NAG®

GENSTAT

Newsletter

Issue No. 26

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

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. 26

NP2131

April 1991

Contents

			Page
1.	Editorial		3
2.	One-Day Conference on Interactive Statistical Modelling, 26 April 1990	*R Butler*	4
3.	Conditional Logistic Regression in Genstat	*J Thompson*	6
4.	Genstat ANOVA Efficiency Factors and Canonical Efficiency Factors for Non-Orthogonal Designs	*K J Worsley, G P H Styan and J Bérubé*	11
5.	Some Comparisons Using Genstat on PC's	*E R Williams*	22
6.	Versions of Genstat 5 for Personal Computers	*P G N Digby*	27
7.	Combining Tables with Variates – An Alternative Approach	*H R Simpson*	29
8.	Explicit and Implicit Loops	*P W Lane*	31
9.	Editing Data Structures	*P W Lane*	33
10.	Fitting Non-linear Models and Estimating Functions of Model Parameters	*M Patefield*	36

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

Editorial

With the continuing trend towards using Genstat on personal computers, two of the articles in this issue should be of interest to many readers. In the first, a user gives detailed comparisons of performance of the various existing versions; in the second, an implementor responds and gives information about future versions.

This issue starts with a summary of the one-day Conference held at Rothamsted last year on interactive statistical modelling, which just missed the last issue. There is also a short response to the article in that issue about combining tables with variates.

There are detailed articles on practical aspects of three different statistical applications. One shows how to use a procedure to improve estimates of standard errors in nonlinear regression, and a second introduces a procedure for conditional logistic regression. The third deals with efficiency factors and general balance in the analysis of designed experiments.

Finally, two articles have been written by one of the editors during a pleasant term as visiting fellow at the Australian National University. They describe a procedure to allow the use of any screen editor to edit Genstat data structures, and the benefits of using implicit rather than explicit loops.

Genstat News

We are pleased to be able to announce that a new Head has been appointed to the Statistics Department at Rothamsted. John Gower retired in April 1990, and since then the Department has undergone a review of its activities to establish the future course of Biomathematics within the whole of the Agricultural and Food Research Council. The development of Genstat was recognized as an important part of the Department's work, and will continue to be supported and encouraged. Professor Vic Barnett was appointed as the new Head of the Biomathematics Division, including the Statistics Department, from 1 May 1991. With his support, Genstat will continue to be run by the Genstat Committee within a strengthened Statistics Department.

Professor Barnett retains his Chair in Probability and Statistics at Sheffield University, which he has held since 1975. He previously served as a member of staff in the Universities of Bath, Newcastle, Western Australia, Birmingham and Manchester, and has been active on major committees of the Royal Statistical Society and the Institute of Statisticians. As well as this mainstream of statistical education and research, he has extensive experience in statistical consultancy and in computing, particularly in his most recent post at Sheffield University as Director of Information Technology. Professor Barnett has agreed to write a short article for the next issue of this Newsletter, in which we hope he will outline his views on statistical computing and Genstat.

Implementation News

As well as the DEC VAX/VMS implementation of Release 2 there is now an implementation for the Sun 3 and implementions for the HP9000/800, the Sequent Symmetry and the IBM RS 6000 should be available shortly.

Introductory Course

The next Genstat Introductory Course is now being arranged and will take place in Nottingham from 29 October to the 1 November 1991. For further details contact Lesley Austen at NAG.

One-Day Conference on Interactive Statistical Modelling, 26 April 1990

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

This conference continues the series of one-day Genstat conferences held on specific topics; as before, the venue was the Conference Hall at Rothamsted. Over 60 participants, mainly from the UK, listened to six talks on Interactive Genstat use, and were also able to see demonstrations of the new fast version of Release 1.3 for 80386-based PCs, and test versions of Release 2.1 for Vax/VMS and Sun 3.

The Conference was opened by Roger Payne answering the question 'Why Work Interactively?' in a talk prepared jointly with John Gower, recently retired head of Biometrics at Rothamsted. He suggested that interactive use of computers allowed a return to the immediate experience of the data that used to be commonplace when statistics were done 'by hand' or with pocket calculators, but with the added benefits of the new powerful tools available in packages such as Genstat. Interactive use of statistics allows immediate response to the results of an analysis, and encourages the user to experiment and to look at plots and tables without the need for the production of a hard copy. Genstat 5 has many facilities that are useful for this, such as high-resolution graphics, commands (such as RDISPLAY and ADISPLAY) to recall results of the last analysis, backing-store to save structures, the COPY command to produce transcripts of a session, and procedures and macros to store commonly used groups of commands.

The facilities outlined in the first talk were used extensively by Pete Digby and Karen Moore (both from Rothamsted) in their analyses described in the second talk. Pete Digby outlined the analysis of data from experiments in which many aspects of several varieties of potato plants and their tubers were recorded. He showed how multivariate techniques could be used to find groupings amongst the varieties and to show which were the major aspects that determined these groupings.

Lunch was followed by four more sessions. The first of these was a gallant effort by Rodger White, in the absence of the main authors Keith Bicknell and Simon Harding (all three from Rothamsted), who described the interactive potential of the new high-resolution-graphics command DREAD (pronounced 'deereed' rather than 'dred'!). This allows information to be read from a graph produced on a screen by Genstat, and procedures can be developed using DREAD to enable interactive modification of such pictures. Rodger illustrated a procedure called ZOOM which uses DREAD to outline an area of a graph and then produce a new graph of this area on a larger scale.

This talk was followed by an illustration of the interactive use of Time Series analysis by Granville Tunnicliffe Wilson (Lancaster University), which he uses in teaching. He was assisted by Mario Ferrelli at the keyboard of a PC, whose display was projected onto a screen. High-resolution graphics were shown on separate overheads, because the hardware needed to project them from the computer was not available. This talk showed clearly how 'utility' procedures could be used to facilitate the reading, storing and cataloguing of data in a regular format, and how easy it is to use the results of one model fitted in conjunction with pictures to suggest an instant modification of the model. Students using this approach can easily gain insights into the uses and theory of time-series analysis.

The teaching theme was continued in the fifth talk by Graham Horgan from the Scottish Agricultural Statistics Service, who discussed the way SASS runs service courses for researchers, and how the teaching of Genstat is carried out as part of these. Interactive use of Genstat by beginners enables faster learning than does batch use because mistakes can instantly be seen and corrected. He suggested that initial computing skills need to be taught separately from statistics – hence the decision by SASS to run a two-day introductory Genstat course which concentrates mainly on syntax and language.

The final talk by Peter Lane (Rothamsted) described the 'menu' system developed for Release 2.1 that allows Genstat to be used by people with no knowledge of Genstat. This uses two new additions to Genstat: a QUESTION directive, which allows a response to be given and recorded, and a 'start-up' file of Genstat statements that is run when Genstat is invoked. The system consists of a set of interlinked menus, starting with a base menu which leads to input, calculation, tabulation, picture and

analysis menus. The analysis menu leads to sub-menus covering most areas of analysis. Peter stressed that this system was by no means a definitive version, and that he was open to suggestions for its modification, also showing how users could modify the existing system to suit their own needs.

This conference illustrated the powerful potential of Genstat 5 as an interactive tool given the availability of suitable hardware, and pointed the way towards new methods of working with Genstat 5.

Conditional Logistic Regression in Genstat

J Thompson
Department of Ophthalmology
Clinical Sciences Building
University of Leicester
PO Box 65
Leicester LE2 7LX

1. Introduction

The analysis of a case-control study by a logistic regression model may be approached in one of two ways. When the strata contain large numbers of cases and controls then a simple logistic regression of the number of cases as a proportion of the total of cases and controls will yield good estimates. However if data are sparse, then a conditional approach has been shown to be preferable. See Breslow and Day [1] and the references therein for a general discussion.

The conditional approach involves the maximisation of a Conditional Likelihood comprising the product over strata of terms of the type,

$$\frac{\Sigma\{\exp(b'x_i)\}}{\Sigma\{\prod\{\exp(b'x_i)\}\}}$$

where b is a vector of parameters and x_i is the vector of covariates for the ith person. If there are m cases in n controls, the upper sum is over all m cases and the lower sum is of all products of m subjects that could be chosen from the $m + n$ available. The term may be thought of as the probability that those particular m subjects succumbed to the disease given that m of the $m + n$ were going to.

Maximising the Conditional Likelihood or its log is made difficult by the lower sum of products which can be very time consuming for a computer to evaluate if m and n are large.

Below we present a Genstat procedure that maximises the log Conditional Likelihood using a time-saving recursive method described by Krailo and Pike [3].

The conditional likelihood is of the same form as the Partial Likelihood suggested by Cox [2] for analysing survival data by the proportional hazards model. Indeed the analysis of case-control studies may be approached via that model. The procedure given may, with minor modifications, be used to analyse survival data. When ties in the survival times occur due to grouping, the use of this procedure is equivalent to using the approximate method suggested by Cox in his original paper and with small data sets should perform better than the frequently employed but less accurate approximation suggested by Peto in his discussion to Cox's paper.

2. The Procedure

Sums of combinations of terms may be obtained in Genstat by using the CUMULATE, SHIFT, MVREPLACE functions together with the multiplication of two variates. Suppose that we start with a variate containing four elements, denoted in Table 1 by (a, b, c, d), and then follow the operations set out in the table. It will be seen that the last element of the variate at the end of each block is one of the required sums of products.

If a, b, c and d are exponentials, then similar simple combinations of the CUMULATE, SHIFT and MULTIPLY operations may be used to produce the first and second derivatives of the sums of products. These operations may be found programmed into the procedure given in the Appendix. The derivatives so obtained may then be used to calculate the derivatives of the log conditional likelihood. Formulae for each of these stages may be found in Krailo and Pike [3].

ORIGINAL	*a*	*b*	*c*	*d*
CUMULATE	*a*	*a+b*	*a+b+c*	*a+b+c+d*
SHIFT	0	*a*	*a+b*	*a+b+c*
MULTIPLY	0	*ab*	*ac+bc*	*ad+bd+cd*
CUMULATE	0	*ab*	*ab+ac+bc*	*ab+ac+bc+ad+bd+cd*
SHIFT	0	0	*ab*	*ab+ac+bc*
MULTIPLY	0	0	*abc*	*abd+acd+bcd*
CUMULATE	0	0	*abc*	*abc+abd+acd+bcd*
SHIFT	0	0	0	*abc*
MULTIPLY	0	0	0	*abcd*
CUMULATE	0	0	0	*abcd*

Table 1

The operations needed to produce sums of all products. `MULTIPLY` refers to multiplication by the original variate and `SHIFT` requires the combination of a shift to the right and the insertion of zero in the first element. That is `MVREPLACE(SHIFT(x; 1); 0)`

3. A Case-Control Example

Marshall *et al.* [4] present the results of a case-control study of an occupational cluster of leather workers with testicular cancer. An extract from their results is shown in Table 2.

Age Stratum		Leather Worker	
		Yes	No
20-25	Cases	2	5
	Controls	1	13
26-40	Cases	2	6
	Controls	1	36
41-54	Cases	1	2
	Controls	16	192

Table 2

Leather work among cases and controls in three age strata

To analyse these data using the conditional method requires the following Genstat code.

```
SCALAR ns,np; VALUE=3,1
FACTOR [LEVELS=3; VALUES=21(1),45(2),211(3)] stratum
VARIATE [VALUES=7(1),14(0),8(1),37(0),3(1),208(0)] status
VARIATE [VALUES=2(1),5(0),1,13(0),2(1),6(0),1,36(0),1,2(0), \
          16(1),192(0)] x [1]
VARIATE [VALUES=1(0)] deriv,coef
SYMMETRICMATRIX [ROWS=1; VALUES=1(0)] cov
CONDML ns; stratum; status; np; x; coef; logL; deriv; cov
```

The resulting estimated coefficient is 1.93 with a standard error of 0.73. This compares with 1.99 (0.74) by unconditional logistic regression. As might be anticipated, the difference is not great but it will be larger when the strata are smaller and more numerous, as for example with matched studies.

4. A Survival Example

To illustrate the method we analyse the data of Freireich *et al.* quoted in Cox [2] and frequently used since to demonstrate survival analyses. The data, presented in Table 3, refer to the remission times of two samples of leukemia patients one treated with a drug and the other with a placebo.

TREATED	6*	6	6	6	7	9*	10*
	10	11*	13	16	17*	19*	20*
	22	23	25*	32*	32*	34*	35*
PLACEBO	1	1	2	2	3	4	4
	5	5	8	8	8	8	11
	11	12	12	15	17	22	23

Table 3

Remission times (weeks) of two groups of leukemia patients.
Asterisks denote censored values

To analyse these date we need to supply the procedure with the remission times (TIME), censoring details (STATUS) and a list of the times when an event actually occurred (OBSTIME), rather than with a list of strata. The procedure should then be modified as given in the appendix.

Analysis may then be obtained by,

```
VARIATE [VALUES=0,1,1,1,1,0,0,1,0,1,1,3(0),     \
                1,1,5(0),21(1)] status
VARIATE [VALUES=21(0),21(1)] x[1]
VARIATE [VALUES=6,6,6,6,7,9,10,10,11,13,16      \
                17,19,20,22,23,25,32,32,34,35,  \
                1,1,2,2,3,4,4,5,5,8,8,8,8,11,   \
                11,12,12,15,17,22,23] time
VARIATE [VALUES=1,2,3,4,5,6,7,8,10,11,12,       \
                13,15,16,17,22,23] obstime
SCALAR ns,np; VALUE=17,1
VARIATE [VALUES=1(0)] deriv,coef
SYMMETRICMATRIX [ROWS=1; VALUES=1(0)] cov
CONDML ns; obstime; status; np; x; coef; time; logL; deriv; cov
```

The resulting estimate for the effect of treatment is 1.63 (standard error 0.43) compared to 1.51 (0.41) using Peto's approximation.

5. Conclusions

No attempt has been made to make the procedure robust and so users must take responsibility for their own data checks. With larger data sets it may be necessary to divide the original variate of exponentials by a suitably chosen constant to avoid overflow in the sums of products.

Although the recursive algorithm is more efficient than nested loops, it is still quite slow in Genstat. No doubt the code for the procedure could be made more efficient but the speed will always be limited by the way Genstat handles procedures. Using Release 1.3 of Genstat 5 on a 20MHz 386 the case-control example required four iterations to converge with each iteration taking 40 seconds. The survival example required three iterations of 90 seconds each.

For large problems it might be preferable to program the algorithm in Fortran and link it to Genstat using one of the methods for extending the package. In the longer term it is to be hoped that Genstat will eventually contain its own efficient algorithm for maximising Partial or Conditional Likelihoods.

6. References

[1] Breslow, N.E. and Day, N.E.
Statistical Methods in Cancer Research : Volume 1.
The Analysis of Case-Control Studies.
IARC, Lyon, 1980.

[2] Cox, D.R.
Regression Models and Life-Tables (with discussion).
J. R. Stat. Soc. B, 34, pp. 187-220, 1972.

[3] Krailo, M.N. and Pike, M.C.
Algorithm AS196: Conditional Multi-variate Logistic Analysis of Stratified Case-Control Studies.
Appl. Stat., p. 95, 1984.

[4] Marshall, E.G., Melius, J.M., London, M.A., Nasca, P.C. and Burnett, W.S.
Investigation of a Testicular Cancer Cluster using a Case-Control Approach.
Int. J. Epidemiol., 19, (2), pp. 269-273, 1990.

7. Appendix

```
PROCEDURE 'CONDML'
          PARAMETERS NAME = 'NS', 'STRATUM', 'STATUS', 'NP', 'X',        \
                                'CF', 'LOGL', 'DV', 'H'; MODE = p
   CALCULATE nh = NP*(NP+1)/2 : & m = NOBSERVATIONS(STRATUM)
   SCALAR [VALUE=0] cend[1...#NP], dend[1...#nh], is, i, j,              \
                    s[1...#NP], bend
   VARIATE [NVALUES=#m] a : & [NVALUES=#NP] delta

   FOR [NTIMES=10]
      SCALAR [VALUE=0] i, is
      CALCULATE a,DV,H = 0
      FOR [NTIMES=NP]
         CALCULATE i = i + 1 : & j = CF$[i] : & a = a + j*X[]
      ENDFOR
      CALCULATE LOGL = SUM(a*(STATUS==1)) : & a = EXP(a)
      FOR [NTIMES=NS]
         CALCULATE is = is + 1
         RESTRICT X[], STATUS, a; CONDITION = (STRATUM==is);             \
                  SAVESET = save
         RESTRICT X[], STATUS, a
         VARIATE pSTATUS; VALUES = !(#STATUS$[save])
         VARIATE pa; VALUES = !(#a$[save])
         SCALAR [VALUE=0] i
         FOR [NTIMES=NP]
            CALCULATE i = i + 1
            VARIATE pX[i]; VALUES = !(#X[i]$[save])
         ENDFOR
         CALCULATE m = NOBSERVATIONS(pSTATUS)
             &      nc = SUM(pSTATUS) - 1
         VARIATE [VALUES=#m(0)] c[1...#NP], d[1...#NP]
         CALCULATE b = CUM(pa)
             &      c[] = CUM(pX[]*pa)
         SCALAR [VALUE=0] i, k
         FOR [NTIMES=NP]
            CALCULATE j = i : & i = i + 1 : & nk = NP - j
            FOR [NTIMES=nk]
            CALCULATE j = j + 1 : & k = k + 1
                &      d[k] = CUM(pX[i]*pX[j]*pa)
            ENDFOR
         ENDFOR
         IF nc > 0
            FOR [NTIMES=nc]
               CALCULATE b = MVREPLACE(SHIFT(b;1)*pa;0)
                   &      c[] = MVREPLACE(SHIFT(c[];1)*pa;0)
                   &      k,i = 0
               FOR [NTIMES=NP]
                  CALCULATE j = i : & i = i + 1 : & nk = NP - j
                  FOR [NTIMES=nk]
                     CALCULATE j = j + 1 : & k = k + 1
                         &      d[k] = MVREPLACE(SHIFT(d[k];1)*pa;0)     \
                                     + pX[i]*c[j] + pX[j]*c[i] + pX[j]*b
                  ENDFOR
               ENDFOR
               CALCULATE c[] = c[] + b*pX[]
               CALCULATE b,c[],d[] = CUM(b,c[],d[])
            ENDFOR
         ENDIF
```

```
            CALCULATE bend = b$[m] : & cend[] = c[]$[m]/bend
                &      dend[] = d[]$[m]/bend
                &      LOGL = LOGL - LOG(bend) : & k,i = 0
            FOR [NTIMES=NP]
               CALCULATE j = i : & i = i + 1 : & nk = NP - j
                   &      DV$[i] = DV$[i] + SUM(pX[i]*(pSTATUS==1)) - cend[i]
               FOR [NTIMES=nk]
                  CALCULATE j = j + 1 : & k = k + 1
                      &      H$[i;j] = H$[i;j] + dend[k] - cend[i]*cend[j]
               ENDFOR
            ENDFOR
            DELETE [REDEFINE=yes] b, c[], d[], pX[], pa, pSTATUS
         ENDFOR
         CALCULATE H = INV(H) : & delta = H*+DV : & CF = CF + delta
         PRINT [IPRINT=*;SQUASH=yes] 'Log Likelihood', LOGL; DEC = 3;          \
               FIELD = 14,7
         EXIT SUM(delta*delta) < 0.00001
      ENDFOR
ENDPROCEDURE
```

For the analysis of survival data the beginning of the procedure should be modified as follows.

```
PROCEDURE 'CONDML'
          PARAMETERS NAME = 'NS', 'OBSTIME', 'STATUS', 'NP', 'X',          \
                            'CF', 'TIME', 'LOGL', 'DV', 'H'; MODE = p
   CALCULATE nh = NP*(NP+1)/2 : & m = NOBSERVATIONS(TIME)
   SCALAR [VALUE=0] cend[1...#NP], dend[1...#nh], is, i, j,                 \
                    s[1...#NP], bend
   VARIATE [NVALUES=#m] a : & [NVALUES=#NP] delta

   FOR [NTIMES=10]
      SCALAR [VALUE=0] i, is, t
      CALCULATE a,DV,H = 0
      FOR [NTIMES=NP]
         CALCULATE i = i + 1 : & j = CF$[i] : & a = a + j*X[]
      ENDFOR
      CALCULATE LOGL = SUM(a*(STATUS==1)) : & a = EXP(a)
      FOR [NTIMES=NS]
         CALCULATE is = is + 1
             &      t = OBSTIME$[is]
         RESTRICT X[], STATUS, a; CONDITION = (TIME>=t); SAVESET = save
         RESTRICT X[], STATUS, a
         VARIATE pSTATUS; VALUES = !(#STATUS$[save])
         VARIATE pa; VALUES = !(#a$[save])
         VARIATE ptime; VALUES = !(#TIME$[save])
         CALCULATE pSTATUS = pSTATUS*(ptime==t)
         SCALAR [VALUE=0] i

etc.
```

Genstat ANOVA Efficiency Factors and Canonical Efficiency Factors for Non-Orthogonal Designs

K J Worsley, G P H Styan and J Bérubé
Department of Mathematics and Statistics
McGill University
805 ouest, rue Sherbrooke
Montréal
Québec
Canada H3A 2K6

1. Introduction

In a recent issue of the Genstat Newsletter Preece ([3] page 40), asked readers to deduce where the 'missing' information on a factor in a complex balanced block design was hidden. His design (5) is reproduced below:

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

(P5)

Two sets of treatments T1 and T3, each on five levels, indicated by upper and lower case letters, are applied to three observations in ten blocks. The model is additive in main effects for the treatments, with no interaction. A Genstat ANOVA can be specified by:

```
BLOCKS     block
TREATMENTS  T1+T3
```

The ANOVA table and information summary for some random data are as follows:

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

VARIATE: Y

SOURCE OF VARIATION               DF          SS        SS%          MS       VR

BLOCK STRATUM
  T1                               4      50.376      23.16      12.594     3.56
  RESIDUAL                         5      17.680       8.13       3.536
TOTAL                              9      68.056      31.29       7.562

BLOCK.*UNITS* STRATUM
  T1                               4      44.492      20.46      11.123     1.40
  T3                               4       9.904       4.55       2.476     0.31
  RESIDUAL                        12      95.029      43.70       7.919
TOTAL                             20     149.424      68.71       7.471

GRAND TOTAL                       29     217.480     100.00

GRAND MEAN                          5.05
TOTAL NUMBER OF OBSERVATIONS          30


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

MODEL TERM                          EF  NON-ORTHOGONAL TERMS

BLOCK STRATUM
  T1                             0.167

BLOCK.*UNITS* STRATUM
  T1                             0.833  BLOCK
  T3                             0.833  BLOCK
```

We note that T3 is not estimable between blocks, or in other words T1 and T3 are confounded on the block-averaged data. Moreover, after adjusting for blocks, the two sets of treatments T1 and T3 are orthogonal. Where has the missing information on T3 gone? An inspection of the design shows that treatment *a* appears in every block in which treatment *A* occurs, so that an effect due to *A* based on blocks alone can never be distinguished from an effect due to *a*. This is the same for all levels of the factors, and so the 'missing' information has been added to the residual. What seems more surprising is that T3 is not estimable with full efficiency within blocks, where it has an efficiency factor of 0.833 = 5/6, adjusted for T1.

Since T3 was not estimable between blocks one might have expected it to be estimated with full efficiency within blocks, that is with an efficiency factor of 1. Equivalently, why do the efficiency factors of T3 not add to 1 over strata? The same phenomenon occurs for the preceding design (4) given by Preece [3], which has the same layout of upper case letters for T1 but a different arrangement of lower case letters, denoted by T2:

				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*	*Ed*	*Ad*	*Be*	*Ca*	*Db*

(P4)

The Genstat efficiency factors for T2 adjusted for T1 are 0.093 between blocks and 0.741 within blocks, which again fall short of a sum of one.

2. General Balance

It may not be well known to Genstat users that Genstat ANOVA adopts slightly different definitions of balance and efficiency factors than those given by James and Wilkinson [2] and Houtman and Speed [1], which are nicely summarised in the Encylopedia of Statistical Sciences article by Speed [4]. Following the notation used in [4], we will say that the dispersion matrix V of a random vector y has 'orthogonal block structure' if it can be written as a linear combination of $k < n$ distinct known pairwise-orthogonal symmetric idempotent matrices $C_1,C_2,...,C_k$ which sum to the identity matrix I, that is $V = \sum\theta_i C_i$ for some (unknown) scalars $\theta_1,\theta_2,...,\theta_k$. The range or column space $R(C_i)$ of C_i is said to be the *i*th *stratum*, with C_i being the orthogonal projection onto this stratum. Suppose that $E(y) = X\beta$ and write the 'hat' matrix $H = X(X'X)-X'$. Then a design with orthogonal block structure is said to have *general balance* if the matrices $HC_1H,...,HC_kH$ commute.

In more practical terms, if the design matrix X has full column rank then the design is generally balanced if X can be transformed by a nonsingular matrix T to an orthonormal matrix $X* = XT$ with the property that $X*'C_iX* = \Lambda_i$, a diagonal matrix, for all $i = 1,2,...,k$. It can be shown that the columns of T are the eignevectors of $(X'X)^{-1}X'C_iX$; the components of Λ_i are the eigenvalues of $(X'X)^{-1}X'C_iX$ or equivalently of HC_iH, and are the 'canonical efficiency factors' of $T^{-1}\beta = X*'X\beta$ in stratum i. If such a matrix T exists, not necessarily unique, then the best linear unbiased estimator (BLUE) of $T^{-1}\beta$ is said to be simply combinable over strata, with weights proportional to Λ_i/θ_i. In a block design this means that the inter-block information about $T^{-1}\beta$ can be recovered by taking a simple weighted average of the inter- and intra-block estimators.

One consequence of this definition is that any design with just two strata is generally balanced, since then $C_2 = I - C_1$ and so HC_1H and HC_2H commute. So Preece's designs (P4) and (P5) are generally balanced. It is merely a question of identifying the linear transformation T of the design matrix X and the associated eigenvalues. Of course the rows of the parameter transformation T^{-1} may not be expressible as contrasts in the levels of a factor and so the design may not be analysable by Genstat ANOVA.

3. Canonical Correlations and Canonical Efficiency Factors

Following James and Wilkinson [2] we note that the within-blocks canonical efficiency factors – the diagonal elements of the matrix Λ_2 – may be interpreted in terms of certain canonical correlations. Let us assume for the moment that the block effects are fixed, and consider the 'general' three-way layout with white noise:

$$E(y) = X_1\beta_1 + X_2\beta_2 + X_3\beta_3, \qquad D(y) = \sigma^2 I, \tag{1}$$

where X_1 is the n by r design matrix for the first treatment factor, X_2 is the n by c design matrix for the second treatment factor, while X_3 is the n by b design matrix for blocks. Following [5], let us consider the following canonical correlations:

(a) $\rho(i.j)$ between $X_i'y$ and $X_j'y$,
(b) $\rho(i.jk)$ between $X_i'y$ and $(X_j,X_k)'y$,
(c) $\rho(i;j|k)$ between $X_i'y$ and $X_j'M_ky$,
(d) $\rho(i.j|k)$ between $X_i'M_ky$ and $X_j'M_ky$,

where $M_k = I - X_kX_k^+$ is the n by n symmetric idempotent matrix which spans (or orthogonal projector which projects onto) the null space of X_k; $i \neq j$, $i \neq k$, $j \neq k$; $i,j,k = 1,2,3$. The quantities $\phi(3.12) = 1 - \rho^2(3.12)$ are the 'canonical efficiency factors' as introduced by James and Wilkinson [2], and these are also the diagonal elements of the matrix Λ_2. Canonical correlations of the types (a), (b) and (d) were studied in detail in [5] – those of type (c) have apparently not been considered explicitly before. When there is at least one $\rho(i.j)$ equal to 1 and when there is just one $\rho(i.jk)$ equal to 1 then the 'tiers' i and j are connected; furthermore it was shown that the three-way layout is completely connected if and only if there is precisely one $\rho(i.jk)$ equal to 1 and precisely one $\rho(i.j)$ equal to 1, $i \neq j, i \neq k, j \neq k$; $i,j,k = 1,2,3$. In Tables 1-3 these unit canonical correlations have been omitted – Preece's designs (P5), (P4) and (P1) are all completely connected. Since the column nullity of the matrix (X_j,X_k) is always at least equal to 1 there will always be a canonical correlation $\rho(i.jk)$ equal to 0 – such zero canonical correlations have also been omitted in the tables that follow.

For design (P5) we will let the columns in X_1 identify the five levels of treament T1 (upper case letters), the columns in X_2 the five levels of treatment T3 (lower case letters), and the columns in X_3 the ten blocks; thus $n = 30$, $r = 5$, $c = 5$, and $b = 10$. We note that the block-T1 and block-T3 incidence matrices are identical, viz. $X_3'X_1 = X_3'X_2$, confirming – as observed above – that the two sets of treatments, T1 and T3, are confounded on the block-averaged data.

We display these non-unit canonical correlations – and the corresponding canonical efficiency factors – in Table 1. We find that for each of the sets of 'tiers' (1.3), (2.3), (1.23) and (2.13), i.e., respectively between T1 and blocks (ignoring T3), between T3 and blocks (ignoring T1), between T1 and (T3 and blocks), and between T3 and (T1 and blocks), there is just one distinct non-unit non-zero canonical correlation $\rho = 1/\sqrt{6}$, in each case with multiplicity $m = 4$; thus $\phi = 1 - \rho^2 = 5/6 = 0.833$. Since as noted above, the two sets of treatments T1 and T3 are orthogonal – after adjusting for blocks – we find that the vectors $X_1'M_3y$ and $X_2'M_3y$ are uncorrelated and so there are no non-zero canoncial correlations $\rho(1.2|3)$. It follows that the T1-T3 incidence matrix $X_1'X_2 = (1/3)NN'$, where $N = X_1'X_3 = X_2'X_3$ is the (common) treatment-blocks incidence matrix. Furthermore we find just one distinct non-unit non-zero canonical correlation: $\rho(1.2) = 1/6$, $\rho(3.12) = \sqrt{2/7}$, $\rho(1.3|2) = \sqrt{6/7}$, and $\rho(2.3|1) = \sqrt{1/7}$, each again with multiplicity 4.

The corresponding table for design (P4) of canonical correlations and canonical efficiency factors follows as Table 2, where as for Table 1 the columns in X_1 identify the five levels of treatment T1 (upper case letters) and the columns in X_3 the ten blocks; the columns in X_2 identify the five levels of treatment T2 (lower case letters) – once again $n = 30, r = 5, c = 5$ and $b = 10$. The matrices X_1 and X_3 are the same for designs (P4) and (P5).

Since the layout of upper case letters T1 to the blocks is the same in designs (P4) and (P5) it is clear that the canonical correlations and efficiency factors (1.3) must also be the same, but we find it interesting to note that they are also the same for (1.3) and (2.3), i.e., between treatments T1 and T2 (upper and lower case letters), and between treatment T2 (lower case letters) and blocks.

tiers	m	ρ	$\phi = 1 - \rho^2$	
1, 2	4	$1/6 = 0.167$	$35/36 = 0.972$	
1, 3	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	$= \lambda_{12}$
2, 3	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	
1.23	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	
2.13	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	$= \lambda_{22}$
3.12	4	$\sqrt{2/7} = 0.535$	$5/7 = 0.714$	$= \mathrm{dg}(\Lambda_2)$
	4	0	1	
1;2|3	4	0	1	
1;3|2	4	$\sqrt{5}/6 = 0.373$	$31/36 = 0.861$	
2;1|3	4	0	1	
2;3|1	4	$\sqrt{5}/6 = 0.373$	$31/36 = 0.861$	
3;1|2	4	$\sqrt{5/42} = 0.345$	$37.42 = 0.881$	
3;2|1	4	$\sqrt{5/42} = 0.345$	$37/42 = 0.881$	$= \lambda^*_{22}$
1.2|3	4	0	1	
1.3|2	4	$1/\sqrt{7} = 0.378$	$6/7 = 0.857$	
2.3|1	4	$1/\sqrt{7} = 0.378$	$6/7 = 0.857$	

Table 1
Canonical correlations ρ and canonical efficiency factors ϕ for Preece's design (P5).

tiers	m	ρ	$\phi = 1 - \rho^2$	
1, 2	4	$1/6 = 0.167$	$35/36 = 0.972$	
1, 3	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	$= \lambda_{12}$
2, 3	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	
1.23	4	$\sqrt{7/27} = 0.509$	$20/27 = 0.741$	
2.13	4	$\sqrt{7/27} = 0.509$	$20/27 = 0.741$	$= \lambda_{22}$
3.12	4	$1/\sqrt{21} = 0.218$	$20/21 = 0.952$	$= \mathrm{dg}(\Lambda_2)$
	4	$1/\sqrt{3} = 0.577$	$2/3 = 0.667$	
1;2|3	4	$\sqrt{5/54} = 0.304$	$49/54 = 0.907$	
1;3|2	4	$5/(6\sqrt{3}) = 0.481$	$83/108 = 0.769$	
2;1|3	4	$\sqrt{5/54} = 0.304$	$49/54 = 0.907$	
2;3|1	4	$5/(6\sqrt{3}) = 0.481$	$83/108 = 0.769$	
3;1|2	4	$\sqrt{3/14} = 0.463$	$11/14 = 0.786$	
3;2|1	4	$\sqrt{3/14} = 0.463$	$11/14 = 0.786$	$= \lambda^*_{22}$
1.2|3	4	$1/3 = 0.333$	$53/54 = 0.981$	
1.3|2	4	$\sqrt{5/21} = 0.488$	$16/21 = 0.762$	
2.3|1	4	$\sqrt{5/21} = 0.488$	$16/21 = 0.762$	

Table 2
Canonical correlations ρ and canonical efficiency factors ϕ for Preece's design (P4).

More interesting is that between blocks and (T1 and T2) there are now two distinct non-unit non-zero canonical correlations $1/\sqrt{3}$ and $1/\sqrt{21}$, each with multiplicity 4, while the two sets of treatments T1 and T2 – after adjusting for blocks – are not at all orthogonal, with the canonical correlation $\rho(1.2|3) = 1/3 = 0.333$.

4. Genstat Balance

Genstat ANOVA requires that if the matrix U is the part of the design matrix X associated with a treatment term then $U'C_iU = \lambda U'U$, for some positive scalar λ, if necessary after adjustment for preceding treatment terms. [If this fails, but if $U'C_iU(U'U) - U'C_iU = \lambda U'C_iU$, the U is declared balanced but only partially estimable.] The matrix U is, however, adjusted for both preceding terms and the stratum effects simultaneously, so that unless the preceding terms are orthogonal after adjustment for stratum effects, the adjusted matrix U is not the same from stratum to stratum. In matrix notation, suppose that there are two treatment terms associated with the design matrices X_1 and X_2, or

$$E(yu) = X\beta = X_1\beta_1 + X_2\beta_2, \tag{2}$$

compare (1). Genstat ANOVA checks that $X_1'C_iX_1 = \lambda_{1i}X_1'X_1$ for some positive scalar λ_{1i} and then combines the X_1 effect with the stratum effects to give the idempotent matrix

$$C_{i|1} = C_i - C_iX_1(X_1'C_iX_1)^{-1}X_1'C_i \tag{3}$$

which removes both the X_1 and stratum effects simultaneously. It then checks to see if $X_2'C_{i|1}X_2 = \lambda_{2i}X_2'X_2$ for some positive scalar λ_{2i}. The reported Genstat ANOVA efficiency factors are λ_{1i} and λ_{2i}. Since $\Sigma C_i = I$, it follows that $\Sigma\lambda_{1i} = 1$; in general, however, we have $\Sigma\lambda_{2i} \leq 1$. Moreover λ_{1i} and λ_{2i} are not in general the same as the canonical efficiency factors in Λ_i, which do add to 1 over strata since they are eigenvalues of pairwise-orthogonal matrices which sum to the identity. In fact it is not hard to prove that if the Genstat ANOVA efficiency factors do add to 1 over strata then they must coincide with the canonical efficiency factors. If $i = 2$ is the within-blocks stratum, then λ_{12} and λ_{22} can be linked to canonical correlations: in the notation of Section 3, $\lambda_{12} = \phi(1,3)$ and $\lambda_{22} = \phi(2.13)$, compare Tables 1 and 2 and Table 3.

If, instead, we adjust for X_1 by leaving C_i alone but remove the X_1 effect from X_2 to give $X_{2|1} = M_1X_2$, where $M_1 = I - X_1(X_1'X_1)^{-1}X_1'$, then we could check to see if $X_{2|1}'C_iX_{2|1} = \lambda^*_{2i}X_{2|1}'X_{2|1}$ for some positive scalar λ^*_{2i}. Genstat ANOVA does in fact already do this when X_1 is intrinsically aliased with X_2, for instance when X_2 is an interaction with X_1 and another factor. Unfortunately $X_{2|1}'C_iX_{2|1}$ is not proportional to the information matrix for B_2 and so this does not produce a valid ANOVA table in the ith stratum unless $X_{2|1}'C_iX_1 = 0$; if, however, this is so for all i then the λ^*_{2i} (and the λ_{1i}) are the canonical efficiency factors, which of course add to 1. In Tables 1-3 we note that $\lambda^*_{22} = \phi(3;2|1)$.

5. Examples

The above matrix calculations can be done quite simply with Genstat matrix functions. For design (P4) the within-blocks canonical efficiency factors are $20/21 = 0.952$ and $2/3 = 0.667$, each repeated four times, compare $\phi(3.12)$ in Table 2, and zero repeated once. The between-blocks canonical efficiency factors are of course one minus these. It is instructive to look at the corresponding eigenvectors (and canonical variates).

The first canonical efficiency factor of $20/21 = 0.952$ corresponds to contrasts in the case sums, of the type:

$$(A+a) - \tfrac{1}{4}\{(B+b)+(C+c)+(D+d)+(E+e)\}, \tag{4}$$

whereas the second efficiency factor of $2/3 = 0.667$ corresponds to contrasts in the case differences, such as:

$$(A-a) - \tfrac{1}{4}\{(B-b)+(C-c)+(D-d)+(E-e)\}, \tag{5}$$

and the zero canonical efficiency factor corresponds to the overall mean (the single unit canonical correlation identifying connectedness of blocks with the two sets of treatments). Note that the overall case difference

$$(A-a) + \tfrac{1}{4}(B-b)+(C-c)+(D-d)+(E-e), \qquad (6)$$

is not estimable (in either stratum) since every observation contains one upper case and one lower case letter – the linear combination (6) lies in the null space of the partitioned design matrix (X_1,X_2) for the two sets of treatments since the column vector of ones belongs to the intersection $\tau(X_1)\cap\tau(X_2)$.

In fact the interpretation of these eigenvectors is much easier if we regard the case of the letters as the factor CASE with two levels, upper and lower, and the letters as the factor LETTER with five levels A, B, C, D and E. The treatment model can then be written as LETTER*CASE. The LETTER main effect (4) has an efficiency factor of 20/21 = 0.952, the LETTER.CASE interaction (5) has an efficiency factor of 2/3 = 0.667, and the CASE main effect (6) is not estimable in either stratum. This breakdown of the model into orthogonal terms is called the treatment pseudo-structure by Houtman and Speed [1].

It turns out that Preece's design (P5) has the same eigenvectors and hence the same treatment pseudo-structure as design (P4) but with efficiency factors of 5/7 = 0.714 for the LETTER main effect (4), and 1 for the LETTER.CASE interaction (5). This means that the LETTER.CASE interaction is not estimable between blocks, which again explains why the upper case letters are confounded with the lower case letter – in fact the block-T1 and block-T3 incidence matrices are identical, that is $X_3'X_1 = X_3'X_2$.

6. Fitting the Treatment Pseudo-Structure Using Multiple Copies

Unfortunately such models, like the diallel cross experiment with equal male and female lines, cannot be written as a Genstat model formula. However Thompson [6] has shown in the Genstat Newsletter how such models can be fitted by making two copies of the data. For Preece's designs (P4) and (P5) one copy is made for each level of the CASE factor. The LETTER factor takes the upper case letters in the first copy and the lower case letters in the second. A PLOT factor with thirty levels is introduced for each pair of identical observations, and the BLOCK factor is nested in the PLOT factor in the blocks declaration. The resulting output for design (P5) with the same random data as before is:

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

VARIATE: Y2

SOURCE OF VARIATION              DF          SS       SS%          MS       VR

BLOCK STRATUM
  LETTER                          4   1.008E  2     23.16   2.519E  1     3.56
  RESIDUAL                        5   3.536E  1      8.13   7.072E  0
TOTAL                             9   1.361E  2     31.29   1.512E  1


BLOCK.PLOT STRATUM
  LETTER                          4   3.496E  1      8.04   8.739E  0     0.55
  LETTER.CASE                     4   7.383E  1     16.97   1.846E  1     1.16
  RESIDUAL                       12   1.901E  2     43.70   1.584E  1
TOTAL                            20   2.988E  2     68.71   1.494E  1

BLOCK.PLOT.*UNITS* STRATUM
  LETTER                          4   0.000E  0      0.00   0.000E  0
  CASE                            1   0.000E  0      0.00   0.000E  0
  LETTER.CASE                     4   0.000E  0      0.00   0.000E  0
  RESIDUAL                       21   0.000E  0      0.00   0.000E  0
TOTAL                            30   0.000E  0      0.00   0.000E  0

GRAND TOTAL                      59   4.350E  2    100.00

GRAND MEAN                          5.05
TOTAL NUMBER OF OBSERVATIONS          60

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

MODEL TERM                       ER  NON-ORTHOGONAL TERMS

BLOCK STRATUM
  LETTER                      0.167
```

```
BLOCK.PLOT STRATUM
  LETTER                        0.417  BLOCK
  LETTER.CASE                   0.417

BLOCK.PLOT.*UNITS* STRATUM
  LETTER                        0.417  BLOCK  BLOCK.PLOT
  LETTER.CASE                   0.583  BLOCK.PLOT
```

The last stratum should be ignored. For the remaining strata, the sums of squares are all twice as large as they should be but the percentage sums of squares for totals and residuals are identical to those in the previous analysis. The canonical efficiency factors can be recovered by ignoring the last stratum. For example, the canonical efficiency of the LETTER main effect is 0.417/(0.167+0.417) = 0.714. Note that the CASE main effect is not estimable in either of the first two strata and the LETTER.CASE interaction is fully estimated in the second stratum. Obviously design (P5) could be very useful for a blocked diallel cross experiment.

7. Three or More Factors

In passing, readers may be interested to know that the additve model with all three factors, T1+T2+T3, does not have a readily interpretable orthogonal treatment decomposition. But taking this a bit further, Preece [3] remarks that these factors are part of the following design for four treatment factors T1, T2, T3 and T4, reproduced below:

			Block							
1	2	3	4	5	6	7	8	9	10	
Aa*Aa*	Bb*Bb*	Cc*Cc*	Dd*Dd*	Ee*Ee*	Aa*Aa*	Bb*Bb*	Cc*Cc*	Dd*Dd*	Ee*Ee*	
Cb*De*	Dc*Ea*	Ed*Ab*	Ae*Bc*	Ba*Cd*	Bd*Ec*	Ce*Ad*	Da*Be*	Eb*Ca*	Ac*Db*	
DeCb	Ea*Dc*	Ab*Ed*	Bc*Ae*	Cd*Ba*	Ec*Bd*	Ad*Ce*	Be*Da*	Ca*Eb*	Dc*Ac*	(P6)

The additive model T1+T2+T3+T4 produces the following ANOVA table and information summary on the same random data:

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

VARIATE: Y

SOURCE OF VARIATION               DF          SS        SS%          MS        VR

BLOCK STRATUM
  T1                               4      50.376      23.16      12.594      2.94
  T2                               4      13.397       6.16       3.349      0.78
  RESIDUAL                         1       4.283       1.97       4.283
TOTAL                              9      68.056      31.29       7.562

BLOCK.*UNITS* STRATUM
  T1                               4      44.492      20.46      11.123      5.41
  T2                               4      30.321      13.94       7.580      3.69
  T3                               4      15.358       7.06       3.839      1.86
  T4                               4      51.038      23.47      12.760      6.21
  RESIDUAL                         4       8.215       3.78       2.054
TOTAL                             20     149.424      68.71       7.471

GRAND TOTAL                       29     217.480     100.00

GRAND MEAN                              5.05

TOTAL NUMBER OF OBSERVATIONS              30

***** 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
  T3                          0.729  BLOCK  T2
  T4                          0.595  BLOCK  T1  T2  T3
```

Obviously the treatments are not orthogonal, but can we use the same trick of taking four copies of the data to get an orthogonal treatment decomposition? Surprisingly, the answer is yes, provided that we create a CASE factor at two levels, upper and lower, and a TYPEFACE factor at two levels, roman and italic. Then the treatment model LETTER*(CASE/TYPEFACE) gives the following ANOVA table and information summary on four copies of the same data:

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

VARIATE: Y4

SOURCE OF VARIATION              DF          SS       SS%          MS       VR

BLOCK STRATUM
  LETTER                          4   1.050E  2     12.07   2.626E  1     1.53
  LETTER.CASE                     4   1.501E  2     17.25   3.752E  1     2.19
  RESIDUAL                        1   1.713E  1      1.97   1.713E  1
TOTAL                             9   2.722E  2     31.29   3.025E  1

BLOCK.PLOT STRATUM
  LETTER                          4   7.738E  1      8.89   1.934E  1     2.35
  LETTER.CASE                     4   5.733E  1      6.59   1.433E  1     1.74
  LETTER.CASE.TYPEFACE            8   4.301E  2     49.44   5.377E  1     6.54
  RESIDUAL                        4   3.286E  1      3.78   8.215E  0
TOTAL                            20   5.977E  2     68.71   2.988E  1

BLOCK.PLOT.*UNITS* STRATUM
  LETTER                          4   0.000E  0      0.00   0.000E  0
  CASE                            1   0.000E  0      0.00   0.000E  0
  LETTER.CASE                     4   0.000E  0      0.00   0.000E  0
  CASE.TYPEFACE                   2   0.000E  0      0.00   0.000E  0
  LETTER.CASE.TYPEFACE            8   0.000E  0      0.00   0.000E  0
  RESIDUAL                       71   0.000E  0      0.00   0.000E  0
TOTAL                            90   0.000E  0      0.00   0.000E  0

GRAND TOTAL                     119   8.699E  2    100.00

GRAND MEAN                             5.05

TOTAL NUMBER OF OBSERVATIONS            120

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

MODEL TERM                        ER  NON-ORTHOGONAL TERMS

BLOCK STRATUM
  LETTER                      0.028
  LETTER.CASE                 0.139

BLOCK.PLOT STRATUM
  LETTER                      0.347  BLOCK
  LETTER.CASE                 0.069  BLOCK
  LETTER.CASE.TYPEFACE        0.208

BLOCK.PLOT.*UNITS* STRATUM
  LETTER                      0.625  BLOCK  BLOCK.PLOT
  LETTER.CASE                 0.792  BLOCK  BLOCK.PLOT
  LETTER.CASE.TYPEFACE        0.792  BLOCK.PLOT
```

Once again the last stratum should be ignored and all sums of squares should be divided by four. The canonical efficiency factors are 0.347/(0.028+0.347) = 0.926 = 25/27 for the LETTER main effect and 0.069/(0.139+0.069) = 0.333 = 1/3 for the LETTER.CASE interaction. The LETTER.CASE.TYPEFACE interaction is fully estimated within blocks. Perhaps this design could be useful for a blocked 'tetrallel' cross experiment!

8. Another Design with Two Factors

The other designs given by Preece [3] do not have such easily interpretable eigenvectors. His first design (1), reproduced below:

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

(P1)

is not even accepted as balanced by Genstat ANOVA, although it must be generally balanced since it has only two strata. The efficiency factors within blocks are 0.954 and 0.680, each repeated four times, 4/5 = 0.800 once, and 0 once. The 0.800 efficiency factor corresponds to the contrast

$$f - (a+b+c+d+e), \tag{7}$$

but the others are harder to interpret.

The canonical correlations ρ and canonical efficiency factors ϕ for design (P1) are given in Table 3. We note that the two efficiency factors within blocks $\phi(3.12)$ with multiplicity 4 are the only efficiency factors in Tables 1-3 that are not rational numbers, being the roots of a quadratic equation with discriminant equal to 5481.

It is interesting to note that the addition of a pseudo-factor PF of the form (7) to the treatment model formula does in fact make the design acceptable to Genstat ANOVA as balanced. The treatment model `UPPER+LOWER//PF` gives the results below. Once again the efficiency factors do not all add to 1 over strata, which means that these are not the canonical efficiency factors.

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

SOURCE OF VARIATION                DF

BLOCK STRATUM
  UPPER                             4
  LOWER                             5
TOTAL                               9

BLOCK.*UNITS* STRATUM
  UPPER                             4
  LOWER                             5
  RESIDUAL                         11
TOTAL                              20

GRAND TOTAL                        29


***** INFORMATION SUMMARY *****
MODEL TERM                         EF  NON-ORTHOGONAL TERMS

BLOCK STRATUM
  UPPER                         0.167
  PF                            0.200
  LOWER                         0.089  UPPER

BLOCK.*UNITS* STRATUM
  UPPER                         0.833  BLOCK
  PF                            0.800  BLOCK
  LOWER                         0.778  BLOCK  UPPER
```

Finally, it should be stressed that the orthogonal treatment decomposition deduced from general balance may not be meaningful in practice. The only advantages are independent inferences about treatment effects and their simple combinability over strata; this is obviously irrelevant if these effects are not of primary interest.

tiers	m	ρ	$\phi = 1 - \rho^2$	
1, 2	4	0	1	
1, 3	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	$= \lambda_{12}$
2, 3	5	$1/\sqrt{5} = 0.447$	$4/5 = 0.800$	
1.23	4	$\sqrt{41}/(6\sqrt{6}) = 0.436$	$175/216 = 0.810$	
2.13	4	$\sqrt{2}/3 = 0.471$	$7/9 = 0.778$	$= \lambda_{22}$
	1	$1/\sqrt{5} = 0.447$	$4/5 = 0.800$	
3.12	4	0.21502034	0.953766252	$= \mathrm{dg}(\Lambda_2)$
	4	0.56606795	0.679567081	
	1	$1/\sqrt{5} = 0.447$	$4/5 = 0.800$	
1;2|3	4	$\sqrt{5}/(6\sqrt{6}) = 0.153$	$211/216 = 0.977$	
1;3|2	4	$\sqrt{41}/(6\sqrt{6}) = 0.436$	$175/216 = 0.810$	
2;1|3	4	$1/(3\sqrt{5}) = 0.149$	$44/45 = 0.978$	
2;3|1	4	$\sqrt{2}/3 = 0.471$	$7/9 = 0.788$	
	4			
	1	$1/\sqrt{5} = 0.447$	$4/5 = 0.800$	
3;1|2	4	$1/\sqrt{6} = 0.408$	$5/6 = 0.833$	
3;2|1	5	$1/\sqrt{5} = 0.447$	$4/5 = 0.800$	$= \lambda_{22}^{*}$
1.2|3	4	$1/6 = 0.167$	$35/36 = 0.972$	
1.3|2	4	$\sqrt{41}/(6\sqrt{6}) = 0.436$	$175/216 = 0.810$	
2.3|1	4	$\sqrt{2}/3 = 0.471$	$7/9 = 0.778$	
	1	$1/\sqrt{5} = 0.447$	$4/5 = 0.800$	

Table 3
Canonical correlations ρ and canonical efficiency factors ϕ for Preece's design (P1).

9. Acknowledgements

We are pleased to acknowledge the helpful comments received from R.A. Bailey and D.A. Preece concerning this paper. Our research was supported in part by the Natural Sciences and Engineering Research Council of Canada and by the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche du Gouvernement du Québec.

10. References

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

[2] James, A.T. and Wilkinson, G.N.
Factorization of the Residual Operator and Canonical Decomposition of Nonorthogonal Factors in Analysis Variance.
Biometrika, 58, pp. 279-294, 1971.

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

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

[5] Styan, G.P.H.
Canonical Correlations in the Three-way Layout.
Proc. Pacific Statistical Congress (I.S. Francis, B.F.J. Manly and F.C. Lam, Eds.).
Elsevier Science, Amsterdam, pp. 433-438, 1986.

[6] Thompson, R.
Diallel Crosses, Partially Balanced Incomplete Block Designs with Triangular Association Schemes and Rectangular Lattices.
Genstat Newsletter, 10, pp. 16-32, 1983.

Some Comparisons Using Genstat on PCs

E R Williams
CSIRO
Division of Water Resources
Canberra Labs.
GPO Box 1666
ACT 2601

1. Introduction

It is very much a time of change with the use of the statistical package Genstat on IBM compatible personal computers (PCs). Since 1986 there have been four versions of Genstat released for PCs. In 1986 there was Genstat version 4.03e (G4.03e), then in 1988 Genstat 5 Release 1.2 (G5.1.2). In the last year there have been two versions of Genstat 5 Release 1.3, the first for use with 80386 or 80486 based PCs (G5.1.3(1)) and more recently a version for use with 8086 or 80286 based PCs (G5.1.3(2)).

The existence of several versions of Genstat for PCs is due to a combination of the switch from Genstat 4 to Genstat 5 and also the very rapid development of PC technology. In the last few years we have gone from the 8086 (XT) micro-processor to the 80286 (AT) and more recently the 80386SX, 80386 and 80486 micro-processors. Table 1 presents some details of the four versions of Genstat and summarizes the micro-processors with which they are compatible. To further highlight the changes that are happening, it will very soon be appropriate to add the 80486 micro-processor and Genstat 5 Release 2 to Table 1. However at present the combinations in Table 1 cover the majority of current users of Genstat on PCs and so the purpose of this note is to carry out some time comparisons of the micro-processors and Genstat versions in the table. Also included in this note are some time comparisons of G5.1.3(1) with and without the 80387 mathematics co-processor and some Genstat comparisons of PCs with a Sun 3/60 workstation.

Genstat Version	Origin	Release Date	Micro-processor compatibility				Co-processor Required
			8086	80286	80386SX	80386	
G4.03e	C.E.M.S France	1986	yes	yes	yes	yes	no
G5.1.2	Rothamsted England	1988	yes	yes	yes	yes	yes
G5.1.3(1)	Lancaster England	1989	no	no	yes	yes	no
G5.1.3(2)	Rothamsted England	1990	yes	yes	yes	yes	yes

Table 1
Some details on Genstat versions available for PCs.

2. Materials and Methods

In order to provide a range of PCs to test the four versions of Genstat, three PCs were selected for detailed comparisons. These were an NEC APC IV Powermate 1, a Samsung SD 700 and a Toshiba 5200; in addition some extra comparisons were made using an NEC APC IV Powermate 1+. Details of specifications are provided in Table 2.

For the time comparisons a number of examples were chosen. For the main comparisons, three examples were taken from the Genstat examples set which is distributed with each version of Genstat. For this note these three examples have been called AVCCOX, LINEAR and NONLINER; Table 3 gives the connection between the names used in the example sets of various Genstat versions. The G4.03e examples are Genstat 4 versions of the other Genstat 5 examples, although REGRESS.IN seems slightly different, having many more comments in the file. Also for G4.03e there were no examples corresponding to NONLINER. For the comparison of the performance of G5.1.3(1) with and without the mathematics co-processor, two examples were

PC	Abbreviation	Hard Disk	RAM	Clock Speed	Micro-processor	Co-processor
NEC APC IV Powermate 1	NECP1	20mb	640k	8mhz	80286	yes
NEC APC IV Powermate 1+	NECP1+	42mb	640k	12mhz	80286	yes
Samsung SD700	SAMSUNG	40mb	2mb	16mhz	80386SX	yes
Toshiba 5200	TOSHIBA	100mb	2mb	20mhz	80386	yes

Table 2
PCs used for the Genstat comparisons.

taken from the Genstat examples set provided with G5.1.3(1), namely NONLINER.DAT and GRAPHX.DAT (in this note called NONLINER and GRAPHX respectively). The final example used in this note has been made available by Dr J.T. Wood and has been used in an earlier unpublished comparison of G5.1.2 and Genstat version 5.1.2 on the Sun 3/60 workstation. Hence this example (called JTWREG) allows a direct comparison for a regression model across quite a range of different computers.

Genstat Version	Example		
	AVCCOX	LINEAR	NONLINER
G4.03e	AVCCOX.IN	REGRES.IN	no standard example
G5.1.2	AVCCOX.GEN	LINEAR.GEN	NONLINER.GEN
G5.1.3(1)	AVCCOX.DAT	LINEAR.DAT	NONLINER.DAT
G5.1.3(1)	AVCCOX.GEN	LINEAR.GEN	NONLINER.GEN

Table 3
Names of Genstat examples in various Genstat distributions.

For most of the examples and combinations of computers and Genstat versions used in the time comparisons, two times have been recorded. For the PCs there is the screen time, which corresponds to the time taken for the job to run with output going directly to the screen; then there is the file time which is the time taken for the job to run with output going to a file. Thus a job generating a lot of output and using a fast processor could show a big difference between screen and file times. For the Sun 3/60 the screen time and CPU time were recorded. The UNIX operating system on the Sun 3/60 is a multi-user system, and so it is probably unreasonable to compare screen times with the PCs; however, CPU time should be able to be compared with file times on the PCs. For the PCs the times were obtained using a stop watch; for the Sun 3/60 the computer clock was used.

3. Results

The time taken to run the Genstat examples AVCCOX, LINEAR and NONLINER for various combinations of Genstat versions and PCs is given in Table 4. Times are in minutes and seconds, the first time is the screen time and the second time (in parentheses) is file time. For G5.1.2 and G5.1.3(2) screen times were so large that it was assumed that outputting to the screen was not a limiting factor and so file times would be expected to be about the same as screen times. Using Genstat version 5.1.3 on the Sun 3/60 workstation, the corresponding times were 1.50 (0.39), 0.32 (0.09) and 1.08 (0.41) for screen and CPU times for AVCCOX, LINEAR and NONLINER respectively.

(a) AVCCOX

Genstat Version	PC		
	NECP1	SAMSUNG	TOSHIBA
G4.03e	11.45 (9.25)	4.09 (3.37)	2.55 (2.28)
G5.1.2	95.36	22.56	12.54
G5.1.3(1)	not compatible	1.20 (0.53)	0.54 (0.28)
G5.1.3(2)	112.37	64.13	47.24

(b) LINEAR

Genstat Version	PC		
	NECP1	SAMSUNG	TOSHIBA
G4.03e	2.49 (2.18)	0.55 (0.49)	0.38 (0.32)
G5.1.2	11.19	2.59	2.01
G5.1.3(1)	not compatible	0.25 (0.19)	0.19 (0.13)
G5.1.3(2)	14.53	7.43	5.03

(c) NONLINER

Genstat Version	PC		
	NECP1	SAMSUNG	TOSHIBA
G4.03e	no standard example	no standard example	no standard example
G5.1.2	36.45	13.12	8.46
G5.1.3(1)	not compatible	1.31 (1.17)	0.50 (0.44)
G5.1.3(2)	51.31	24.15	15.22

Table 4
Screen times and file times (in parentheses) needed to run Genstat examples AVCCOX, LINEAR and NONLINER.

The results in Table 5 provide a comparison of screen times with and without a mathematics co-processor using G5.1.3(1). Finally, Table 6 gives screen and file times to run the example JTWREG. Included in the table are times on the Sun 3/60 workstation for Genstat versions 5.1.2 and 5.1.3 as well as the time taken to run an equivalent job using the statistical package GLIM version 3.77.

(a) NONLINER

PC	Co-processor	
	yes	no
SAMSUNG	1.31	6.40
TOSHIBA	0.50	3.22

(b) GRAPHX

PC	Co-processor	
	yes	no
SAMSUNG	1.28	7.19
TOSHIBA	0.50	3.38

Table 5
Screen times needed to run Genstat examples NONLINER and GRAPHX using G5.1.3(1) with and without mathematics co-processor.

(a) PCs

Genstat Version	NECP1	NECP1+	SAMSUNG	TOSHIBA
G4.03e	1.15 (1.12)	0.44 (0.42)	0.23 (0.22)	0.16 (0.15)
G5.1.2	11.44	6.47	3.10	2.27
G5.1.3(1)	not compatible	not compatible	0.39 (0.38)	0.21 (0.20)
G5.1.3(2)	12.49	7.12	4.13	2.57

(b) Sun 3/60

Package	Sun 3/60
Genstat version 5.1.2	1.28 (0.43)
Genstat version 5.1.3	0.32 (0.24)
GLIM	1.03 (0.30)

Table 6
Screen times and file times (in parentheses) needed to run example JTWREG on PCs and Sun 3/60.

4. Discussion

The results in Table 4 show an enormous difference in the performance of various Genstat versions. Looking say at AVCCOX the time taken with the most recent version of Genstat (G5.1.3(2)) is over 47 minutes on the TOSHIBA compared with just 28 seconds using G5.1.3(1). On the NECP1 for the same example we have a time of about nine minutes for G4.03e compared with 112 minutes for G5.1.3(2).

Table 4 also allows a comparison of PCs for the same version of Genstat. So looking for example at the performance of the 80386SX and 80386 micro-processors for the 386 version of Genstat (G5.1.3(1)) we see that the time ratio is about 60% for the comparison of the two processors. Table 5 focuses on G5.1.3(1) which is clearly the best performing version of Genstat and shows that despite the fact that G5.1.3(1) does not require a mathematics co-processor to run, performance is vastly enhanced with the presence of the co-processor.

Table 6 supports the findings in Table 4 and in addition shows that there has been an improvement in speed between Genstat versions 5.1.2 and 5.1.3 on the Sun 3/60 workstation, although this may reflect some changes in the operating environment for the multi-user system. The time for G5.1.3(1) on the Toshiba is comparable to the CPU time for Genstat 5.1.3 and GLIM on the Sun 3/60. Table 7 highlights this by collecting together some figures already given for file times and CPU times on the TOSHIBA and Sun 3/60 workstation respectively.

Example	Machine	
	TOSHIBA	Sun 3/60
AVCCOX	0.28	0.39
LINEAR	0.13	0.09
NONLINER	0.44	0.41
JTWREG	0.20	0.24

Table 7
File times for the TOSHIBA and CPU times for the Sun 3/60 workstation for examples AVCCOX, LINEAR, NONLINER and JTWREG.

Version G5.1.3(2) of Genstat is in fact an update of G5.1.2 and the poor performance of both is mainly due to the large amount of disk swapping that is required as a result of difficulties in fitting the program into the computer random access memory. The memory requirements for these versions are so large (e.g. 581k for G5.1.2) as to necessitate the removal of most memory resident programs in order to run Genstat and this detracts from the comfort of the operating environment. On the other hand both G4.03e and G5.1.3(1) can be run from within other packages such as DATACHAIN which is a front-end for Genstat 4, Genstat 5 and SAS.

5. Conclusions

There are many other comparisons which can be extracted from the tables, but we leave these to the reader. We are of course aware that the comparisons are specific to the examples chosen and the machines selected to run the versions of Genstat. We have tried to span a range of examples by selecting one with a lot of output (AVCCOX) and one with a lot of computation and very little output (NONLINER); the screen and file times reflect this. Equally we have tried to include a range of PCs with different micro-processors and clock speeds. Hence we believe that some general conclusions can be drawn despite the specific nature of our exercise:

(a) The 386 version of Genstat (G5.1.3(1)) performs very well and is comparable to the Sun 3/60 version of Genstat 5.1.3. However of course G5.1.3(1) does not run on 8086(XT) and 80286(AT) PCs.

(b) The first release of Genstat on PCs, namely G4.03e performs much better than either G5.1.2 or G5.1.3(2) and should still possibly be preferred for XT and AT PCs, despite the fact that Genstat 4 has been superseded by Genstat 5.

Versions of Genstat 5 for Personal Computers

P G N Digby
Statistics Department
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Herts AL5 2JQ

This note is intended to accompany the preceding article from Emlyn Williams about the various virtues of different versions of Genstat for Personal Computers. Also I would like to describe what is happening with regards to versions of Release 2 of Genstat 5 for PCs.

Our original intention, with Release 1.2 of Genstat 5 for PCs, was to provide a version of Genstat 5 that could be used by people with limited computing availability. The advantages of Genstat 5 over previous versions of Genstat are well-known, and need not be repeated here. However, there were several short-comings of that (first) PC version of Genstat 5:

(a) there were no facilities for high-resolution graphics;
(b) because the executable program was contained in a single file (of about 3.3 Mb) it was difficult for NAG to distribute reliably to different types of (supposedly) IBM-compatible PCs;
(c) there was a severe penalty to the user of having the definition of all of the Genstat commands held within the user's data-space for the duration of a job;
(d) the executable program needed a lot of RAM space to run, so that, for example, it could not be used on some networked PCs.

With Release 1.3 we addressed all of these problems. The executable program for Release 1.3 is split into separate files for each part of the overlay structure: none of these is too large to fit onto a 360Kb diskette, which overcomes NAG's distributional problem.

At the time that Release 1.3 for PCs was being implemented we were also developing the 'directive-cache' system that is used in Release 2. This works by holding the definitions of all Genstat's directives in a separate direct-access file, rather than reading them into Genstat's internal storage space at the beginning of each run. As statements are read, an internal 'cache' is checked to see if it contains the definition of the relevant directive: if not, the directive-definition is read into the 'cache' from the direct-access file.

A pre-release version of this directive-cache system was introduced into Release 1.3 for PCs, with space for a single directive in the internal cache. This saved considerably on the amount of internal space used by Genstat for its system information, and thus gave us the opportunity to address the problem of the RAM space used, with the result that three different versions of Release 1.3 are supplied to run in 506Kb, 539Kb, or 566Kb. However, an additional consequence of the directive-cache used in Release 1.3 is the need to be more careful over the use of the internal space with regard to system information. So that this does not overflow we needed to introduce some extra garbage-collection (the internal equivalent of using the DELETE directive) after each statement.

All of these changes provided more opportunity for the space used by Release 1.3 on a PC hard disk to become fragmented – in particular the direct-access file used to hold the directive definitions and the area used to hold the directory information for the overlay files – and thus for Release 1.3 to be slower than Release 1.2.

However, our own checking of a pre-release version suggested that run-times were comparable, and in some instances improved. The version that we used for checking was prior to the introduction of the high-resolution graphics and the directive caching. At that stage the executable program was held in about 120 files; the subsequent inclusion of high-resolution graphics increased that considerably. This means that the directory holding these files now needs three rather than two areas of disk space, which implies that the overlay-loader needs to search over a larger area of disk in order to find the location of the required overlay file, before it can even read the file into memory. Because of user pressure to provide the improved facilities of Release 1.3 over Release 1.2 we perhaps did not do as much bench-marking as we might have liked.

For Release 2 we have investigated the effect on performance of holding the executable program in one file (as for Release 1.2), many files (i.e. one per overlay region, as for Release 1.3), or several files each containing one or more overlay regions. Our results suggest very strongly that there is an initial improvement in performance when several files are used, but that this is negated when many files are used: the critical aspect seems to be the size of the directory holding the files. Therefore, for Release 2 we are combining various parts of the overlay structure into single files. This will reduce the size of the directory used to hold the overlay files, so that the directory itself will not become fragmented, provided that it is used to hold only the overlay files and Genstat's support files. This means that, as different parts of the overlay are required, Genstat's overlay-loader need search only a single area of the disk to find the location of the relevant file. Of course, this will not help much if the overlay file is itself fragmented, or if the directory area is remote from the overlay files. We intend that the installation system for Release 2 be improved to make it more likely for a contiguous area of disk-space to be used for the directory holding the overlay files, and for the support files that are accessed during a run of Genstat. However, this can happen only if the disk space used by existing files is compacted prior to installing Genstat.

In Release 2 the size of the directive-cache can be varied: its size and its initial contents are specified at the stage when Genstat does its initial bootstrap. The size of the cache will affect the speed of Genstat – run-times will decrease as the cache-size is increased – but more internal system-information space is used by a larger cache. At the time of writing we have not fully investigated the effect of the size of the directive-cache on the performance of Release 2 for PCs.

We have done some preliminary bench-marking of a development version of Release 2 against Releases 1.3 and 1.2. Obviously the performance of any program will vary from PC to PC; however, our timings on an 80286-based PC for three of the supplied examples are given in the following table, from which it can be seen that the performance of Release 2 is now again comparable with Release 1.2.

Example	Genstat Version Times in minutes:seconds		
	1.2	**1.3**	**2.1**
AVCCOX	36:17	65:32	36:41
BOXJEN	14:00	18:06	13:22
MLTVAR	9:35	16:44	12:47

Users of Release 1.3 for 80386-based PCs may well wonder what all this discussion of timings, performance and so on is really about. That version of Genstat is extremely fast; however, it does depend on a far more powerful CPU chip than the 8086-chip used in the original PCs. I am pleased to be able to say that (at the time of writing) Release 2 is also being implemented for 80386-based PCs.

I am very grateful to several colleagues for their help, guidance and advice – in particular Simon Harding, Steve Haywood, and Roger Payne.

Combining Tables with Variates – An Alternative Approach

H R Simpson
20 Northall Road
Eaton Bray
Dunstable
Bedfordshire *LU6 2DQ*

Although it is true that the CALCULATE directive does not yet allow qualification of tables (Hamilton, [1]), it is not true that it is impossible to qualify table identifiers anywhere in a Genstat program.

Though the distinction between compile-time and execution-time or run-time (which was necessary in earlier versions of Genstat) is almost irrelevant in Genstat 5, statements still have to be compiled before they can be executed. The compiler, in fact, will accept qualified tables; but if the coded information is passed to other parts of the program a fault will be reported.

However, some things can be done without troubling those other parts. The compiler can never write to subsets of structures (defined either by qualification or restriction) but it can pick up the values defined by a qualification. It can be asked to do so by using the substitution symbol # (note that # is not necessary if the subset is a single value). In particular, options that expect a single real value may be set to an element of any numerical structure, and instead of

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

(where Bmean is a table classified by one factor) – which will fail –

```
VARIATE V; VALUES=!(#Bmean$[#Block])
```

can be used, which may be even simpler than SAS.

This holds for tables classified by more than one factor. (Remember, of course, that there is a limit on the dimensionality of Genstat tables: nine.) There is no need to resort to a procedure, as the following output shows.

```
 1  FACTOR [LEVELS=3] Block; VALUES=!(1,3,2,3,1,2)
 2  FACTOR [LEVELS=2] Rep; VALUES=!(3(1,2))
 3  TABLE [CLASSIFICATION=Rep,Block] RBmean; \
 4     VALUES=!(12.5,15.5,16.0,22.5,25.5,26.0)
 5  VARIATE Z; VALUES=!(RBmean$[#Rep; #Block])
 6  PRINT Rep,Block,Z; FIELDWIDTH=8

   Rep   Block       Z
     1       1   12.50
     1       3   16.00
     1       2   15.50
     2       3   26.00
     2       1   22.50
     2       2   25.50

 7  "and if you are prepared to be sufficiently devious you may
-8   even use a qualified table in a CALCULATE statement:"
 9  TABLE [CLASSIFICATION=Rep,Block] XYZ; VALUES=!(2,1,3,4,6,5)
10  VARIATE V; VALUES=!(12.5,16.0,15.5,13.7,15.2,14.3)
11  & [NVALUES=2] SS
12  CALCULATE SS = !(V$[XYZ$[1,2; 2,1]])
13  PRINT [ORIENTATION=across] SS

          SS       12.50       13.70
```

Note that the two qualifier lists in XYZ$[1,2; 2,1] are processed in parallel and the term is treated as XYZ$[1; 2],XYZ$[2; 1]. A different effect is obtained if the lists are bundled: XYZ$[!(1,2); !(2,1)] (or, equivalently, XYZ$[J; K] where J and K are structures with appropriate values) defines a sub-table, the intersection of the first and second levels of the first factor with the second and first levels of the second factor. The implications for CALCULATE are fairly horrendous, since an expression might involve an arbitrary number of such sub-tables.

In Release 1 these features were not completely tested and some uses of qualified tables may fail inexplicitly. Release 2 should do much better, and the only failing of which I am aware is that if an invalid qualification for a table is found – e.g. more qualifier lists than dimensions or a position value greater than the number of levels of the relevant factor – then the diagnostic message may print the wrong structure name. This has been cured for Release 3.

There is a similar fault, also cured for Release 3, in the diagnostic message that will appear if too many '`[`' are found before matching '`]`' begin to appear. You will be told that the limit is nine. This should be 31, and

```
PRINT !e(A$[B$[C$[D$[E$[F$[G$[H$[I$[J$[K$[L$[M$[N$[O$[P$[Q$[R$[S$[T$[U$[     \
   V$[W$[X$[Y$[Z$[Z$[Y$[X$[W$[V$[U]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]])
```

is acceptable, if only just.

Since writing the above, two more failings in the Release 2 implementation of qualified structures have surfaced:

1. If SM is a symmetric matrix, `CALCULATE V=SM$[...]` works; but `CALCULATE V=X$[SM$[...]]` does not.
2. If F is a factor, `F$[3]` should return the numerical level corresponding to the third value, but it does not do so in all circumstances and its use should be avoided.

References

[1] Dr R. Sackville Hamilton.
Combining Tables with Variates.
Genstat Newsletter, 25, pp. 57-58, 1990.

Explicit and Implicit Loops

P W Lane
Statistics Department
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Herts AL5 2JQ

The FOR directive provides a powerful mechanism for repetitive work. In fact, the ability to specify a series of dummies to be substituted at each pass of the loop makes it more convenient and flexible than otherwise equivalent looping mechanisms in other computing languages. For example, to draw a series of graphs of the variables *ya,yb,yc,yd* against the variables *xa,xb,xc,xd*, the following loop is easy to construct:

```
FOR y=ya,yb,yc,yd; x=xa,xb,xc,xd
   GRAPH y; x
ENDFOR
```

The apparent shortness of a FOR loop, however, can belie the amount of work that is actually being performed. In particular, it must be remembered that in Genstat 5, each statement is compiled and executed at the same time; therefore, each statement in a loop must be compiled and executed the number of times that the loop is repeated. On a slow computer, such as an 80286-based PC, this can result in a surprisingly long wait for a loop to complete.

Another example where a loop is clearly needed is in the formation of the matrix power of a square matrix. This needs to be done in the study of transition processes, for example, where a transition matrix is powered up to find the state of the process after a given number of steps. Genstat does not provide a special function for taking powers of matrices. At first sight, the obvious way to carry out this powering operation in Genstat would be to use an explicit loop:

```
MATRIX [ROWS=7; COLUMNS=7] Matrix; VALUES=!(...)
& Result; VALUES=Matrix
FOR [NTIMES=59]
   CALCULATE Result = Result*+Matrix
ENDFOR
```

Here, the matrix called Matrix, with seven rows and seven columns, is taken to the power 60, using the NTIMES option of the FOR directive to specify the number of passes through the loop.

The work can be speeded up by replacing the explicit loop with an implicit one. Many directives in Genstat provide implicit looping by the use of lists, which greatly reduces the amount of time spent in compiling the commands.

```
MATRIX [ROWS=7; COLUMNS=7] Matrix; VALUES=!(...)
& Result; VALUES=Matrix
CALCULATE 59(Result) = Result*+Matrix
```

The CALCULATE statement could be written

```
CALCULATE 59(Result) = 59(Result)*+59(Matrix)
```

to make it clear that there are three parallel lists each of 59 identifiers. However, the standard rule in Genstat is that the first list in the parameters of a statement defines the number of operations, and other lists in the parameters are recycled until the first list is exhausted. CALCULATE will carry out each operation in turn, so that the structure Result used for the second calculation will be the result of the first calculation, and so on.

The formation of a matrix power can actually be achieved with much less work than this. On a slow computer, it might well be worth reducing the number of matrix operations; also, it is possible that rounding error may be a problem if the matrix has many rows and the power is large. In that case, the fewer operations that are done the better. The following statements find the 60th power using only eight matrix multiplications, compared to the 59 used previously, with only a small overhead in storage of temporary matrices.

```
MATRIX [ROWS=7; COLUMNS=7] Matrix; VALUES=!(...)
& M[1]; VALUES=Matrix
CALCULATE M[2,4,8,16,32] = M[1,2,4,8,16]*+M[1,2,4,8,16]
& Result = M[32]*+M[16]*+M[8]*+M[4]
```

The following short procedure carries out the powering operation for any given power. Though the explicit loop has to be reinstated to cope with the generality, it is repeated only `LOG2(POWER)` times; for example, with `POWER=60`, it is repeated only five times.

```
PROCEDURE 'MPOWER'
PARAMETER 'MATRIX',     "Input matrix: must be square" \
          'POWER',      "Input scalar: must be positive integer" \
          'RESULT'      "Output matrix"
   CALC work = MATRIX
   SCALAR start,index; VALUE=1,POWER
   FOR [NTIMES=POWER]
      IF INTEGER(index = index/2) < index
         IF start
            CALC RESULT = work
            SCALAR start; VALUE=0
         ELSE
            CALC RESULT = RESULT*+work
         ENDIF
         EXIT index==0.5
         CALC index = INTEGER(index)
      ENDIF
      CALC work = work*+work
   ENDFOR
ENDPROCEDURE
```

[Note: it is also possible to compute the power of a symmetric matrix using the eigenvalues calculated by FLRV].

An expanded version of this procedure has been accepted for Procedure Library 2[2].

This article was written while the author was a visiting fellow in the Statistics Department of the Australian National University, Canberra.

Editing Data Structures

P W Lane
Statistics Department
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
Harpenden
Herts AL5 2JQ

The EDIT directive in Genstat is designed to edit text structures only. In order to change values of other structures, you have to use directives such as CALCULATE or EQUATE, or else PRINT the values into a text, edit the text, and READ back into the structure. The new Menu System also provides some facilities for editing, but only of variates.

Any computing system on which Genstat is available also provides one or more editors. These may be distributed with the operating system, such as the EDT editor with Vax/VMS or the vi editor with Unix, or they may have been purchased to provide an alternative editing capability. In any case, a user of Genstat is likely to have a favourite editor on the system being used, and this is almost certainly a screen editor, which will usually be much more convenient to use than the line-editing style of the EDIT directive in Genstat. Genstat does not attempt to provide such facilities because explicit links to screen editors would lead to serious machine-dependency.

However, the SUSPEND directive in Genstat is provided specifically to allow communication between Genstat and the operating system. It can be used in particular to call an editor while Genstat is suspended. Therefore, the ability to edit data structures can be provided by a simple procedure, allowing you to make use of your favourite editor rather than having to learn another set of editing conventions. Of course, the procedure is bound to be machine-dependent because of the difference in editor's names and commands to delete files, so it may not be suitable to include such a procedure in the Genstat Procedure Library. But any site, or any user, can define a local procedure library containing a procedure for editing data that is tailored to local conditions.

I have written such a procedure specifically for Release 2.1 of Genstat in the VAX/VMS environment. In its current form, it can cope with all data structures except language structures (such as pointers) and compound structures. It is about 100 lines long, but most of this consists of special action to deal with potential problems in reading general strings into text and factor structures, and allowing a set of vectors of equal length to be edited together. To make it quicker for a reader to implement an editing facility at any site, a simplified procedure is listed here. It omits the awkward cases of texts and factors, but can deal with any single numerical structure, and it includes a loop to check for valid reinput of the values.

To use the procedure once it is stored in a library attached to Genstat, just type

```
EDATA identifier
```

to edit the values of a data structure. Genstat will be suspended, and the editor will be invoked to let you modify the values: the values are printed into a file, and you can control the format of this printing with the FIELDWIDTH and DECIMALS parameters of EDATA. When you have finished making changes, exit from the editor as usual, and the procedure will attempt to read the new values back into the data structure. If it fails, it will ask you if you want to try again – either returning you to the editor, or abandoning depending on your reply.

If you have more than one editor at your fingertips, you can choose between them by using the option EDITOR. For example, to use an editor called textedit rather than the default, give the command

```
EDATA [EDITOR='textedit'] identifier
```

Procedure EDATA

```
PROCEDURE 'EDATA'

" Procedure to call external editor to edit data. "
OPTION 'EDITOR','DELETE','FILE','DELAPPEND'; \
  MODE=p; DEFAULT='ed','del','edata.tmp',';*'; \
  NVALUES=1; DECLARED=yes; TYPE=4('text'); PRESENT=yes
PARAMETER 'DATA','FIELDWIDTH','DECIMALS'; MODE=p; \
  DEFAULT=*,!(*),!(*); SET=yes,no,no; DECLARED=yes; PRESENT=yes; \
  TYPE=!t(scalar,variate,matrix,diagonal,symmetric,table), \
  2('scalar','variate'); NVALUES=*,1,1

" Print current values into file, without labelling beyond identifier. "
SCALAR [VALUE=*] outc,inc
OPEN FILE; CHANNEL=outc; FILETYPE=output
PRINT [CHANNEL=outc; SQUASH=yes] \
  '*** Edit data values, but retain the first three lines of this file ***'
PRINT [CHANNEL=outc; RLPRINT=*; CLPRINT=*] DATA; \
  FIELDWIDTH=#FIELDWIDTH; DECIMALS=#DECIMALS
CLOSE outc; FILETYPE=output

CONCATENATE [edcom] EDITOR,' ',FILE
& [delcom] DELETE,' ',FILE,DELAPPEND
DUMMY reply; VALUE=2

FOR [NTIMES=999]

  " Give edit command to operating system. "
  SUSPEND [edcom]

  " Retrieve values, ignoring messages, blank line and identifiers."
  OPEN FILE; CHANNEL=inc; FILETYPE=input
  SKIP [CHANNEL=inc; FILETYPE=input] 3
  " Check for error in reading."
  DISPLAY [CHANNEL=null]
  SET [DIAGNOSTIC=*]
  READ [CHANNEL=inc] DATA
  SET [DIAGNOSTIC=w,f]
  GET [FAULT=dcheck]
  CLOSE inc; FILETYPE=input

  IF dcheck
    " Display fault and repeat."
    PRINT 'Warning from procedure EDATA: Data cannot be read back in.'
    QUESTION [PREAMBLE='Do you want to try again?'; RESPONSE=reply; \
      MODE=t; DEFAULT='y'] 'y','n'; CHOICE='yes','no'
    EXIT [EXPLANATION='Procedure EDATA has abandoned editing'] reply==2
  ELSE
    " Exit if data correctly read."
    EXIT
  ENDIF
ENDFOR

" Delete temporary files used for editing."
SUSPEND [delcom]

ENDPROCEDURE
```

Modifications for Release 1.3

The procedure makes use of several features introduced in Release 2.1, so cannot be implemented without modifications for use with Release 1.3. The modifications required are listed here.

(1) The OPTION and PARAMETER statements cannot use the new error-checking and default-setting parameters. Change them as follows, including IF blocks to assign defaults to the auxiliary parameters, and check the type of structure to be edited. Other checks can be included if desired, to avoid producing diagnostics within the procedure, such as when a structure is supplied with no values already defined.

```
OPTION 'EDITOR','DELETE','FILE','DELAPPEND'; \
  MODE=p; DEFAULT='ed','del','edata.tmp',';*'
PARAMETER 'DATA','FIELDWIDTH','DECIMALS'; MODE=p
  IF UNSET(FIELDWIDTH)
    DUMMY FIELDWIDTH; VALUE=!(*)
  ENDIF
  IF UNSET(DECIMALS)
    DUMMY DECIMALS; VALUE=!(*)
  ENDIF
  GETATT [ATT=type] DATA; SAVE=patt
  IF patt[1] .NI. !(1,4,5,6,7,8)
    PRINT 'Procedure EDATA cannot edit this type of structure'
    EXIT [CONTROL=proc]
  ENDIF
```

(2) Explicit channels should be used, rather than using the new feature to ask for the next available channel by supplying a missing value to the CHANNEL parameter of OPEN.

```
SCALAR [VALUE=4] outc,inc
```

(3) The CLPRINT option of PRINT suppresses printing of identifiers as well as other column labels in Release 1.3. Thus, the SKIP statement needs to be changed to:

```
SKIP [CHANNEL=inc; FILETYPE=input] 2
```

(4) The DISPLAY statement cannot be used in Release 1.3 with the option setting CHANNEL=identifier. Replace it by

```
DISPLAY
```

This will unfortunately mean that if a diagnostic has already occured in the job, and DISPLAY has not been used, then when the procedure is called the diagnostic will be displayed again. This could be avoided by opening another file and directing the output from DISPLAY into it by use of the OUTPUT directive.

(5) The QUESTION statement should be replaced with a PRINT statement followed by a READ statement to receive the reply.

```
PRINT 'Do you want to try again?  Type 1 for yes, 2 for no:'
READ [END=*] reply
```

(6) The EXIT directive in Release 1.3 does not have an option EXPLANATION, so replace it by:

```
IF reply==2
  PRINT 'Procedure EDATA has abandoned editing'
  EXIT
ENDIF
```

Modifications for Other Operating Systems

The procedure is designed to be easily modified for other operating systems. All that should need changing are the default settings for the options EDITOR, DELETE, FILE, DELAPPEND. For example, on a Unix system the list of defaults could look like

```
DEFAULT='vi','rm -f','edata.tmp',' '
```

Of course, the procedure cannot be used if the SUSPEND directive is not available in the implementation of Genstat.

This article was written while the author was a visiting fellow in the Statistics Department of the Australian National University, Canberra.

Fitting Non-linear Models and Estimating Functions of Model Parameters

M Patefield
Department of Applied Statistics
University of Reading
Whiteknights
PO Box 217
Reading RG6 2AN
United Kingdom

1. Introduction

Standard curves, such as the line plus exponential, are fitted adequately by the FITCURVE directive. Non-standard curves may be fitted by the FITNONLINEAR directive, which although usually producing adequate estimates of the model parameters, frequently gives a computationally inaccurate estimate of their variance-covariance matrix and consequently of their standard errors. A procedure FITIMPROVE gives improvements in these features.

Explicit functions of parameters may be estimated adequately using the RFUNCTION directive. However, particularly when the parameters have an ill-conditioned variance-covariance matrix, RFUNCTION can give an inaccurate estimate of the variance-covariance matrix of the functions. A procedure IFUNCTION estimates both implicit and explicit functions of parameters and calculates their variance-covariance matrix with greater precision.

Both of the procedures FITIMPROVE and IFUNCTION require the user to supply expressions for derivatives. Copies of these procedures, together with documentation and examples are available by E-mail from user SNSPATED @ UK.AC.RDG.AM.CMS. The Genstat program producing the results in Tables 2 and 3 of this article is also available.

2. Non-linear Models

Consider the generalized non-linear model $E(y_i) = \mu_i(\theta)$, $i = 1,2,...,n$, involving unknown parameters θ. Estimates $\hat{\theta}$ are obtained by minimizing the deviance

$$D = \Sigma w_i d(y_i,\mu_i)$$

where w_i is the weight attached to the ith observation. The form of the function $d(y_i,\mu_i)$, the deviance per observation, is given in [1] for the distributions Normal, Poisson, Binomial, Gamma and Inverse Gaussian. The (Fisher) information matrix I_θ (equal to the expectation of the Hessian matrix) has elements

$$I_{jk}^{(\theta)} = E\left(\frac{\partial^2(\frac{1}{2}D)}{\partial\theta_j\partial\theta_k}\right) = \frac{1}{2}\Sigma w_i E\left(\frac{\partial^2 d}{\partial\mu_i^2}\right)\frac{\partial\mu_i}{\partial\theta_j}\frac{\partial\mu_i}{\partial\theta_k}$$

where the expectation on the right-hand side is a simple function of μ_i for the standard distributions.

The procedure FITIMPROVE requires expressions to evaluate the fitted values (μ_i) of the model and their derivatives ($\partial\mu_i/\partial\theta_j$) with respect to the parameters θ. It produces improved estimates $\hat{\theta}$ (compared with those produced by FITNONLINEAR) and evaluates the information I_θ at $\hat{\theta}$. The asymptotic variance-covariance matrix of the parameter estimates is itself estimated by

$$V_\theta = I_\theta^{-1}(s^2)$$

where s^2 is the dispersion. The method of evaluation of I_θ^{-1} is controlled by the NOTRANSFORM option of FITIMPROVE. For NOTRANSFORM = yes I_θ^{-1} is evaluated using the INVERSE function for square matrices which is more accurate than the algorithm used by Genstat for inversion of symmetric matrices. If I_θ is ill-conditioned (near singularity) small errors in the calculation of its elements I_{jk}^{θ} may produce larger errors in its inverse I_θ^{-1}. An alternative

method of calculating I_θ^{-1} is invoked by the default NOTRANSFORM = no. This method is based on an orthogonal transformation to the parameters ϕ satisfying $\theta = \theta_0 + U\phi$ where U is chosen to make $I_\phi = U^T I_\theta U$ well-conditioned and θ_0 is arbitrary. The inverse matrix I_θ^{-1} is calculated using $I_\theta^{-1} = UI_\phi^{-1}U^T$. However, direct calculation of I_ϕ will not increase accuracy over direct inversion of I_θ. Increased accuracy is achieved by calculating the elements of I_ϕ as

$$I_{jk}^{(\phi)} = E\left(\frac{\partial^2(\frac{1}{2}D)}{\partial\phi_j\partial\phi_k}\right) = \frac{1}{2}\Sigma w_i E\left(\frac{\partial^2 d}{\partial\mu_i^2}\right)\frac{\partial\mu_i}{\partial\phi_j}\frac{\partial\mu_i}{\partial\phi_k} \qquad (1)$$

with

$$\frac{\partial\mu_i}{\partial\phi_j} = \sum_l U_{lj}\frac{\partial\mu_i}{\partial\theta_l}.$$

The matrix I_ϕ with elements given by (1) is then inverted and I_θ^{-1} obtained as $I_\theta^{-1} = UI_\phi^{-1}U^T$. A convenient choice of U to make I_ϕ well-conditioned is the matrix of latent vectors of I_θ. Algebraically I_ϕ will then be diagonal, but when calculated by (1) will only be approximately so due to the increase in accuracy.

	Applied nitrogen (tonnes/ha per annum)				
Block	0	0.1	0.2	0.4	0.8
a	5.951	9.0845	10.864	12.095	11.026
b	4.8875	7.084	10.33	13.60185	14.365
c	6.898	9.697	11.618	13.0966	12.266

Table 1
Dry matter yields (1985) from plots cut at 4-weekly intervals (tonnes/ha)

2.1. Example

Gains in precision achieved by the procedure FITIMPROVE are illustrated by fitting the model

$$E(y) = a + br^x + cx^p, \qquad (2)$$

with p known and normally distributed errors, to the data of Table 1, where y is the yield and x is the nitrogen fertiliser application. The data is an extract from a randomized block experiment described more fully elsewhere [2,3]. Two values of p are considered: $p = 1$ and $p = 1.125$. The ill-conditioning of the information matrix I_θ is apparent from the condition numbers (ratio of largest to smallest eigenvalues) of the correlation matrix which are 153802 and 290727 for $p = 1$ and $p = 1.125$ respectively. Computationally accurate parameter estimates and their standard errors are given in Table 2. Standard errors are based on a dispersion of $s^2 = 1.407$, the residual mean square from the randomised block analysis of variance. The accuracy of the estimates $\hat{\theta}$ and their standard errors are given in Table 3 for four fitting techniques. The assessments of accuracy are made by comparison with accurate values produced by Fortran programming in quadruple precision arithmetic. The figures quoted in Table 3 are the maximum percentage errors over the four parameters $\theta = (r,b,c,a)$.

	$p = 1$		$p = 1.125$	
$\hat{r}$	0.117	(0.414)	0.140	(0.707)
$\hat{b}$	–22.7	(49.5)	–22.2	(67.8)
$\hat{c}$	–14.9	(36.9)	–14.1	(45.7)
$\hat{a}$	28.6	(49.7)	28.1	(68.1)

Table 2
Parameter estimates and their standard errors (in brackets)

Fitting technique	$p = 1$ Maximum % error in		$p = 1.125$ Maximum % error in	
	$\hat{\theta}$	s.e.($\hat{\theta}$)	$\hat{\theta}$	s.e.($\hat{\theta}$)
(a) FITCURVE	0.012	0.016	–	–
(b) FITNONLINEAR	0.010	5.75	0.071	14.46
(c) FITIMPROVE with NOTRANSFORM = yes	0.00039	0.047	0.00026	0.052
(d) FITIMPROVE with NOTRANSFORM = no	0.00039	0.00048	0.00026	0.00048

Table 3
Maximum Percentage Error in Estimating θ and in the Standard Error of

The four fitting techniques used are:

(a) Using the FITCURVE directive. For $p = 1$, model (2) reduces to the standard line plus exponential curve. All estimates and standard errors are reasonably accurate. For $p = 1.125$ the model cannot be fitted by FITCURVE.

(b) Using the FITNONLINEAR directive. Although satisfactory parameter estimates are obtained for both $p = 1$ and $p = 1.125$ there are substantial errors in calculating their standard errors.

(c) Using the procedure FITIMPROVE with NOTRANSFORM = yes. The parameter estimates are more accurate than using either (a) or (b). Their standard errors are reasonably accurate.

(d) Using the procedure FITIMPROVE with NOTRANSFORM = no (the default). All parameter estimates and standard errors are more accurate than using either (a) or (b).

FITNONLINEAR obtains numerical estimates of the derivatives of the fitted values (the $\partial\mu_i/\partial\theta_j$) and hence standard errors of parameter estimates may be inaccurate. For instance, with $p = 1.125$ the computationally accurate value for the standard error of $\hat{r}$ is 0.707, but FITNONLINEAR produces a value of 0.604.

FITIMPROVE not only increases the accuracy of parameter estimation, but also improves the accuracy of the inverse matrix I_θ^{-1}, the variance-covariance matrix $s^2 I_\theta^{-1}$, and the standard errors and correlations of the parameter estimates. The advantage of using NOTRANSFORM = no is particularly apparent for this example where the information matrix I_θ is ill-conditioned. In the example, the estimates are calculated using the Gauss-Newton method. They may be calculated by the Newton-Raphson method which uses the observed rather than the expected (Fisher) information matrix by setting the option SEMETHOD to Newton-Raphson. In this case additional expressions are required for calculation of the second derivatives of the fitted values with respect to the parameters.

3. Estimating Functions of Model Parameters

Explicit functions $f(\theta)$ of parameters θ are estimated by $f(\hat{\theta})$ and their variance-covariance matrix is estimated by

$$V_f = \frac{\partial f}{\partial \theta} V_\theta \left(\frac{\partial f}{\partial \theta}\right)^T \qquad (3)$$

where V_θ is the variance-covariance matrix of $\hat{\theta}$ resulting from previously fitting a model using FIT, FITCURVE, FITNONLINEAR or the procedure FITIMPROVE.

When some (or all) of the functions are available only as solutions of implicit equations $z(\theta f(\theta)) = 0$, then they may be estimated by the procedure IFUNCTION which solves such non-linear equations iteratively for $f(\hat{\theta})$ and computes $\partial f/\partial\theta$ as the solution of

$$\frac{\partial z}{\partial \theta} + \frac{\partial z}{\partial f}\frac{\partial f}{\partial \theta} = 0.$$

To calculate V_f accurately, the variance-covariance matrix V_θ needs to be calculated accurately. However, when V_θ is ill-conditioned this does not guarantee accuracy in V_f. One way of achieving an accurate V_f is first to fit the model in terms of the locally orthogonal parameters ϕ satisfying $\theta = \theta_0 + U\phi$ obtaining estimates $\hat{\phi}$ with variance-covariance matrix V_ϕ. IFUNCTION is then used to estimate $f(\theta) = f(\theta_0+U\phi)$ by $f(\theta_0+U\hat{\phi})$ and the variance covariance-matrix of the functions f is estimated by

$$V_f = \frac{\partial f}{\partial \phi} V_\phi \left(\frac{\partial f}{\partial \phi}\right)^T. \qquad (4)$$

As

$$\frac{\partial f}{\partial \phi} = \frac{\partial f}{\partial \theta} U$$

then

$$V_f = \left(\frac{\partial f}{\partial \theta} U\right) V_\phi \left(\frac{\partial f}{\partial \theta} U\right)^T,$$

which, utilising $V_\theta = UV_\phi U^T$, is algebraically equivalent to (3). However, for appropriate choice of U such as the matrix of latent vectors of I_θ, fitting the model in terms of ϕ and using (4) rather than fitting in terms of θ and using (3) will reduce computational errors as V_ϕ will be a well-conditioned matrix.

3.1. Example

Two functions N_0 (the fertiliser application to achieve maximum yield and Y_0 (the corresponding yield) of the parameters $\theta = (r,b,c,a)$ are estimated. For $p = 1$, N_0 and Y_0 are given explicitly by

$$N_0 = \log(-c/(b \log r))/\log r$$

$$Y_0 = a + c(N_0 - 1/\log r)$$

but for general values of p, N_0 is obtained by solving

$$z = br^{N_0}\log r + cpN_0^{p-1} = 0$$

and Y_0 evaluated as

$$Y_0 = a + br^{N_0} + cN_0^p.$$

Computationally accurate values of the estimates of N_0 and Y_0 together with their standard errors are given in Table 4. N_0 and Y_0 are also estimated by eight separate techniques. For each technique the maximum percentage error over N_0 and Y_0 of the estimates of (N_0,Y_0) and of their standard errors are given in Table 5.

	$p = 1$	$p = 1.125$
$\hat{N}_0$	0.551 (0.0575)	0.554 (0.0592)
$\hat{Y}_0$	13.38 (0.879)	13.39 (0.973)

Table 4
Estimates of Functions N_0, Y_0 and their standard errors (in brackets)

Estimating Technique	$p = 1$ Maximum % error in		$p = 1.125$ Maximum % error in	
	$\hat{N}_0, \hat{Y}_0$	s.e.$(\hat{N}_0, \hat{Y}_0)$	$\hat{N}_0, \hat{Y}_0$	s.e.$(\hat{N}_0, \hat{Y}_0)$
(e) Fit using (a), then RFUNCTION	0.00020	38.05	–	–
(f) Fit using (a), then IFUNCTION	0.00019	0.015	–	–
(g) Fit using (b), then RFUNCTION	0.00014	20.61	–	–
(h) Fit using (b), then IFUNCTION	0.00014	8.05	0.00098	21.98
(i) Fit using (c), then IFUNCTION	0.000011	0.038	0.000022	0.067
(j) Fit using (d), then IFUNCTION	0.000011	0.013	0.000022	0.0099
(k) FITNONLINEAR to estimate ϕ, then IFUNCTION	0.00045	2.70	0.0012	7.19
(l) FITIMPROVE to estimate ϕ, then IFUNCTION	0.000011	0.000078	0.000011	0.000012

Table 5
Maximum Percentage Error in Estimating Functions N_0, Y_0 of parameters and in their Standard Errors

The eight techniques are:

(e) Using FITCURVE to estimate θ and then using RFUNCTION to estimate N_0, Y_0. This may only be used for $p = 1$ and produces highly inaccurate standard errors of $\hat{N}_0$, $\hat{Y}_0$.

(f) Using FITCURVE to estimate θ and then using IFUNCTION to estimate N_0, Y_0. This may only be used for $p = 1$ and produces reasonable estimates and standard errors.

(g) Using FITNONLINEAR to estimate θ and then using RFUNCTION to estimate N_0, Y_0. This may only be used for $p = 1$ as N_0 is not available as an explicit function of $\theta = (r,b,c,a)$ for $p = 1.125$. Highly inaccurate standard errors are produced by this technique.

(h) Using FITNONLINEAR to estimate θ and then using IFUNCTION to estimate N_0, Y_0. Inaccurate standard errors result from this technique.

(i) Using FITIMPROVE with NOTRANSFORM = yes to estimate θ and then using IFUNCTION to estimate N_0, Y_0. The standard errors produced by this technique are reasonably accurate but are less accurate than (f) for $p = 1$.

(j) Using FITIMPROVE with NOTRANSFORM = no to estimate θ and then using IFUNCTION to estimate N_0, Y_0. The standard errors produced are reasonably accurate (similar to (f)).

(k) Using FITNONLINEAR to estimate ϕ and then using IFUNCTION to estimate N_0, Y_0. The standard errors are inaccurate.

(l) Using FITIMPROVE to estimate ϕ and then using IFUNCTION to estimate N_0, Y_0. The standard errors of the estimates $\hat{N}_0$, $\hat{Y}_0$ are much more accurate than those produced by techniques (e) to (k).

Inaccurate calculation of V_θ by FITNONLINEAR is responsible for the inaccurate standard errors produced by (g), (h) and (k). RFUNCTION estimates $\partial f / \partial \theta$ numerically and inaccuracies in this process are responsible for the inaccurate standard errors produced by (e) and (g).

Techniques (f), (h), (i) and (j) fit model (2) in terms of the parameters θ. The variance-covariance matrix V_θ is saved for use by IFUNCTION in calculating the variance-covariance matrix V_f of the functions N_0, Y_0 using (3). Techniques (k) and (*l*) fit the model in terms of the locally orthogonal parameters ϕ given by $\theta = \theta_0 + U\phi$ with θ_0 given by the estimates of θ obtained with FITNONLINEAR (technique (b)). The transformation matrix U used is the matrix of latent vectors of the information matrix obtained using fitting technique (c). Almost identical results for (k) and (*l*) are obtained if the matrix of latent vectors of the inverse matrix obtained using technique (b) is used.

Technique (j) is accurate enough for most practical purposes. Technique (*l*) is more accurate but requires additional expressions to fit the model in terms of the locally orthogonal parameters ϕ. Also the extent of the differences resulting from use of techniques (j) and (*l*) is due to the extent of the ill-contitioning of I_θ. Technique (j) will be as good as technique (*l*) when I_θ has off-diagonal elements near to zero.

4. References

[1] McCullagh, P. and Nelder, J.A.
Generalized Linear Models. (2nd Edition).
Chapman and Hall, New York, 1989.

[2] Patefield, W.M.
Estimating response rates from non-linear curves.
Submitted to Biometrics.

[3] Tallowin, J.R.B., Kirkham, F.W., Brookman, S.K.E. and Patefield, W.M.
Response of an old pasture to applied nitrogen under steady state continuous grazing management.
Journal of Agricultural Science, Cambridge, 115, pp. 179-194, 1990.