdots \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} a + \begin{bmatrix} F_1 & 0 & \cdots & 0 \\ 0 & F_2 & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & & F_{ng} \end{bmatrix} b$$

where F_i is a nf by ni matrix with ijkth element being f_{ijk}, the value of function f_k for the jth observation of group i, $i = 1...ng$, $j = 1...ni$, and $k = 1...nf$. A set of expressions for the above equation can be written in Genstat, and then FITNONLINEAR used to estimate all the parameters directly.

4. Comparison of the Two Methods

A potential advantage of the first method is that only the nonlinear parameters are explicitly estimated, whilst the second method explicitly estimates $ng*(1+k)$ more parameters. Thus, in theory, the estimation process for the second method should be both more time consuming and take up more computer space, especially with a large number of factor levels and/or a large number of functions. A second potential advantage of the first method is that the structures needed for the estimating equations should take up less space: the second method requires a dummy variate for each column of the dummy variable matrices in the equation, whilst the first method should be able to make use of less space-consuming matrices instead.

These potential advantages for the first method led to a search for an efficient method of translating it into Genstat. Below is a set of commands to use the method for the case when there were four groups (*ng* = 4) and three functions (*nf* = 3), *f*[1...3] with nonlinear parameters *n*1, *n*2, *n*3.

```
RESTRICT 4(y); factor.EQ.1...4; SAVE=s[1...4]
RESTRICT y
VARIATE [NVALUES=3] ybar
   "save position of data for each factor level"
CALCULATE ybar$[1...4] = MEAN( ELEMENTS(y; s[]) )
   "set up matrices to hold inverses of xTx"
MATRIX [ROWS=3; COLUMNS=3] inv[1...4]
   "set up X matrices"
FOR i=1...4
   MATRIX [ROWS=s[i]; COLUMNS=3] X[#i]
ENDFOR
   "set up equation to evaluate X matrices"
FOR i=1...3
   EXPRESSION e[i]; VALUE=!e( X[]$[*; #i] = \
      ( X[]$[*; #i] = ELEMENTS(f[i]; s[]) ) - (MEAN( X[]$[*; #i] ) )
ENDFOR
   "set up final estimating equation"
EXPRESSION e[4]; VALUE=!e( ELEMENTS(4(fit); s[]) = \
   #ybar + X[] *+ INVERSE( inv[] = TRANSPOSE(X[]) *+ X[] ) \
   *+ TRANSPOSE(X[]) *+ ( ELEMENTS(y; s[]) - #ybar ) )
MODEL y; FITTED=fit
RCYCLE n1,n2,n3
FITNONLINEAR [CALCULATION=e]
```

This needs to be followed by more calculations to obtain the linear parameters. The FITNONLINEAR statement minimises the sum of the squared differences between `fit` and the data y, which is why FITNONLINEAR has no parameters. The programming involved becomes a compromise between many complex expressions and many intermediate structures to hold results. In the case of a single function, the programming can be simplified quite dramatically, because the matrix algebra reduces to linear algebra, and so there is no need for the INV and TRANS functions.

In contrast, the Genstat statements for the Explicit dummy variable method are straightforward:

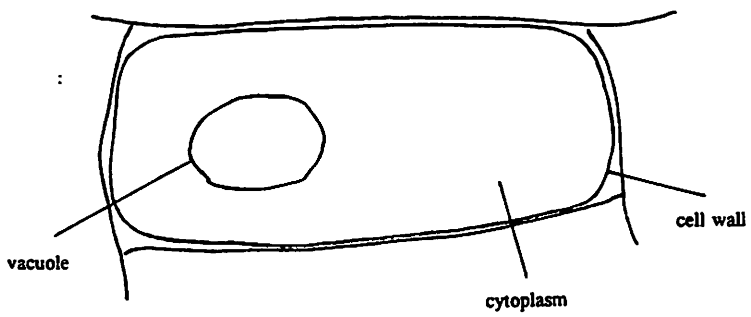
```
FOR i=1...3
   EXPRESSION e[i]; VALUE= \
      !e( fit[#i][1...4] = #f[i] * ( factor .EQ. 1...4 ) )
ENDFOR
MODEL y
RCYCLE n1,n2,n3
FITNONLINEAR [CALCULATION=e] fit[][],factor
```

The dummy variables fit[][] are used to estimate the nonlinear and slope (b) parameters, and the 'factor' parameter ensures separate constants are fitted as well. It was intended to compare the efficiency of these two approaches in terms of time and space used, but it very quickly became apparent that the INVERSE function used in the first method had too many problems associated with it to make the method reliable. The precision of the arithmetic used in the INVERSE algorithm is not great enough, and it does not easily deal with nearly singular matrices. Secondly, the X^TX matrices are easy to make singular in the estimation process, especially if the functions involved are of the same type (eg. all exponentials). Thirdly, when a variate is fitted explicitly (as in the second method), Genstat performs various rescaling actions on the independent variable to stabilise the process: this does not occur with the function minimisation used in the first method. Because of these problems, any potential advantages of the first method are cancelled (except perhaps in the case of a single function), so a procedure to perform analysis of non- parallelism should be based on the second method.

Such a procedure (FITPARA) has been written using Genstat 5 Release 2; it fits all four models and pools the results to produce an accumulated analysis of variance table, and pooled results for the fourth model (separate curves). It is hoped to submit this procedure for inclusion in a future release of the Genstat Procedure Library.

5. Example of Analysis of Parallelism for a Nonlinear Model

This data is from a series of experiments designed to examine the uptake of herbicide into moss cells (Mabb [1]).



Fragments of moss were immersed in a solution of radiolabelled herbicide until the cells were saturated. The herbicide then resided in three parts of the cells (vacuoles, cytoplasm and cell walls) in differing proportions. The aim was to discover how much herbicide was present in each of these three parts. The herbicide solution was removed and replaced with a wash. The herbicide within the cells gradually diffused from all of the parts out into the wash. After an interval, this wash was removed, replaced with another, and the level of radioactivity measured in the first wash. This process was repeated with several washes. The whole experiment was carried out on two replicate runs. The radioactivity measured for any given wash (in disintegrations per minute) can be modelled by the sum of three differenced exponentials, each associated with the leaching of the herbicide out of one of the three cell parts:

$$dpm = diff(b_1 e^{-k1.time}) + diff(b_2 e^{-k2.time}) + diff(b_3 e^{-k3.time})$$

where b_i is a measure of the amount of herbicide in cell part i at the beginning of the washing process, and k_i is a measure of the rate of leaching from that part. It was hoped that the parameters b_i and k_i, would be the same between replicates, giving a general measure of the amount of herbicide held in each part of the cell. This required analysis of parallelism for this model of the sum of three nonlinear functions. The data was analysed using the procedure described above, with replicate runs being used as a grouping factor, time as an explanatory variate, and dpm per wash as the dependent variate.

In the following program, three expressions are set up, one per differenced exponential function. These expressions include checks to prevent the EXP function being passed values that are too large for it to deal with. After setting up the *y* variate and the parameters with MODEL and RCYCLE, the FITPARA procedure is then used for the analysis. The options of the procedure are similar to those of the FITNONLINEAR directive, with an additional option MODEL which selects which of the four models is to be the final one fitted (Single curve *s*, Separate Constants *sc*, Separate Linear *sl*, Separate nonlinear *sn*.). The procedure fits all models up to the most complex fitted, (although if CONSTANT is set to omit, the Separate Constants model is obviously omitted). The accumulated ANOVA produced shows that common nonlinear (i.e. rate parameters k_i) can be used for both runs, indicating that the rate of leakage from the various parts was similar between the two runs, and that separate linear parameters are not required, showing that the original amount in each part did not vary significantly between runs. The procedure is then used to fit the single-curve model to obtain the parameter estimates for it. The observed results and the final model are illustrated in the figure.

```
  1  JOB 'Parallel Curves Example'
  2  OPEN 'proc.stor'; CHANNEL=1; FILETYPE=procedure
  3  READ [PRINT=data,summary; SETNVALUES=yes; SERIAL=yes] time,dpm

  4  0 1 3 6 10 15 20 30 40 50 60 80 100 120 140 160 180 210
  5  240 270 300 360 420
  6  0 1 3 6 10 15 20 30 40 50 60 80 100 120 140 160 180 210
  7  240 270 300 360 420
  8  :

   Identifier   Minimum       Mean   Maximum    Values   Missing
         time       0.0      122.4     420.0        46         0
  9   *
 10   557.2 816.1 783.3 561.4 461.5 298.7 306.9 253.3 212.6
 11   190.2 205.8 221.2  195.4 217.7 211.8 182.3 225.5 214.1 222.0
 12   201.8 301.2 270.1
 13   *
```

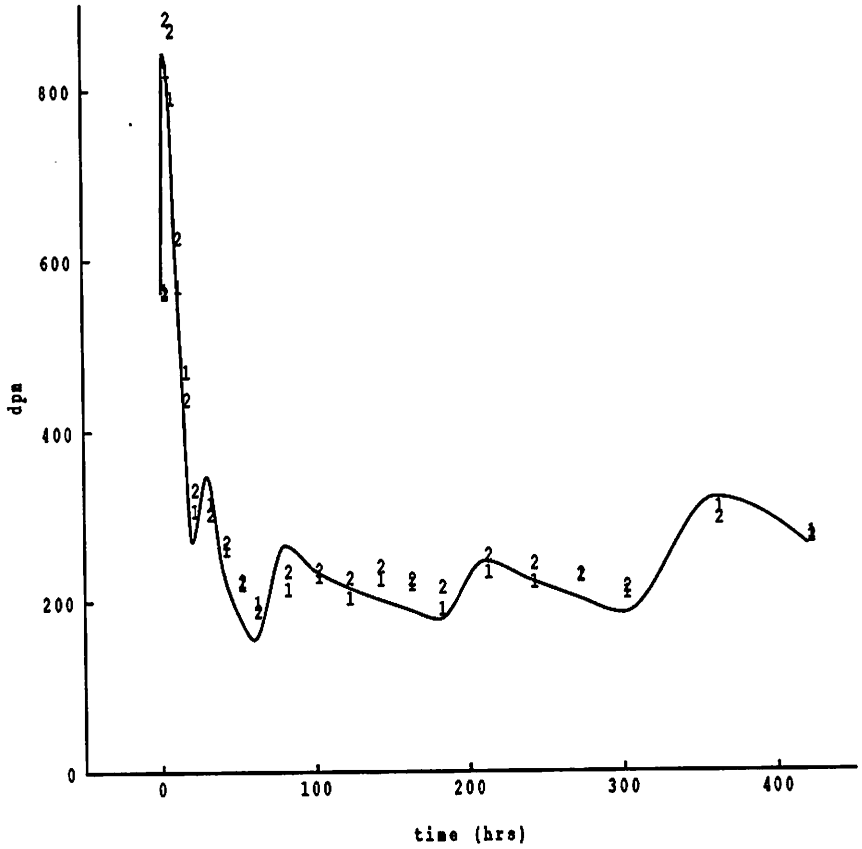
```
14   554.0 875.6 862.1 618.2 429.1 323.4 295.2 262.0 216.0
15   180.1 227.1 228.3 218.5 232.2 217.0 208 245.9 236.0 222.8
16   208.4 288.3 266.7
17  :
        dpm     180.1      330.1     875.6         46          2     Skew
18  FACTOR [LEVELS=2; VALUES=23(1...2)] run
19
20  EXPRESSION exp; VALUE=!e(DIFF(EXP(-(a,b,c = k1,k2,k3*time) * \
21     (a,b,c .LT. 35)*(a,b,c .GT. 0) - 35*(a,b,c .GE. 35))))
22
23  MODEL dpm; FITTED=fit
24
25  RCYCLE [MAXCYCLE=50] k1,k2,k3; LOWER=0,0; INITIAL=0.003,0.05,2
26
27  FITPARA [PRINT=m,a; CONSTANT=omit; CALCULATION=exp; MODEL=sn] \
28     FACTOR=run; VARIATE=time

***** Nonlinear Regression Analysis *****

    Response variate: dpm

         Explanatory: time
      Grouping factor: run
         Fitted Curve: Constant + Slope*f( time ; theta)
   Nonlinear Parameters(theta): k1 k2 k3

***** Nonlinear Regression Analysis *****

*** Accumulated analysis of Variance ***

Change                               d.f.         s.s.         m.s.      v.r.
+ time                                  6      6399336      1066556   1186.89
+ time.run                              3         4770         1590      1.77
+ Separate Nonlinear                    3         1367          456      0.51
Residual                               32        28756          899

Total                                  44      6434230       146232
29
30  FITPARA [PRINT=e; CONSTANT=omit; CALCULATION=exp; MODEL=s] \
31     FACTOR=run; VARIATE=time


  Common Curve for all Levels of run


30.....................................................................


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

*** Estimates of parameters ***

                    estimate          s.e.
k1                  0.003192      0.000300
k2                    0.0649        0.0157
k3                    0.2298        0.0226
* Linear
Slope[1]              -4798.          179.
Slope[2]              -1426.          305.
Slope[3]              -2224.          343.
```

6. References

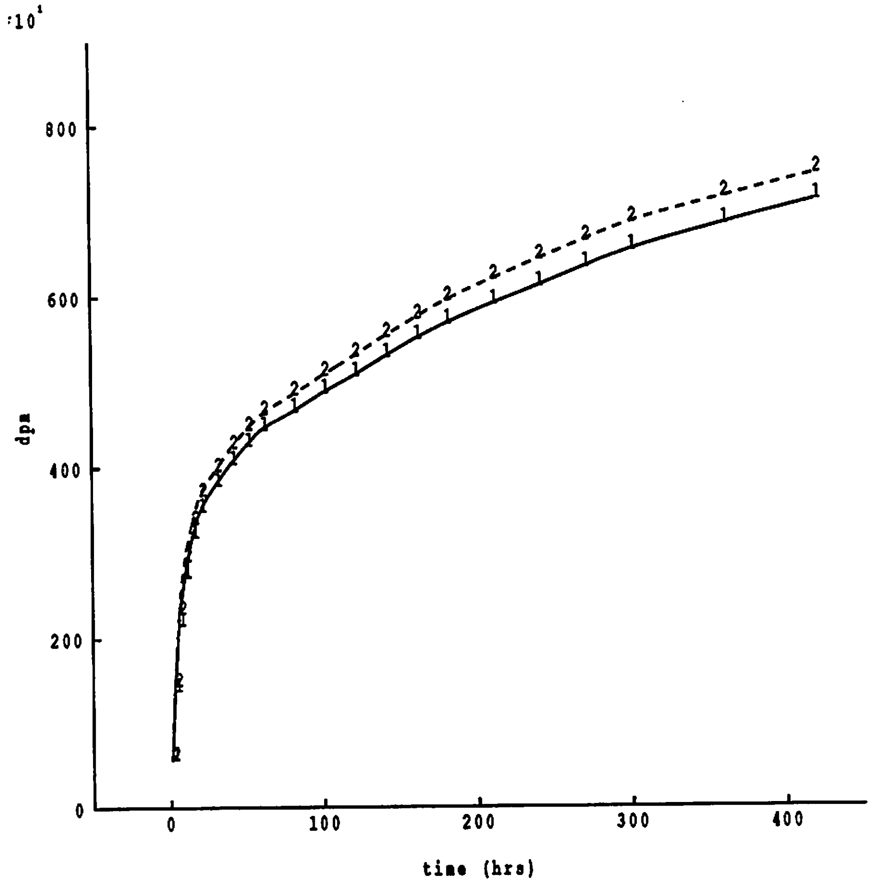
[1] Mabb, L.
Uptake and Effects of Herbicides on the Lawn Moss *Rytidiadelphus Squamus.*
Phd Thesis, London University, 1989.

[2] Ross, G.J.S.
Maximum Likelihood Program 3.08.
NAG, Oxford, 1987.

[3] Ross, G.J.S.
The Use of Nonlinear Regression Methods.
In: *Crop-Modelling in Mathematics and Plant Pathology,* (D.A. Rose and
D.A. Charles-Edmonds, Eds.)
Academic Press, 1981.



Fitted Model



Fitted Model Cumulated Over Time

Fitting a General Growth Model to Data

A Keen
Institute of Agricultural Engineering
Wageningen
The Netherlands

1. Introduction

Schnute [1] has described a general four-parameter growth model for a continuous non-negative response y_t with expectation μ_t that is a non-decreasing function of time t. First the derivation of the model and its generality will be discussed and also its parameterisation. Then aspects of fitting the model using Genstat are considered. This includes the default choices that have been made in procedure FITSCHNUTE that has been designed for fitting the general model or one of the submodels. The procedure itself will not be discussed in detail, as it has been submitted to the Genstat Procedure Library.

2. The Model

The growth model is a non-negative and non-decreasing function of t. This means that $\mu_t \geq 0$ and the growth rate $d\mu_t/dt \geq 0$. In the course of time many functions have been proposed to describe growth in this situation. Some of these proposals are given in the first two columns of Table 1. Obvious restrictions on the parameters are necessary in order to ensure that the function is non-negative and non-decreasing. Note that the linear function is included in pth power, exponential in pth power of exponential, logistic in Richards (also called generalized logistic) and monomolecular is included in Von Bertalanffy. Also it is a well-known fact that the Gompertz function is a limiting case of the Richards function (for $h \rightarrow 0$) and the Von Bertalanffy function (for $b \rightarrow 0$).

Model Name	Function	$z = (d\mu/dt)/\mu$	$w = (dz/dt)/z$
linear	$\alpha + \beta t$	β/μ	$-z$
pth power	$(\alpha+\beta t)^p$	$p\beta/\mu^{1/p}$	$-z/p$
exponential	$\alpha + \beta e^{\gamma t}$	$\gamma(1-\alpha/\mu)$	$-(-\gamma+z)$
pth power of exp.	$(\alpha+\beta e^{\gamma t})^p$	$p\gamma(1-\alpha/\mu^{1/p})$	$-(-\gamma+z/p)$
monomolecular	$\mu_\infty[1-e^{-\alpha(t-t_0)}]$	$\alpha(\mu_\infty/\mu-1)$	$-(\alpha+z)$
Gompertz	$\mu_\infty e^{-e^{-\alpha(t-t^*)}}$	$\alpha\log(\mu_\infty/\mu)$	$-\alpha$
logistic	$\mu_\infty/[1+e^{-\alpha(t-t^*)}]$	$\alpha(1-\mu/\mu_\infty)$	$-(\alpha-z)$
Von Bertalanffy	$\mu_\infty[1-e^{-\alpha(t-t_0)}]^{1/b}$	$(\alpha/b)\{(\mu_\infty/\mu)^b-1\}$	$-(\alpha+bz)$
Richards	$\mu_\infty/[1+he^{-\alpha(t-t^*)}]^{1/h}$	$(\alpha/h\{1-(\mu/\mu_\infty)^h\}$	$-(\alpha-hz)$

Table 1

An important function of μ_t is the relative or specific growth rate z_t:

$$z_t = (d\mu_t/dt)/\mu_t = d\log(\mu_t)/dt$$

If z_t is constant, then growth is exponential or unrestricted; so starting with z_t as a basis for further discussion is relevant if it is relevant to know whether and how growth is restricted compared to unrestricted growth. As z_t is an important feature of μ_t, then the way z_t changes that is,

$$w_t = (dz_t/dt)/z_t,$$

may also be considered as an important feature of z_t (and therefore of μ_t). Evaluating z_t as a function of μ_t, it turns out, that for all the tabulated functions z_t is a linear function of μ_t^q:

$$z_t = b_0 + b_1\mu_t^q = b_1(b_0/b_1+\mu_t^q)$$

This can be verified in the third column of Table 1, where z_t is given for all functions in the table. Solving this differential equation results in the general expression that μ_t^q is a linear function of e^t. This solution requires one boundary condition, e.g. that $\mu = \mu_1$ if $t = t_1$. The resulting parameterisation, however, does not seem very useful. Observe that the relative growth rates of the Richards and the Von Bertalanffy function are identical when taking $b = -h$. This indicates that these functions can be written in the same form, as one expression. It may be noted that in the original expressions their parameterisation differs: t_0 in the Von Bertalanffy function is the starting point for growth (that is undefined in the Richards function), whereas in the Richards function it is the point of inflexion. To write it as one expression, obviously the same parameterisation must be used.

In the fourth column of Table 1 the relative growth rate of z_t, w_t, is given and can be seen to satisfy, for all tabulated functions, a simple linear equation:

$$w_t = -(a+bz_t)$$

Now two boundary conditions are required for a general solution of this equation. Schnute [1] chose:

$$\mu_t = \mu_i \text{ if } t = t_i \text{ ; } i = 1,2$$

The resulting equation (also using the definition of z_t) is:

$$\mu_t^b = \mu_1^b + (\mu_2^b - \mu_1^b)A_t(a) \qquad [1]$$

where:

$$A_t(a) = \frac{1 - e^{-a(t-t_1)}}{1 - e^{-a(t_2-t_1)}}$$

Note that the solution can also be obtained from the Richards or Von Bertalanffy equation, just changing the parameters μ_∞ and t_0 or t^* into μ_1 and μ_2.

Parameter b is dimensionless and is unaffected by a linear scale transformation of y and a linear transformations of t. Parameter a has dimension t^{-1}, so a linear transformation of t changes a. The boundary conditions define two parameters μ_1 and μ_2, both with the dimension of y_t. The advantage of using μ_1 and μ_2 is that these parameters always exist and are well estimable if t_1 and t_2 are chosen properly. Changing t_1 and t_2 changes the value of μ_1 and μ_2 respectively, but has no effect on a or b. In general: μ_1 and μ_2 determine location and scale of the curve, while a and b determine its shape.

The growth model can be seen as a generalisation of a straight line through the points (t_1, μ_1) and (t_2, μ_2). For a straight line the shape is fixed taking $a = 0$ and $b = 1$. In general, parameter b takes care of a power transformation of the expected response μ_t (while the definition of the curve can be extended to the complete real axis defining $\mu_t = 0$ if $\mu_1^b + (\mu_2^b - \mu_1^b)A_t(a) < 0$). Parameter a defines the transformation of the t-axis, with $a = 0$ for the identity transformation. $A_t(a)$ has an upper asymptote if $a > 0$ and a lower asymptote if $a < 0$. The two parameters a and b together produce curves whose shape can vary from a straight line to S-shaped curves for $a > 0$ and to some image of such curves for $a < 0$, where growth starts at some timepoint t_0 with large growth rate that decreases and later increases again.

Submodels are models with fixed values of parameters a and/or b of the growth model, e.g. $a > 0$ and $b = -1$ specifies the logistic function, $a > 0$ and $b = 0$ the Gompertz function. The generality of the model can also be restricted by specifying a particular range of a and b. Included in the general model are the curves 'exponential', 'logistic', 'glogistic', 'gompertz' and 'ldl' of FITCURVE (the last four with CONSTANT=omit only).

In biology, S-shaped curves are most commenly applied. A marked feature of such curves is the position of the point of inflexion, relative to the upper asymptote: μ^*/μ_∞. This position is completely determined by parameter b:

$$\mu^*/\mu_\infty = (1-b)^{1/b}$$

For the logistic function $b = -1$ and $\mu^*/\mu_\infty = \frac{1}{2}$. The Richards curves range from $\mu^*/\mu_\infty \approx 1$ (for b large negative) to $\mu^*/\mu_\infty = 1/e$ (for $b = 0$, the Gompertz curve). The directive

FITCURVE with option CURVE=glogistic fits such curves (based on Normal likelihood only). However, if μ^*/μ_∞ happens to be smaller than $1/e$, then a warning is given that a boundary is reached and that the Gompertz curve may be a better choice. There is no possibility of crossing the Gompertz border with FITCURVE. The Schnute model allows a continuous transition from the Richards model to curves with μ^*/μ_∞ smaller than $1/e$. In fact, the S-shaped curve for $b > 0$ then drops below 0, while values below 0 are undefined and changed into zeroes. The point where growth starts is defined as t_0. The limit for the S-shape is $b = 1$, at which value in fact only the upper arc of the S has been left behind.

Parameter a determines for S-shaped curves the curvature of both arcs. This is illustrated by the relative growth rate in the point of inflexion z^* and by the distance between t^* and t_0 (both are defined if $0 < b < 1$):

$$z^* = a/(1-b)$$

$$t^* - t_0 = \log(1/b)/a$$

The nicest feature of the model, however, is its flexibility. Members of the family of curves may or may not have asymptotes, or may or may not deviate much from a straight line or from an exponential model.

3. Fitting the Model

For fitting the model, not only the systematic part, but also the random part has to be modelled. The response y_t with mean μ_t is:

$$y_t = \mu_t + \varepsilon_t$$

Usually ε_t's belonging to different observations are independent (due to the design of the experiment). For given values of the parameters a and b the model is in fact a generalized linear model (GLM) in $A_t(a)$, with power link and exponent b. Also, for b fixed, it is a nonlinear model with two linear parameters and one nonlinear parameter. However, the part-linearity of the model cannot be exploited in general. Consider the model with fixed b:

$$\mu_t^b = \alpha + \beta A_t(a)$$

For non-Normal distributions that are specified for ε_t, the right-hand side of the model must be non-negative (for the Poisson distribution) or even strictly positive (for gamma and inverse Normal distribution). This requirement causes the GLM-formulation to be unsatisfactory. Another problem is the existence of responses equal to 0, which are not allowed for the gamma and inverse Normal distribution. This is a matter mainly concerning the variance function, which will be discussed later.

The consequence of these problems is that a general fitting procedure must be based on a nonlinear regression formulation of the problem, solved by optimisation. Genstat directives that enable the fitting of a user-defined model are MODEL, RCYCLE and FITNONLINEAR, while calculation of fitted values takes place in a number of expressions. For this fitting procedure to be successful it is very important to take care of the following:–

(a) The parameterisation has to be good.

Goodness of parameterisation in this sense is interpreted as: correlations between estimates must be low and they must not deviate too far from linearity. Because a and b are supposed to be good parameters (transformations may be considered, but no serious effort in this direction has been made), the parameters to be discussed are μ_1 and μ_2. They are defined by t_1 and t_2, where t_1 is a timepoint in the beginning of the observed growth period and t_2 a timepoint at the end of the growth period. The requirements are that t_1 and t_2 must be far enough apart (to obtain uncorrelated estimates) and that μ_1 should not be too close to 0. A good choice for t_2 is the last timepoint for which an observation has been obtained. However, the choice of t_1 is less simple. The requirement that μ_1 should not be close to 0 is extremely important! A more precise statement about this requirement is: for all combinations of parameters in the search for an optimum the value of μ_1 must be positive. A practical approach is to fit a simple model that approximates the curve locally well and to choose t_1 as the point at which $\hat{\mu}_1/\hat{\mu}_2 \approx 0.3$.

(b) μ_t must be calculated for all relevant values of the parameters without overflow problems.

The problems, with their solutions, are:

(i) $A_t(a) \rightarrow 0/0$ if $a \rightarrow 0$. The limit is $(t-t_1)/(t_2-t_1)$.

(ii) $A_t(a)$ is approximated by the limit if $a(t_2-t_1) < 0.001$.

(iii) For $b \rightarrow 0$, $\mu^b \rightarrow 1$. The limit of $(\mu^b-1)/b$ then is $\log(\mu)$.

For $b < 0.001$ this approximation is used, in the way that in equation [1] μ^b is replaced by $\log(\mu)$. This is the Gompertz approximation.

$|b|$ and $|a(t_2-t_1)|$ must not be too large because these quantities are values of exponents.

Limits taken are: 20 for $|b|$ and 40 for $|a(t_2-t_1)|$. The curves for extreme values of these quantities are not of practical interest and furthermore their shape will not be changed by changing values beyond these limits.

Most approximations and restrictions can be implemented in the expressions for calculating fitted values. The limits for a and b are best defined by the parameters LOWER and UPPER of the RCYCLE directive.

(c) Good initial values for the parameters are required.

A simple and effective method of obtaining good initial values for μ_1 and μ_2 is to apply an approximating function. A possibility is to fit a spline, for example. Initial values for a and b that are usually sufficiently accurate are those belonging to a straight line: 0 for a and 1 for b. Sometimes this may be too crude. A grid search in a and b with fixed values for μ_1 and μ_2 may then be a good method.

(d) Choice of distribution and weights have to be relevant.

Where points (a) to (b) relate to the numerical aspects of the fitting procedure, the choice of distribution and weights relate to the biological problem. The choice, however, may induce numerical problems.

The most important aspect of the random component of the model ε_t is its variance function. For positive variables with responses of extremely different size assuming constant variance is usually not a good approximation of reality. At first glance a power variance function seems reasonable:

$$\sigma_t^2 = \phi\mu_t^q$$

which can be specified by Genstat through the DISTRIBUTION option of directive MODEL. $q = 0, 1, 2$ and 3 can be chosen by taking the Normal, Poisson, gamma and inverse Normal distribution respectively. Increasing q implies increasing the relative importance of deviance contributions from observations with small values for μ_t. If μ_t is very small the variance function is usually unrealistic for non-Normal distributions, due to rounding and approximation errors. For the gamma and inverse Normal distribution it is theoretically impossible to obtain observations equal to 0, but in practice, due to rounding and approximation errors, such observations do occur. In calculating the deviance small values of y_t or μ_t cause numerical problems. A better variance function would be:

$$\sigma_t^2 = \phi\mu_t^q + c$$

with c a small constant that represents the variance component due to round-off and approximation error. This variance function, however, can not be applied by standard means with Genstat. Therefore an approximation is used:

$$\sigma_t^2 = \phi(\mu_t+c)^q$$

This variance function can be simply implemented without affecting the systematic part of the model by adding a small constant to y_t as well as to μ_t.

4. References

[1] Schnute, J.
A versatile growth model with statistically stable parameters.
Can. J. Fish. Aquat. Sci., **38**, pp. 1128-1140, 1981.

The Inefficiency of the Recovery of Inter-block Information about Non-orthogonal Treatments from Genstat ANOVA

K J Worsley
Department of Mathematics and Statistics
McGill University
805 Sherbrooke St. O.
Montréal
Québec H3A 2K6
Canada

1. Introduction

The inefficiency that I wish to consider only arises in the rare case of non-orthogonal treatment factors where we want to recover the inter-block information by a linear combination of separate stratum effects. In a recent Genstat Newsletter, Preece [4] has looked at such designs in some detail. Similar designs have been used for blocked diallel cross experiments – see Ceranka and Majza [1] for an example. In this article, I give examples of balanced non-orthogonal two-factor block designs where the linear combination of stratum effects is either not fully efficient for either term, or fully efficient for one term but not the other, or fully efficient for both terms. In practice the lack of efficiency is usually very small, but it is possible to find an extreme example, mentioned by Preece [3], page 499, where the efficiency is zero for both terms. In this example there is no information about either term in any stratum, and yet both terms are estimable!

2. Recovery of the Inter-block Information

Consider the following block design given by Preece [4,(4)], with two treatment factors *T1* (upper case letters) and *T2* (lower case letters) for 30 observations in 10 blocks:

Block

1	2	3	4	5	6	7	8	9	10
Aa	*Bb*	*Cc*	*Dd*	*Ee*	*Aa*	*Bb*	*Cc*	*Dd*	*Ee*
Cb	*Dc*	*Ed*	*Ae*	*Ba*	*Bd*	*Ce*	*Da*	*Eb*	*Ac*
De	*Ea*	*Ab*	*Bc*	*Cd*	*Ec*	*Ad*	*Be*	*Ca*	*Db*

(D1)

The commands

```
BLOCKS block
TREATMENTS T1 + T2
```

give the following ANOVA table and information summary:

```
***** ANALYSIS OF VARIANCE *****

SOURCE OF VARIATION              DF

BLOCK STRATUM
  T1                              4
  T2                              4
  RESIDUAL                        1
TOTAL                             9

BLOCK.*UNITS* STRATUM
  T1                              4
  T2                              4
  RESIDUAL                       12
TOTAL                            20

GRAND TOTAL                      29
```

```
***** INFORMATION SUMMARY *****

MODEL TERM                           EF   NON-ORTHOGONAL TERMS

BLOCK STRATUM
  T1                               0.167
  T2                               0.093  T1

BLOCK.*UNITS* STRATUM
  T1                               0.833  BLOCK
  T2                               0.741  BLOCK  T1
```

Both treatment factors are estimable but non-orthogonal in both strata. As a result, the effect of *T1* is not adjusted for *T2*, as Preece [4] points out, so we shall concentrate exclusively on the effect of *T2*, which is adjusted for *T1*. Let $\hat{\beta}_B$ and $\hat{\beta}_W$ be vectors of the estimated effects of *T2* between blocks and within blocks respectively, as calculated by Genstat ANOVA. Then $\hat{\beta}_B$ and $\hat{\beta}_W$ are independent and unbiased for the true effect β, with information matrices proportional to their unadjusted (for *T1* and blocks) information matrix, M say. The constants of proportionality, or efficiency factors, are $e_B = 0.093$ between blocks and $e_W = 0.741$ within blocks. Thus if we know the stratum variances σ_B^2 between and σ_W^2 within blocks, we can recover the inter-block information by weighting the stratum estimates proportional to their efficiency factors and inversely proportional to their variances. The resulting estimator, $\hat{\beta}_{SC}$ say, which we shall call the simply combined estimator of β, is:

$$\hat{\beta}_{SC} = \{e_B\hat{\beta}_B/\sigma_B^2+e_w\hat{\beta}_W/\sigma_W^2\}/\{e_B/\sigma_B^2+e_w/\sigma_W^2\}. \tag{1}$$

Of course $\hat{\beta}_{SC}$ is unbiased, and the variance of the estimator $\mathbf{t}'\hat{\beta}_{SC}$ of any estimable contrast $\mathbf{t}'\beta$ is:

$$\mathrm{Var}(\mathbf{t}'\hat{\beta}_{SC}) = \mathbf{t}'M^{-}\mathbf{t}/\{e_B/\sigma_B^2+e_W/\sigma_W^2\} \tag{2}$$

where M^{-} is the generalised inverse of M.

3. The Efficiency of the Simply Combined Estimator

All this is perfectly valid, but unfortunately $\hat{\beta}_{SC}$ does not recover all the inter-block information available. In more technical language, $\hat{\beta}_{SC}$ is not the best linear unbiased estimator (BLUE) of β and is not fully efficient; that is to say, there is another unbiased estimator of β which has less variance. The ratio of these variances is the efficiency of the Genstat analysis that I wish to study.

Why is $\hat{\beta}_{SC}$ not fully efficient? The reason is that in general the fully efficient estimator of the effects of both *T1* and *T2* must combine information from all estimated effects, not just from the effects separately. One method of doing this is to try to transform (or reparameterise) the model so that all effects are orthogonal in all strata. If this can be done, the design is said to be generally balanced as defined by Speed [5]. This can always be done for any design with just two strata. After this transformation, inter-block information can be recovered by a weighted average as in (1), then transformed back to the original parameters. The resulting estimator $\hat{\beta}$ is fully efficient. Conversely, if the fully efficient estimator can be formed in this way, then all treatment contrasts are said to be simply combinable, and Speed [5] proves that the design must be generally balanced.

We now turn to the efficiency of $\hat{\beta}_{SC}$ relative to $\hat{\beta}$. This is usually calculated for the equal-variance or white-noise case of $\sigma_B^2 = \sigma_W^2 = \sigma^2$, say. Then $\hat{\beta}$ can be found quite simply by ignoring the blocks. For the design (D1) and the commands

```
BLOCKS
TREATMENTS T1 + T2
```

we get:

```
***** ANALYSIS OF VARIANCE *****

SOURCE OF VARIATION              DF

*UNITS* STRATUM
  T1                              4
  T2                              4
  RESIDUAL                       21
TOTAL                            29

GRAND TOTAL                      29

***** INFORMATION SUMMARY *****

MODEL TERM                       EF   NON-ORTHOGONAL TERMS

*UNITS* STRATUM
  T2                          0.972   T1
```

As we can see, $T2$ is still non-orthogonal to $T1$ with efficiency factor $e = 0.972$. Hence

$$\mathrm{Var}(\mathbf{t}'\hat{\beta}) = \mathbf{t}'M^{-}\mathbf{t}\sigma^2/e \qquad (3)$$

The efficiency e_{SC} of $\mathbf{t}'\hat{\beta}_{SC}$ relative to $\mathbf{t}'\hat{\beta}$ is

$$e_{SC} = \mathrm{Var}(\mathbf{t}'\hat{\beta})/\mathrm{Var}(\mathbf{t}'\hat{\beta}_{SC}) = (e_B+e_W)/e = 0.858. \qquad (4)$$

which is less than one. Since this does not depend on which contrast $\mathbf{t}$ we take, we can refer to it as the efficiency of $\hat{\beta}_{SC}$ relative to β.

4. Designs Where the Simply Combined Estimator is Fully Efficient

This inefficiency is not just caused by the non-orthogonality of the treatment factors. The following is an example where $\hat{\beta}_{SC}$ is fully efficient, even though the factors are non-orthogonal. The two treatment factors $T1$ (upper case letters) and $T2$ (lower case letters) are applied to 12 units in 4 blocks:

Block			
1	2	3	4
Aa	*Ab*	*Ac*	*Aa*
Ba	*Bb*	*Bc*	*Bb*
Ca	*Cb*	*Cc*	*Cc*

(D2)

For design (D2) the output from the commands

```
BLOCKS block
TREATMENTS T1 + T2
```

is:

```
***** ANALYSIS OF VARIANCE *****

SOURCE OF VARIATION              DF

BLOCK STRATUM
  T2                              2
  RESIDUAL                        1
TOTAL                             3

BLOCK.*UNITS* STRATUM
  T1                              2
  T2                              2
  RESIDUAL                        4
TOTAL                             8

GRAND TOTAL                      11
```

```
***** INFORMATION SUMMARY *****

MODEL TERM                          EF   NON-ORTHOGONAL TERMS

BLOCK STRATUM
  T2                             0.750

BLOCK.*UNITS* STRATUM
  T2                             0.188   BLOCK  T1
```

The treatment factors are non-orthogonal with the efficiency of *T2* equal to $e_B = 0.75$ between blocks and $e_W = 0.1875$ within blocks. When the blocks are removed we get:

```
***** ANALYSIS OF VARIANCE *****

SOURCE OF VARIATION                 DF

*UNITS* STRATUM
  T1                                 2
  T2                                 2
  RESIDUAL                           7
TOTAL                               11

GRAND TOTAL                         11

***** INFORMATION SUMMARY *****

MODEL TERM                          EF   NON-ORTHOGONAL TERMS

*UNITS* STRATUM
  T2                             0:937   T1
```

The efficiency of *T2* is now $e = 0.9375$, giving $e_{SC} = (e_B + e_W)/e = 1$. Hence the interblock information is fully recovered by $\hat\beta_{SC}$, and so $\hat\beta_{SC} = \hat\beta$, at least for the case of equal stratum variances. It can be shown (see Houtman and Speed [2], Proposition 4.1) that $e_{SC} = 1$ is necessary and sufficient for full efficiency of $\hat\beta_{SC}$ even when the stratum variances are unequal. However, if the roles of *T1* and *T2* are reversed, then it can be checked that the efficiency for *T1* is $e_{SC} = 0.8$, so that the simply combined estimator for *T1* is not fully efficient. In fact it can be shown that the fully efficient estimator of the effect α of *T1* is

$$\hat\alpha = \hat\alpha_W + (\hat\beta_W - \hat\beta)/4 \qquad (5)$$

where $\hat\alpha_W$ is the within-blocks estimator of α and $\hat\beta = \hat\beta_{SC}$ is the fully efficient estimator of the effect β of *T2* found by the weighted average of $\hat\beta_B$ and $\hat\beta_W$ as in (1).

It is not hard to find a design that does allow fully efficient recovery of inter-block information by a weighted average for both factors, even though they are not orthogonal. The two treatment factors *T1* (upper case letters) and *T2* (lower case letters) are applied to 30 units in 6 blocks:

Block

1	2	3	4	5	6
Aa	*Bb*	*Cc*	*Ac*	*Ba*	*Cb*
Bb	*Cc*	*Ac*	*Ba*	*Cb*	*Aa*
Cc	*Ac*	*Ba*	*Cb*	*Aa*	*Bb*
Ac	*Ba*	*Cb*	*Aa*	*Bb*	*Cc*
Ba	*Cb*	*Aa*	*Bb*	*Cc*	*Ac*

(D3)

For either factor, $e_B = 0.30$, $e_w = 0.72$ and $e = 0.75$, giving $e_{SC} = 1$. This example (D3) was in fact concocted from a Youden square for six levels by grouping the levels in two different ways. The efficiency of all contrasts in the six levels, including contrasts in *T1* and *T2*, is $e_W/e = 0.96$ within blocks. If the third and fifth rows of (D3) are deleted then the resulting design has the same properties as (D2): full efficiency for *T2*, but not for *T1*.

5. A Design Where the Simply Combined Estimator has Zero Efficiency

It should be pointed out that in practice the inefficiency of the simply combined estimator is not very serious. First of all, σ_B^2 is usually much greater that σ_W^2 in which case $\hat{\beta}_W$ is almost as efficient as $\hat{\beta}$ or $\hat{\beta}_{SC}$. Moreover, the sampling properties of $\hat{\beta}$ or $\hat{\beta}_{SC}$ are not as straightforward as those of $\hat{\beta}_B$ or $\hat{\beta}_W$ when σ_B^2 and σ_W^2 are estimated by the between and within blocks means squared error. For these two reasons, a small amount of efficiency is usually sacrificed for the sake of simplicity.

However it is possible to find an extreme example where there is no within or between block information about either factor, adjusted for the other, so that the simply combined estimator does not exist, and yet there is information when the blocks are removed, that is, $e_{SC} = 0$. Consider the following design with the same factors as (D1) but with two observations per block instead of three:

Block

1	2	3	4	5	6	7	8	9	10
Ab	*Bc*	*Cd*	*De*	*Ea*	*Ac*	*Bd*	*Ce*	*Da*	*Eb*
Ba	*Cb*	*Dc*	*Ed*	*Ae*	*Ca*	*Db*	*Ec*	*Ad*	*Be*

(D4)

The first row contains all pairs of letters, and the second row has them reversed. The commands

```
BLOCKS Block
TREATMENTS T1 + T2
```

give:

```
***** ANALYSIS OF VARIANCE *****

SOURCE OF VARIATION              DF

BLOCK STRATUM
  T1                              4
  RESIDUAL                        5
TOTAL                             9

BLOCK.*UNITS* STRATUM
  T1                              4
  RESIDUAL                        6
TOTAL                            10

GRAND TOTAL                      19

***** INFORMATION SUMMARY *****

MODEL TERM                       EF   NON-ORTHOGONAL TERMS

BLOCK STRATUM
  T1                          0.375

BLOCK.*UNITS* STRATUM
  T1                          0.625 BLOCK

ALIASED MODEL TERMS
  T2
```

The effect of *T2* is aliased with that of *T1* in both strata, and yet when blocks are removed *T2* is estimable:

```
***** ANALYSIS OF VARIANCE *****

SOURCE OF VARIATION              DF

*UNITS* STRATUM
  T1                              4
  T2                              4
  RESIDUAL                       11
TOTAL                            19
```

```
GRAND TOTAL                     19

***** INFORMATION SUMMARY *****

MODEL TERM                      EF  NON-ORTHOGONAL TERMS

*UNITS* STRATUM
  T2                         0.937  T1
```

Precisely the same results are obtained if *T*1 and *T*2 are fitted in reverse order, or even in fact if (D4) is treated as a row-column design. This example has the rather surprising property that the two estimable treatment factors are not estimable in any of the strata, adjusting for each other. It is also rather curious that the fully efficient estimator $\hat{\beta}$ of the effect of *T*2 adjusting for *T*1 does not depend on the stratum variances σ_B^2 or σ_W^2. As a result $\hat{\beta}$ and its sum of squares can be found from the second analysis with the blocking factor removed. However the variance of $\hat{\beta}$ and the expected sums of squares do depend on the stratum variances. It can be shown that the null expectation of the mean sums of squares for *T*2 is $(5\sigma_B^2+3\sigma_W^2)/8$. The orthogonal treatment structure of (D4) is the same as that of (D1).

6. References

[1] Ceranka, B. and Mejza, S.
Analysis of diallel table for experiments carried out in BIB designs-mixed model.
Biometrical Journal, **30**, pp. 3-16, 1988.

[2] Houtman, A.M. and Speed, T.P.
Balance in Designed Experiments with Orthogonal Block Structure.
Ann. Statist., **11**, pp. 1069-1085, 1983.

[3] Preece, D.A.
Some Balanced Incomplete Block Designs for Two Sets of Treatments.
Biometrika, **53**, pp. 497-506, 1966.

[4] Preece, D.A.
Genstat Analyses for Complex Balanced Designs with Non-interacting Factors.
Genstat Newsletter, **21**, pp. 33-45, 1988.

[5] Speed, T.P.
General Balance.
Encyclopedia of Statistics (S. Kotz, N. Johnson and C.B. Read, Eds.), Wiley, New York, **3**, pp. 320-326, 1983.

Using the EDIT Directive to Deal with Awkward Textual Data

T R Butler-Stoney
AFRC Institute of Grassland and Environmental Research
Welsh Plant Breeding Station
Plas Gogerddan
Aberystwyth
Dyfed SY23 3EB

The EDIT directive does not seem to be much used, so this application may serve to show how it can be useful when dealing with awkward textual data.

The program below is really a cut-down version of a longer program; in practice data files are longer and there are more columns. It reads data output from a data-base system (Vax Datatrieve), which I use to store chemical results. In the example, the input file contains information on the Datatrieve 'domains' that store sample descriptions and chemistry results crossed over sample-number.

To read the data as text, Genstat must use fixed format because some of the strings contain non-alphanumeric characters and are not quoted. It would be possible to edit quotes (') into the data file, but this would have to be done each time Datatrieve is run. Unfortunately, variety names (accessions) are not justified, so the strings that should correspond to the same variety are treated as distinct because of leading blanks found in the fixed format. I use EDIT to remove these leading blanks. The method here can be included in a Genstat procedure and used in all programs using this type of data.

```
   1  TEXT accession
   2  VARIATE sampleno,oil,protein
   3  READ [PRINT=data,error,summary; SETNVALUES=yes; LAYOUT=fixed; SKIP=*] \
   4     sampleno,accession,oil,protein ; FIELDWIDTH= \
   5     5,        17,        6,  6

   6  04444 84-93Cn          09.20 12.75
   7  04444 84-93Cn          08.23 11.19
   8  04445       84-12Cn    07.94 15.38
   9  04446       84-12Cn    08.09 13.19
  10  04447       84-12Cn    07.95 12.56
  11  04448       84-12Cn    09.18 12.94
  12  04454        Pennal    06.00 10.75
  13  04454        Pennal    05.67 09.75
  14  04455        Lustre    06.75 09.69
  15  04456       84-14Cn    06.58 12.56
  16  04462         84-176Cnl 07.58 12.19
  17  04463         Pennal   06.23 10.63
  18  04464         Image    06.77 09.38
  19  04476 Pennal           06.14 10.88
  20  :

    Identifier   Minimum      Mean   Maximum     Values   Missing
      sampleno      4444      4454      4476         14         0
           oil     5.670     7.308     9.200         14         0
       protein      9.38     11.70     15.38         14         0
  21
  22  "When the text is used to form a factor, the leading spaces are
 -23   significant and cause Pennal to be treated as three varieties."
  24  SORT [INDEX=accession; GROUPS=factor; LABELS=name1]
  25  TABULATE [PRINT=nobs,mean,min,max; CLASS=factor] protein

                      Nobservd        Mean     Minimum      Maximum
          factor
 Pennal                      1       10.88       10.88        10.88
       Lustre                1        9.69        9.69         9.69
       Pennal                2       10.25        9.75        10.75
        Image                1        9.38        9.38         9.38
        Pennal               1       10.63       10.63        10.63
        84-176Cnl            1       12.19       12.19        12.19
      84-12Cn                4       13.52       12.56        15.38
      84-14Cn                1       12.56       12.56        12.56
 84-93Cn                     2       11.97       11.19        12.75
```

```
"Strip the leading spaces using EDIT."
EDIT [!t('g/ // :')] accession
"Sort and tabulate again and all 'Pennal's are combined."
SORT [INDEX=accession; GROUPS=factor; LABELS=name1]
TABULATE [PRINT=nobs,mean,min,max; CLASS=factor] protein

                  Nobservd        Mean     Minimum     Maximum
     factor
      Image              1        9.38        9.38        9.38
     Lustre              1        9.69        9.69        9.69
     Pennal              4       10.50        9.75       10.88
    84-12Cn              4       13.52       12.56       15.38
    84-14Cn              1       12.56       12.56       12.56
  84-176Cn1              1       12.19       12.19       12.19
    84-93Cn              2       11.97       11.19       12.75
```

The EDIT command

```
g/ //
```

is a global instruction to replace all occurrences of the space character by nothing. Thus, all the blanks are removed from the strings stored in the text structure called accession.

This application of Genstat was developed during the course of work on exploitation of the genetic potential of oats for use in feed and human nutrition which was funded by the Home-Grown Cereals Authority.

Combining Tables with Variates

Dr R Sackville Hamilton
Unit of Plant Population Biology
School of Biological Sciences
UCNW
Bangor LL57 2UW

A problem that frequently occurs in manipulating data with a hierarchical structure is combining tables with variates. Suppose, for example, that in a blocked experiment one wishes to standardize the values of a variate to deviations from the block means, stored in a table. The problem is that the *n*th element of the variate corresponds not to the *n*th element of the table but to the element of the table corresponding to the *n*th elements of the factors classifying the table. Say that the block means are in table Bmean as follows:

```
Bmean
Block    1     2     3
         12.1  15.5  16.0
```

The required variate V should then have values depending on the values of the block factor Block as follows:

```
Unit  Block   V
 1      1   12.1
 2      3   16.0
 3      2   15.5
 4      3   16.0
 5      1   12.1
 6      2   15.5
```

SAS users will know how easily this can be done in SAS, using the MERGE and BY statements of the DATA step, with the table(s) in one dataset, the variate(s) in a second, and the factor(s) (the BY variables) in both.

There would be no problem in Genstat if it were possible to qualify a table identifier, since the variate V could be formed by the single statement:

```
CALCULATE V = Bmean$[Block]
```

However, the CALCULATE directive does not yet allow qualification of tables. The following procedure serves as a substitute.

```
PROCEDURE 'TEXPAND'
"
         Take a table classified by any number of factors and return a
         variate. The variate will have the same number of values as each
         of the classifying factors, and will contain on exit the values of
         the table cells corresponding to the values of the factors.
"
   PARAMETER 'TABLE',       "I: table containing values to be expanded" \
             'FACTORS',     "I: pointer to the factors classifying TABLE" \
             'VARIATE'      "O: variate to contain the expanded table on exit"

   CALCULATE Nv = NVALUES(FACTORS[1])
   VARIATE [VALUES=(1)#Nv] VARIATE
   "
         We need the ordinal levels of the factors, not their actual levels;
         hence the use of the NLEVELS and NEWLEVELS functions.
   "
   FOR F=FACTORS[]
      CALCULATE N1 = NLEVELS(F)
      & VARIATE = (VARIATE-1)*N1 + NEWLEVELS(F; !(1...N1))
   ENDFOR
   "
         Cannot qualify a table, so put its values into an unnamed variate.
   "
   CALCULATE VARIATE = !(#TABLE)$[VARIATE]
   ENDPROCEDURE
```

The parameter FACTORS is necessary because it is not possible in Genstat 5 Release 1.3 to determine what factors classify a table. In Release 2.1, the GETATTRIBUTE directive can access this information. For example,

```
GETATTRIBUTE [ATTRIBUTE=classification] TABLE; SAVE=Pfac
```

sets up a pointer called Pfac with one value which is itself a pointer to the classifying factors of the table.

Genstat 5 has no simple equivalent of Genstat 4's FLOAT function: hence the need for the FOR loop. If the default action of NEWLEVELS were equivalent to FLOAT, the FOR loop could be simplified to:

```
CALCULATE Nd = NVALUES(FACTORS)
CALCULATE (VARIATE)#Nd = (VARIATE-1)*NLEVELS(FACTORS[]) + \
                         NEWLEVELS(FACTORS[])
```

To keep the code simple, no attempt has been made to do any error-checking within the procedure, even though there are abundant opportunities for errors; for example, making sure that `FACTORS` agrees with the actual table classification, checking that the table has no margins, and so on.

Example

A set of measurements is made on each leaf on the main stem of a number of plants. The leaves are numbered sequentially, from 1 at the base of the stem to `Nleaves` at the top. Each plant may have a different total number `Nleaves` of leaves.

It is considered that the proximity of a leaf to the top of the stem may be important, so we need to reverse the leaf-numbering system to give `N_to_top`, with values ranging from 0 for the top-most leaf to `Nleaves-1` for the leaf at the base. This is done by subtracting the variate `Leaf_no` from the table `Nleaves`.

```
FACTOR [NLEVELS=Nplants] Plant
VARIATE Leaf_no
READ [CHANNEL=2] Plant,Leaf_no
      "
      Note: tabulate Nleaves as the maxima of Leaf_no, not as counts,
            in case of missing data.
      "
TABULATE [CLASSIFICATION=Plant] Leaf_no; MAXIMA=Nleaves
TEXPAND Nleaves; FACTORS=!p(Plant); VARIATE=Top_leaf
CALCULATE N_to_top = Top_leaf - Leaf_no
```