GENSTAT NEWSLETTER
NO. 3

JANUARY 1977

As many of you know the Genstat Reference Manual and User's Guides are being revised and combined into one manual. The new manual will be released at the same time as Genstat 4.01. Many of the additional facilities that will be available in Genstat 4.01 are available in Genstat 3.09 also, and details of these are given at the end of this newsletter. The old reference manual and user's guides are not being amended except for the few brief error corrections appended.

It is a pleasure to record that this newsletter is not so introverted as the last, containing, as it does, contributions from several organisations outside Rothamsted.

RANDOM SAMPLING OF A DATA MATRIX

It is sometimes necessary, e.g. in planning a survey, to take a random sample of fixed size from a data vector or matrix. This involves sampling without replacement if the units of the sample are all to be distinct. One way of accomplishing this in Genstat is to set up a variate to number the units consecutively, and then reorder this variate randomly, using a parallel variate of random numbers generated by the function RANDU [6.2.1]. Then, if a sample of size S is required, the values 1 to S in the reordered numbering variate can be used to pick out the required sample, by using 'RESTRICT' [6.7.3].

Suppose that V is the identifier of a data vector, or set of such vectors. Then, if N and S are respectively the number of units in V, and the size of the desired sample (where S<N), the following statements will produce the sample.

```
'VARIATE'       RECNO $ N = 1 ... N
:               RAN $ N
'INTEGER'       PICK $ S = 1 ... S
'FACTOR'        REC $ N, N
'CALC'          RAN = RANDU (SEED)
:               RECNO = ORDER (RECNO;RAN)
'GROUP'         REC = INTPT (RECNO)
'RESTRICT'      V $ REC = PICK
```

The sample can then be subjected to any operation governed by restriction, e.g. print, calculate, etc.

NOTES

1. If N and S are not known, they must be read or calculated in a previous block of instructions.

2. If the data matrix is large and space is a problem, as will often be the case where a sample is required, then RAN and RECNO can be devalued before bringing V into core, so long as REC has been formed. N would have to be known in advance in this case (or alternatively use could be made of the workfile).

3. The choice of the value SEED will depend on context. See description of RANDU (6.2.1).

Brian G. Miller
Gordon Bell
Forestry Commission,
Midlothian

DIRECTORY FULL!

The diagnostic SP - 1 appears when the limit on the number of named or unnamed identifiers has been exceeded. These limits are derived from the settings of the options NID and NUNN of the REFERENCE statement.

The state of the directory can be monitored by using 'DUMP/MAIN=Y' to display the current contents of a labelled common /MAINAC/, which contains, inter alia, four variables MAXID, MAXUNN, NID and NUNN. The last two must not be confused with the options of REFERENCE with the same name.

MAXID, the maximum number of named structures,is simply the setting of the option NID, by default 100. NID is the current number of named structures

and $0 \leq NID \leq MAXID$.

Unnamed structures are numbered in the range $MAXID \leq NUNN \leq MAXUNN$ where MAXUNN = MAXID + 50 + 5*K/4 and K is the setting of the option NUNN. (50 + 5*K/4) is used instead of K to make some allowance for the unnamed structures created internally.

SP - 1 will appear therefore if either NID > MAXID or NUNN > MAXUNN. The named structures can be controlled by the user, but the unnamed structures cannot. However, the 'DEVALUE' command, if used at the beginning of a new block of instructions, will remove from the directory all unnamed structures set up by previous blocks of instructions which are no longer required.

From 3.09 the contents of common /MAINAC/ will always be output when a job fails whether DUMP = Y is specified in the REFERENCE statement or not.

Howard R. Simpson
R. E. S.

PLOTTING POINTS RELATIVE TO CANONICAL AXES

It is probably well known that if the sequence:-

```
'UNITS'      N $ n
'SCALARS'    G = g: P = p: T: C = c
'READ'       V (1...C)
'FACT'       F1 $ G, N = number list of formal values
'DSSP'       WS $ V(1...C); F1
'SSP'        WS
'MATRIX'     L $ P, 2: CMNS $ G, 2
'DIAGMAT'    R $ 2
'CVA'        WSSP = WS; RESULTS = L, R, T; SCORES = CMNS
```

is followed by

```
'VARI' X, Y $ G
'EQUATE' X, Y = CMNS $ (1, 1X) g, 1X
'FACTOR' F $ G, G = 1...g
'GRAPH/EQXY=Y, NRF = 61' Y; X $; F
```

then a graph of the group means relative to the first two canonical axes is produced. Each point is identified by a formal level of factor F and the graph is scaled such that circular confidence intervals may be drawn round each point.

However, I always like to see the scatter of all the data relative to the canonical axes, as this gives one an idea of the accuracy of the confidence intervals and also provides a visual check for outliers.

This is obtained by post-multiplying the n × p data matrix <u>Z</u> by the canonical loading matrix <u>L</u> and adjusting each set of canonical scores to have a mean of zero. Hence given <u>Z</u> we simply add the sequence below.

```
'VARI' X1, Y1 $N
'MATR' CSCRS $ N, 2
'CALC' CSCRS = PDT(Z; L)
'EQUA' X1, Y1 = CSCRS $ (1, 1X) n, 1X
'CALC' X1, Y1 = (X1, Y1) - MEAN(X1, Y1)
'GRAPH/EQXY = Y, NRF = 61' Y1; X1$; F1
```

If the graph is likely to be crowded and there are less than 10 levels of F1 the sequence 'NAME' N1 = 1,2,3,4,5,6,7 'FACTOR' F1 $ N1 = number list of formal values say, would ensure that only one digit is plotted per point. (Otherwise points appear as 01, 02 etc.)

N.B. Small letters in GENSTAT statements represent actual values.

AVOIDING EMPTY CELLS IN ANOVA OUTPUT

If a RESTRICT directive on a variate removes all the values of a particular level (or levels) of a factor the output from an ANOVA directive will contain a number of empty cells. One simple way to avoid this is to set up a different factor for each restricted ANOVA that is the same length as the variate, but only contains the levels used.

To take a very simple example suppose we had taken 5 measurements on each of 4 pigs but we wish to analyse pigs 1 and 2 and then pigs 3 and 4, we could set up factors PIG12 and PIG34 as shown below

```
PIGS   11111  22222  33333  44444
PIG12  11111  22222  22222  22222
PIG34  11111  11111  11111  22222  (formal levels)
```

and the sequence

```
'INTE' C$2 = 1,2 : D$2 = 3,4
'FOR'  J=PIG12, PIG34; K = C,D
'TREAT' J
'RESTR' DATA $ PIGS = K
'ANOVA' DATA
'REPE'
```

would give two analyses of variance with no empty cells in the tables of means.

With much longer variates, or where the levels of the factor are in a random order the LIMITS function of the 'GROUPS' directive is very useful.

For example suppose there are now 20 pigs then the sequence

```
'CALC' INDEX = FLOAT(PIGS)
'VARI' VLIM $ 9 = 1.5, 2.5, 3.5, 4.5, 5.5, 6.5, 7.5, 8.5, 9.5
'GROUPS' PIG1 = LIMITS(INDEX; VLIM)
'CALC' VLIM = VLIM + 10
'GROUPS' PIG2 = LIMITS(INDEX; VLIM)
```

would give values to the declared PIG1 and PIG2 such that separate analyses of pigs 1 - 10 and pigs 11 - 20 would have no empty cells.

CORRECTING FOR ROUNDING ERROR WHEN FORMING GENERALISED INVERSES OF SYMMETRIC MATRICES

A generalised inverse is the inverse of a singular matrix. The INV function of the CALCULATE directive will compute the inverse of a positive semi-definite symmetric matrix, i.e. a symmetric matrix with no negative latent roots. Unfortunately rounding errors in the computing of the matrix sometimes causes small discrepancies leading to latent roots that are very small but negative. The inversion directive will then fail.

The solution is to use LRV to obtain the latent roots and vectors of the symmetric matrix, M in the diagonal matrix L and the square matrix X respectively. The negative elements in L are then set to zero and a positive semi-definite matrix $\underline{M} = \underline{X}\ \underline{L} * \underline{X}'$ (L* = corrected diagonal matrix) is calculated. This will appear to be exactly the same as the original matrix the only difference is that GENSTAT will now recognise it to be positive semi-definite, and INV will work. But having $\underline{L}*$ and $\underline{X}$ it is easier to calculate the generalised inverse M^{-1} directly

$$\text{as } \underline{M}^{-1} = \underline{X}\ \underline{L}^{*-}\ \underline{X}'$$

where $\underline{L}^{*-}$ is the diagonal matrix obtained by inverting the non-zero (diagonal) elements of $\underline{L}*$.

The sequence of GENSTAT directives might then be - given a symmetric N×N matrix $\underline{M}$

```
'MATR' X $N,N
'SYMM' INVM $ N
'DIAG' L $N
'SCAL' SC
'LRV' M;X, L, SC
'CALC' L = L*(L.GE.0.0)
'CALC' M = PDT(X; PDTT(L;X))
'PRINT' M $ 12.5
'CALC/ZDZ = ZERO' L = (L/L)/L
'CALC' INVM = PDT(X;PDTT(L;X))
'PRINT' INVM $ 12.5
```

Hal McFie
M. R. I.

A GENSTAT PROGRAM FOR THE ITERATIVE SCALING ALGORITHM

Introduction

Iterative scaling or the iterative proportional fitting procedure has a distinguished history going back to 1940 when it was first proposed by Denning and Stephen. Details of the algorithm and most of the relevant references can be found in either Bishop et al (1975) or Haberman (1974). It is currently used by workers dealing with multiway tables either

(A) to adjust a table so that its margins conform to specified margins while preserving interactions

or

(B) to compute the maximum likelihood estimates of the fitted values under suitable log-linear models.

The facilities incorporated in the Genstat language, in particular, its table structures and operations, allow a short and comprehensible coding of the alogorithm. Haberman (1972) wrote a Fortran subroutine to implement the algorithm. Though very useful, it is not easy to understand, as Fortran only has a limited range of data structures. GLIM provides a more general solution to problem B, but it places severe restrictions on the size of the table and in its current version cannot solve problem A.

The program

The algorithm adjusts the internal cells of a multiway table (here named INTERNAL) to conform to specified margins of a second table (EXTERNAL). The specified margins are stored as tables and may be of any dimension. They are called EXT(1), EXT(2)......and INT(1), INT(2)..... Each specified margin of INTERNAL is successively scaled so that it equals the corresponding margin of EXTERNAL. This is repeated until convergence sets in. The criterion used here is based on an analogue of a log-likelihood ratio statistic, GSQ. Crudely speaking, this procedure corresponds to adjusting a two way table by first taking percentages of the rows, then of the columns, then of the rows, then of the columns ... etc.

A cautionary note: we had wished to code the algorithm as a general purpose MACRO but unfortuantely it is not possible to dynamically declare tables dimensioned by variable numbers of factors in current versions of Genstat.* We thus chose to present a simple coding of the algorithm and this means the program does not do any safety checks. In particular the program may well fail if either INTERNAL or EXTERNAL have zero or negative entries.

Example 1

This data set comes from Jeffreys (1960) and gives the breakdown of 388 Buckinghamshire Social Welfare workers by age, sex and marital status. It illustrates problem A.

	Age	<35	35-44	45+
Status				
Single	Male	9 (7.71)	3 (2.08)	3 (1.25)
	Female	55 (14.30)	60 (12.65)	95 (11.99)
Married	Male	11 (7.56)	27 (15.05)	49 (16.35)
	Female	18 (3.76)	21 (3.55)	37 (3.75)
Total				388 (100)

The iteratively scaled breakdown is given in brackets.

The original breakdown is difficult to comprehend because of the disparities in the margins, there being more female than male workers, more single than married workers and more elder than younger workers. The breakdown is thus scaled so that there are equal numbers in each of the three one-way margins: Age (33.33, 33.33, 33.33), Sex (50, 50) and status (50, 50).

If there were no interaction between these factors then all entries in the scaled table should be near 100/12 = 8.33. Thus we note that
(i) only the young single males and young married males occur in the 'right' proportions.

(ii) there are too few married females in all age groups.

(iii) there are too many old married men.
and finally
(iv) there is little difference between the 35-44 and 45+ categories and these two levels can be combined without loss of information.

```
'REFERENCE'  ITERATIVESCALING
      ''N IS THE DIMENSION OF THE TABLE
        M IS THE NUMBER OF SPECIFIED MARGINS
        INTERNAL IS ADJUSTED TO CONFORM TO SPECIFIED MARGINS''
      ''DECLARE TABLES''
'SCALAR'  N=3  :  M=3  :  GSQ  :  L
'FACTOR'  X(1)$2  :  X(2)$2  :  X(3)$3
'TABLE'   EXTERNAL, INTERNAL, HOLD $ X(1...3)
'VALUE'   INTERNAL=9,3,3,55,60,95,11,27,49,18,21,37
'CALC'    EXTERNAL=100/12
'CALC'    INTERNAL=INTERNAL*SUM(EXTERNAL)/SUM(INTERNAL)
'CALC'    HOLD=INTERNAL
      ''SPECIFY MARGINS AND COMPUTE''
```

* This is now possible. See later item under Section 2 of new facilities for Genstat 3.09

```
'TABLE'    EXT(1), INT(1) $ X(1)
   :       EXT(2), INT(2) $ X(2)
   :       EXT(3), INT(3) $ X(3)
'CALC'     EXT(1...M)=EXTERNAL
     ''SCALE''
'LABEL'    L
'FOR'      Y=EXT(1...M)  ;  Z=INT(1...M)
'CALC'     Z=INTERNAL
   :       INTERNAL=INTERNAL*Y/Z
'REPEAT'
'CALC'     GSQ=2*SUM(HOLD*LOG(HOLD/INTERNAL))
   :       HOLD=INTERNAL
'PRINT'    GSQ $ 12.4
'JUMP'     L*(GSQ.GT.0.0001)
'PRINT'    INTERNAL, EXTERNAL $ 12.2
'RUN'
```

```
GSQ       33.8579
GSQ        1.9596
GSQ        0.3219
GSQ        0.0356
GSQ        0.0040
GSQ        0.0004
GSQ        0.0001

                                 INTERNAL
                X(3)        1        2       3
X(1)            X(2)
   1               1     7.71     2.08    1.25
                   2    14.30    12.65   11.99
   2               1     7.56    15.05   16.35
                   2     3.76     3.55    3.75

                                 EXTERNAL
                X(3)        1        2       3
X(1)            X(2)
   1               1     8.33     8.33    8.33
                   2     8.33     8.33    8.33
   2               1     8.33     8.33    8.33
                   2     8.33     8.33    8.33
```

Example 2

This data set is the famous Bartlett plum root data, first used to illustrate a test of no three way interaction in a 2^3 table. The reader will find the original data and the same fitted values given on page 89 of Bishop et al (1975), as output from EXTERNAL and INTERNAL.

```
'REFERENCE'   ITERATIVESCALING
      ''N IS THE DIMENSION OF THE TABLE
        M IS THE NUMBER OF SPECIFIED MARGINS
        INTERNAL IS ADJUSTED TO CONFORM TO SPECIFIED MARGINS''
     ''DECLARE TABLES''
'SCALAR'   N=3  :  M=3  :  GSQ  :  L
'FACTOR'   X(1)$2  :  X(2)$2  :  X(3)$2
'TABLE'    EXTERNAL, INTERNAL, HOLD $ X(1...3)
'CALC'     INTERNAL=10
'VALUE'    EXTERNAL=156,84,84,156,107,133,31,209
```

```
'CALC'     INTERNAL=INTERNAL*SUM(EXTERNAL)/SUM(INTERNAL)
 :         HOLD=INTERNAL
  ''SPECIFY MARGINS AND COMPUTE''
'TABLE'    EXT(1), INT(1) $ X(1), X(2)
 :         EXT(2), INT(2) $ X(2), X(3)
 :         EXT(3), INT(3) $ X(3), X(1)
'CALC'     EXT(1...M)=EXTERNAL
  ''SCALE''
'LABEL'    L
'FOR'      Y=EXT(1...M) ; Z=INT(1...M)
'CALC'     Z=INTERNAL
 :         INTERNAL=INTERNAL*Y/Z
'REPEAT'
'CALC'     GSQ=2*SUM(HOLD*LOG(HOLD/INTERNAL))
 :         HOLD=INTERNAL
'PRINT'    GSQ $ 12.4
'JUMP'     L*(GSQ.GT.0.0001)
'PRINT'    INTERNAL, EXTERNAL $ 12.2
'RUN'
```

```
GSQ          207.5871
GSQ            3.8679
GSQ            0.0805
GSQ            0.0029
GSQ            0.0001
GSQ            0.0000

                                  INTERNAL
                X(3)          1            2
X(1)            X(2)
  1               1        161.10        78.91
                  2         78.90       161.09
  2               1        101.90       138.09
                  2         36.10       203.91


                                  EXTERNAL
                X(3)          1            2
X(1)            X(2)
  1               1        156.00        84.00
                  2         84.00       156.00
  2               1        107.00       133.00
                  2         31.00       209.00
```

This illustrates problem B; the basic change from problem A has been to input the original data into EXTERNAL rather than INTERNAL. As the hypothesis does not fix any value for two-way interaction the table is scaled on all 3 two-way margins.

References

Bishop, Y., Fienberg, S., Holland, P. (1975)
Discrete Multivariate Analysis. MIT.

Haberman, S. (1972). Algorithm AS51. Log-linear fit for Contingency Tables. App. Stat. 21, 2, 218-225.

Haberman, S. (1974) The analysis of frequency data Chicago Univ. Press

Jeffreys, M. (1965) An anatomy of social welfare services Michael Joseph.

T. M. Shih and J. Whittaker
University of Lancaster.

NEW FACILITIES FOR GENSTAT 3.09

Genstat 3.09 contains several additional facilities which will not be described in the current reference manual as a new manual is being produced to be released with Genstat 4.01. Brief descriptions of these new facilities are given here for those users who may find them helpful. These descriptions are listed under the appropriate sections for the reference manual.

Section 1. Syntax of the Genstat language

Zero repetition in lists

Pre-or post-repetition factors with value 0 can now be used. For example

```
'VALU' X=3(1, 2), 2 (6,0(7,8),9)
```

will set

```
X = 1,1,1,2,2,2,6,6,9,9
```

and in

```
'VALU' Y= M! (6), (3, 2)N
```

the value of either M or N may be zero;

```
If M = 3, N = 0        Y = 6,6,6
If M = 0, N = 2        Y = 3,2,3,2
```

Section 2. Program structure and execution

Looping over sets of structures

The Genstat loop mechanism, using 'FOR' and 'REPE', is a considerable advance over similar facilities provided in other high level languages. However it suffers from one limitation, it is impossible to loop over <u>sets</u> of structures. For example

```
'SETS' SET(1) = X(1...4)
  .    SET(2) = Y(1...3)
  .    SET(3) = Z(1...5)
'FOR'  SET = SET(1...3)
;
'REPEAT'
```

sets up a loop which is traversed 12 times, not 3, since by the standard rule for expanding sets at compile time the 'FOR' statement is equivalent to

```
'FOR'  SET = X(1...4), Y(1...3), Z(1...5)
```

If SET(1...3) had been declared as <u>pointers</u> instead of sets, the loop set up would be traversed only 3 times; but this would not solve the problem since pointers cannot be expanded into their components.

Ron Baxter of C.S.I.R.O., Australia suggested that 'SETS' should have a new option to allow pointers to be expanded. This has been implemented for 3.09, the option name is POINTER, its settings are P (default) for pointer or S for set. If

```
'POINTER' A = U,B,B,I
```

then

```
'SET/POIN=P'  X=R,A,S,H
```

would still set X = R,A,S,H since A would not be replaced by its components.

but

```
    'SET/POIN = S'  Y = R,A,S,H
```

would expand A and Y would now point to R,U,B,B,I,S,H

To loop over sets of identifiers,

```
    'POINTER'  SET(1) = X(1...4)
     :         SET(2) = Y(1...3)
     :         SET(3) = Z(1...5)
  * 'FOR'      SET = SET(1...3)
    'USE/R'    M $
    'REPEAT'
```

where

```
    'MACRO'    M $
 ** 'SET/POIN = S' S = SET
    'ENDM'
```

and within the body of the macro use the set identifier S to refer to the set of identifiers used.

This may appear to be clumsy, but it was essential to preserve the two basic rules.

SETS are always expanded by the compiler

POINTERS are not

and no better solution has been suggested.

Howard R. Simpson
R. E. S.

Section 6. Basic data operations

'CALCULATE'

Three new functions have been added

1. CUM

This calculates the cumulative totals of variates

Example:

The statements

```
'VARI' X(1) = 1...6
 :     X(2) = 2,4...12
'CALC' Y(1,2) = CUM(X(1,2))
```

gives Y(1) and Y(2) the following values

```
Y(1) = 1,3,6,10,15,21
Y(2) = 2,6,12,20,30,42
```

2. ELEM

This allows operations on specified elements of data structures

Examples:

The statements

```
'VARI' V = 1...10
```

```
        'CALC' V = V/ELEM(V;10)
```

gives V values .1,.2...1

The instructions

```
        'VARI' V = 1...4 : LST = 1...3
        'CALC' ELEM(V;LST+1) = ELEM(V;LST)
                        +ELEM(V;LST+1)
```

leaves V with values 1,3,5,7

The general form of the function is

```
        ELEM(ilist; expression)
```

The elements of the ilist can be identifiers of variates or matrices but not tables.

3. CHOL

The Choleski decomposition of a symmetric matrix A gives

A = L'L

This function returns the elements of L in the diagonal and lower off-diagonal locations of a square matrix the upper off-diagonal locations being filled with zeros.

Example:

```
        'MATRIX' B $ 6,6
        'SYMMAT' A $ 6
        'READ' A
        'CALC' B = CHOL(A)
```

To find positions of elements in a list

A new directive returns the positions of elements in lists of values.

```
        'POSITION' ilist 1 = ilist 2 $ ilist 3
```

Ilist 1 will contain positions of elements of ilist 2 in ilist 3. Ilist 1 can contain variates, integer structures or factors. Ilist 2 and ilist 3 can contain variates, integer structures, factors or names but values of structures in these two ilists must be of the same mode. If ilist 3 contains repeated values the position of the first occurence of the value will be taken.

Modification of a GROUP function

The RANK function can now have two arguments although the second is optional and any existing programs using this function will continue to work.

The general form of the function is now

```
        RANK(ilist 1 [; ilist 2] )
```

where the data structures in ilist 1 and ilist 2 must both be variates or both be name structures.

Example:

```
        'VARIATE'  X=4.7,5.2,5.2,6.4,6.8,6.8
        'FACTOR' F $ 4
        'GROUP' F = RANK(X;Y)
```

gives F values 1,2,2,3,4,4 as before and sets up values for Y of 1,2.5,4,5.5. The values of Y are the mid-ranks for each group. Thus the mid-rank for group 2 is 2.5 as elements associated with group 2 occur between positions 1 and 4 where the mid-rank is (1+4)/2. For simplicity the values of X are given here in ascending order though, of course, they do not have to be presented in this order.

Example:

```
      'NAME' X = B,A,E,C,A,B
      'FACT' F $ 4
      'GROUP' F = RANK(X;Y)
gives F values 1,2,3,4,2,1
and Y values B,A,E,C
```

Note that the labels are not ordered alphabetically.

Norman Alvey and Pam Tett
R. E. S.

Section 7. Regression

New regression facilities

In the last newsletter, advance notice was given of the extension of the regression section, and the effects this would have on the current facilities. The new facilities are available in release 3.09, but the description of how to use them will not generally be available until the new Genstat Manual is released next year.

The first extension is that model formulae can now be used in the existing directives. The statements below are used to analyse a factorial experiment where some of the data are missing,causing imbalance, if a missing-value analysis is not used. The dependent variate is AVERAGE, the average weekly weight gain of calves over an eight week period. INTAKE is a covariate: intake of food concentrates, HOUSE is a factor defining four different locations used in the experiment, and MILK_SUB is a treatment factor giving the level of milk substitution in each calf's diet.

```
'REGRESS' AVERAGE+INTAKE+HOUSE*MILK_SUB
'Y' AVERAGE
'FIT/PRIN = C, ANDEV = IT' INTAKE+HOUSE*MILK_SUB
```

The FIT statement fits the terms in the model formula in the order of expansion, i.e. INTAKE, HOUSE, MILK_SUB, HOUSE, MILK_SUB. The new option ANDEV produces an analysis of variance table, partitioning the regression sum of squares between each term in the model. As the design is not balanced, the order of fitting the terms is important. For example, the SS assigned to the term HOUSE should be interpreted as the SS due to

HOUSE eliminating INTAKE and ignoring MILK_SUB and HOUSE. MILK_SUB

The major extension is the inclusion of techniques to analyse Generalised Linear Models. Most of the facilities available in the interactive program GLIM have been incorporated. Some of the models that can now be analysed are:

1. Log-linear models for contingency tables
2. Quantal-response models for bioassay. Either the logit or the probit function may be used. However, direct estimation of LD 50's and fiducial limits are not available. It is hoped that a macro for calculating these quantities will be ready soon.
3. Series dilution experiments. If the solution density is supposed to be log-linear with time, then data from quantal response testing by the dilution method can be analysed.

4. Estimation of variance components.
5. Functions of data. For example, when an observed variate is considered to be Normally distributed, but the effects to be fitted are linear on a log scale, it is clearly unsatisfactory to fit a linear regression model to the log of the observed variate. The new facilities allow the original variate to be analysed, using a log 'link' function.
6. Split lines. Some split-line response curves can be fitted using an approximate technique.

Example:

Suppose we want to test a hypothesis of independence between the factors defining a 2 way contingency table. Say the factors are ROW and COLUMN, and the cell counts are stored in the variate COUNT.

```
'REGRESS' COUNT, ROW,COLUMN
'Y/ERROR = POISSON, LINK = LOG' COUNT
'FIT/PRIN = AU' ROW, COLUMN
```

The FIT statement will produce output under two headings. Firstly, the letter A will give the residual deviance, corresponding to the residual sum of squares in the linear regression model. For the log-linear model for contingency tables , the residual mean deviance is approximately χ^2, and this statistic is used to test for independence between the factors. The letter U will produce a table with columns observed value, fitted value and residual - an extension to the letter F, which gives fitted values and residuals only, printed across the page.

If you are interested in using these facilities before the new manual is released, please get in touch with me for more information.

Peter Lane
R. E. S.

Section 8. Analysis of Design Experiments

ORTH option

The ORTH option, which was made available in 3.06, allows a simplified dummy analysis to be used. With orthogonal designs, this simplified metod takes far less computing time than the full dummy analysis. The ORTH = Y option causes the program to use this method initially but, if any non-orthogonality is found, the program will switch to the full dummy analysis to allow the analysis to be completed. This is acceptable if the design really is non-orthogonal and ORTH had been set to Y as the result of the user mistakenly believing the design to be orthogonal. However, if the design contains many model terms the full dummy analysis may take a considerable amount of computing time, which would not be justified if the non-orthogonality were caused by a miss-punched factor value! Thus in 3.09 it is possible to specify ORTH = C (compulsary). If non-orthogonality is then discovered, the analysis terminates with diagnostic AN-14 and the program prints details of the first non-orthogonal term and the stratum being analysed.

ACCESSING PARTIAL EFFECTS

In designs with non-orthogonal treatment structures, sequential effects are printed by the program, if requested by the PR option (see the Reference manual 8.7 p1 and 8.8 p2). For example, if A and B are non-orthogonal in treatment formula A*B, the effects printed are for A ignoring B and then for B eliminating A. The PEFF list of EXTRACT, available in 3.09, allows tables of partial effects to be accessed. For example,if A and B are non-orthogonal as above, the statements

```
'TREATMENTS' A*B

'ANOVA' X ; OUT = XOUT

'EXTRACT' XOUT; A + B; PEFF = XA, XB
```

would store partial effects for A and B, from the analysis of variate X, in tables XA and XB respectively. i.e. XA contains effects of A eliminating B, and XB contains effects of B eliminating A (which, in this analysis, are the same as the sequential effects). The present bug in ANOVA which causes means containing effects from mutually non-orthogonal terms (as, for example, means for A.B above) to be printed incorrectly, is caused by the program using sequential rather than partial effects. A warning is printed when this occurs, and correct means can be obtained by extracting partial effects and using the table addition facilities in CALCULATE. The standard errors printed by ANOVA are appropriate for means formed from sequential effects, but these should be usable provided the efficiency factors of the non-orthogonal terms are not too different from 1. It is hoped to provide a more satisfactory solution to this problem in the next Genstat release.

To allow partial effects to be stored, it was necessary to alter the ACON structure of ANOVA, thus ACON structures saved from 3.08 (on backing store) will not be usable with 3.09. In such cases, the ACON should be devalued and the BLOCK and TREATMENT formulae redefined to allow a new ACON to be set up.

Roger Payne
R. E. S.

Section 10. Multivariate Analysis

New Criteria for Non-hierarchical Classification (CLASSIFY)

Genstat 3.09 will include two new criteria available for use with the CLASSIFY directive. These are : (1) minimisation of the determinant of the pooled within-class dispersion matrix, and (2) maximisation of the Mahalanobis' distances between class means. They can be used by setting the CRITERION option to W or T respectively.

If A is the pooled within-class sums of squares and products matrix and B is the total sums of squares and products matrix, then (1) is equivalent to minimising det (A) and (2) is equivalent to maximising trace (BA^{-1}). Friedman, H. P. and Rubin, J. (1967) J.A.S.A. 62, 1159-1186 and Scott, A. J. and Symons, M. J. (1971) Biometrics, 27, 217-219 discuss these criteria.

Colin Banfield
R. E. S.

Genstat Reference Manual

Amendment List No. 9 (for April 1973 Manual)
Amendment List No. 5 (for January 1975 Manual)

Note: * after line number indicates lines counted from bottom of page.

Page No.	Line No.	Amendment
7.3 p1	7 - 9	Lines should read "The Y directive initialises the set of independent variates to consist of an intercept only."
	12	Line should read "'X-set operation [/option list] [ilist1 of variates and/or factors] [; sequence of nameable lists]"
	18	Delete line 18
	19	Line should read " [INV = identifier]"
	26	Alter "nameable lists except VAR" to "nameable lists"
7.4 p2	3	Line should read "Ilist1 gives variates and factors"
7.4 p3	6	Alter "REGRESS directive" to "REGRESS or Y directive"
	15*	Line should read "not fitted and a message is printed"
	14* - 12*	Delete lines 14* - 12*
7.4 p4	12,13	Lines should read "defined by the order in which the corresponding terms were included in the model"
A1 p1		After "AN-7 Numerical analysis failure" add "AN-8 Out of date Acon AN-9 Negative or missing weight or missing factor value AN-10 Term only partially estimated AN-11 Unbalanced submodel AN-12 Illogical Compound submodel AN-13 Invalid term in EXTRACT list AN-14 Orthogonality check fails"
A1 p3	9*	Line should read "MV-7 CLASSIFY: input variates must be binary" After line 9* add "MV-8 CLASSIFY: Determinant is zero MV-9 ADPT: Error in coordinates input MV-10 Multivariate structure is of wrong type MV-11 Input structure has incorrect dimensions"

Page No.	Line No.	Amendment
A1 p3	16*	Before line 16* add "IO-15 Incompatible structures in parallel read"
A1 p5	10*,11*	Lines should read "VA-1 Incompatible subsets of values i.e. structures restricted in different ways"
	1*	After line 1* add "VA-11 Invalid or incompatible type(s) of data structure (s) VA-12 " " " mode(s) of values (e.g. real and integer) VA-13 " " " number(s) of values VA-14 " " " number(s) of factor levels VA-15 " " " classifying set(s) VA-16 " " " inclusion of margin(s) of tables"