Issue No. 17 *1986(A)*

The GENSTAT Newsletter

NAG
NUMERICAL ALGORITHMS GROUP

Editors

R W Payne
Rothamsted Experimental Station
Harpenden
Hertfordshire
AL5 2JQ

M G Richardson
NAG Central Office
Mayfield House
256 Banbury Road
Oxford
OX2 7DE

Printed and produced by the Numerical Algorithms Group

ISSN 0269-0764

GENSTAT NEWSLETTER
Issue No. 17

NP1190

1986 June

Contents

Fourth Genstat Conference

Enclosures

- Genstat Newsletter Order Form
- Genstat 4.03E Order Form
- Genstat 4.03E Price Schedule
- Genstat 4.03E Service Summary
- Genstat 4.03E Implementation List
- Genstat Newsletter Notice Board Sheet
- The Institute of Statisticians

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

Editorial

This issue of the Newsletter includes a further four papers from the Fourth Genstat Conference, including that of Dr J. Coursol, describing the version of Genstat which he has mounted on various micros, Genstat 4.03E. All the conference papers for which manuscripts were received have now been published.

NAG has now begun to market Genstat 4.03E and a product description and order form are enclosed. We believe that this will prove to be a very popular product which, we hope, will considerably expand the Genstat user base. The special terms being offered to educational sites should make it particularly attractive to those institutions with a 'teaching laboratory' equipped with a number of similar micros.

The new Genstat 4 Macro Library is also now available. To minimize costs, the Library and its documentation will be distributed together on magnetic tape. Full details are provided in the enclosed product information.

Corrigendum

16.9 Macro Library, Manual and Notice Board Amendment

Under **Macro Library: Error** the tenth line should begin:

and replace `VSET` by `N` in the next line.

(In the original correction, `N` was mistakenly rendered as `END`.)

Fitting a Weibull Distribution and Obtaining Corrected Standard Errors for the Parameter Estimates

J N S Matthews
Department of Biomathematics
University of Oxford
5 South Parks Road
Oxford
United Kingdom *OX1 3UB*

Introduction

By noticing that the kernel of the log-likelihood mimics that of a Poisson variate, Aitkin and Clayton (1980) were able to fit a variety of distributions to survival data using GLIM. The distributions included the exponential and Weibull and, when one of these is appropriate, the macros given by Aitkin and Francis (1980) are a very convenient way of fitting a regression model to the data.

In practice, one often finds that it is the Weibull density which would be appropriate and it is unforturnate that in this case the parameter standard errors given by GLIM are incorrect. Aitkin and Clayton give the method for correcting these values and, when the model is sufficiently simple (as in the first example in Aitkin and Clayton using data published by Gehan), these corrections are relatively easy to compute. However, one of the advantages of this approach is the ease with which complicated models can be fitted and, in these circumstances, finding the corrections can be very tedious. Roger and Peacock (1982) give a method which fits these models in such a way as to produce correct standard errors. However, for data classified by factors this method can also be cumbersome, as the user must define his own dummy variables.

The macros presented here allow the use of arbitrary model formulae and calculate the adjusted standard errors. This has been done by rewriting the macros of Aitkin and Francis in Genstat and using weighted SSP structures to allow straightforward calculation of the corrected standard errors.

Description of the Macro

The macro `WEIBULL` requires two variates and a model formula is input.

`T` – contains survival times,
`CEN` – contains censoring indicators; `0` = censored, `1` = uncensored,
`MD` – contains a model formula defined by a `'SET/LIST=M'`.

These quantities are unchanged by the macro. For simplicity it has been assumed that a `'UNIT'` statement, defining the length of `T` and `CEN`, is in force. Due to internal naming conventions and the way Genstat expands model formulae, it would be prudent if variates in the formula began with letters other than `A`. The fitting of the Weibull model is iterative and this version of the macro allows a maximum of 15 iterations; this value can be changed by altering the value of the local scalar `NIT`.

The macro first fits an exponential distribution and prints the fitted parameters; the standard errors are correct in this case. The Weibull model is then fitted and the parameters printed, together with their underestimated and corrected standard errors; the standard error of the shape parameter is also given.

Scaled residuals are available in `FV`. The macro `PRT` is simply a convenient place to keep some printing commands.

The Macro

```
'MACRO' WEIBULL $
'LOCAL' NIT,NUC,AA2,Z,DV,RDF,X,LL2,I,LL1,LL2,LL3,LL4
,AAA,SHSE,TVR,DTVR,TSE,SQP,SSY1,SSY2,SVM,C,CV,TVR2,IND,B
''

     MACRO TO FIT WEIBULL DISTRIBUTION TO CENSORED SURVIVAL
     TIMES, USING THE METHOD OF AITKIN AND CLAYTON.  THE
     OUTPUT INCLUDES CORRECTED STANDARD ERRORS FOR THE
     ESTIMATED PARAMETERS.

     INPUT  :   SURVIVAL TIMES IN T,
                CENSORING INDICATOR IN CEN (1=UNCEN, 0=CEN)
                IT IS ASSUMED THERE IS A UNIT STATEMENT
                IN FORCE FOR THE LENGTH OF T AND CEN.

''
'START'
'SCAL' C,NIT,RDF,NUC,SHP,Z,DV,X,I,SHSE,SSY1,SSY2
'CALC' AAA=1
'CALC' AA2=LOG(T+0.5*(T.EQ.0)) 'CALC' I,NIT,SHP=0,15,1
'CALC' NUC=SUM(CEN)
'CALC' Z=2*SUM(AA2*CEN) 'CALC' ZZ1=AA2
'TERM/LIMA=5,OFFSET=ZZ1' MD+CEN+ZZ1
'Y/ERROR=POISSON' CEN
'LINE' 2
'CAPT' '' EXPONENTIAL FIT **************''
'LINE' 3
'FIT/PRIN=C' MD ;FVAL=FV;DF=RDF;COEF=CV
'CALC' DV=Z-2*SUM(CEN*LOG(SHP*FV)-FV)
'CALC' SSY1=NVAL(CV) 'CALC' SSY2=SSY1+1
'R'
'USE' PRT $
'LINE' 5
'LABE' LL1
'CALC' I=I+1
'CALC' X=SUM(AA2*(FV-CEN))  'CALC' X=0.5*(SHP-NUC/X)
'JUMP' LL2*((X.LT.0.00001)*(X.GT.-0.00001))
'CALC' SHP=SHP-X  'CALC' ZZ1=SHP*AA2
'TERM/LIMA=5,OFFSET=ZZ1' MD+CEN+ZZ1
'Y/ERROR=POISSON' CEN
'FIT/PRIN=Z' MD ; FVAL=FV ; DF=RDF
'CALC' DV=Z-2*SUM(CEN*LOG(SHP*FV)-FV)
'CALC' RDF=RDF-(SHP.NE.1)
'USE/R' PRT $
'JUMP' LL3*(I.GT.NIT)
'JUMP' LL1
'LABE' LL2
'FIT/PRIN=C' MD ; FVAL=FV ; DF=RDF;VCOV=TVR2
'CALC' RDF=RDF-(SHP.NE.1)
'LINE' 5
'CAPT' '' CAUTION, THESE S.E. S ARE UNDERESTIMATED''
'SYMM' TVR $ SSY2 'DIAG' DTVR $ SSY1 'VARI' TSE $ SSY1
```

```
'VARI' IND,B $ SSY1
'CALC' IND=1 'CALC' IND=CUM(IND) 'CALC' IND=CUM(IND)+1
'VARI' SVM $ SSY2
'DSSP/LIMA=5' SQP $ AA2+AAA+MD
'SSP/WT=FV,WSP=WKSP' SQP
'EQUA' TVR,SVM,NIT=SQP
'DEVA' SQP,WKSP
'CALC' TVR=TVR+NIT*PDTT(SVM;SVM)
'CALC' B=ELEM(TVR;IND) 'CALC' C=ELEM(TVR;1)
'DEVA' TVR
'CALC' C=C+(NUC/(SHP*SHP))
'CALC' C=C-TPDT(B;PDT(TVR2;B))
'CALC' SHSE=SQRT(1/C)
'CALC' DTVR=TVR2+(1/C)*PDT(TVR2;PDT(B;TPDT(B;TVR2)))
'CALC' DTVR=SQRT(DTVR)  'EQUA' TSE=DTVR
'LINE' 2
'LINE' 3
'CAPT' '' ****** S.E. OF SHAPE PARAMETER *****''
'LINE' 2
'PRIN' SHSE
'LINE' 3
'CAPT' '' ******* ADJUSTED STANDARD ERRORS *****''
'LINE' 2
'PRIN' TSE
'JUMP' LL4
'LABE' LL3 'CAPT' ''****** NO CONVERGENCE ******''
'LABE' LL4
'ENDMACRO/LOCAL=DESTROY'


'MACRO' PRT $
'LINE' 2
'CAPT' ''DEVIANCE      DF       SHAPE PARAMETER''
'PRIN/P,LABR=1,LABC=1' DV,RDF,SHP $ 10.3,6.0,8.4
'ENDMACRO'
```

Example

The example below concerns the analysis of the persistence of the tear film on the cornea in an opthalmological trial. Each of the 6 patients (factor PAIR) was treated with two forms of drops (factor TRT), one type being administered to each eye. The tear film 'survival' time was noted pre-treatment and at six times post-treatment (factor TIME).

The output shown is purely for illustrative purposes, it is not meant to imply that such models should routinely fitted to data from six patients. Also, it is plainly essential that some form of residual plotting is essential; a Genstat equivalent of the RESPLOTS macro of Aitkin and Francis should be used.

```
GENSTAT V    RELEASE 4.04B
COPYRIGHT 1984 LAWES AGRICULTURAL TRUST (ROTHAMSTED EXPERIMENTALSTATION)

   1  'REFE/NID=200,NUNN=200'  WEIBM
   2  'UNIT' $ 168
   3  'FETCH/FILE=1' SURV
   4  'GET' SURV $ PWEIBULL
   5  'R'



   6  'FACT' PAIR $ 6=4(1...6)7 'FACT' TIME $ 7=24(1...7)
   7  'FACT' TRT $ 2= 2(1,2)42
   8  'FACT' READ $ 2=(1,2)84
   9  'READ/P' T,CEN
  10  'R'

    IDENTIFIER   MINIMUM      MEAN   MAXIMUM    VALUES   MISSING
             T      0.00     12.87     30.00       168         0
           CEN    0.0000    0.8929    1.0000       168         0  SKEW


 180  'SET/LIST=M' MD=PAIR+TIME*TRT
 181  'R'



 182  'USE' WEIBULL $


 EXPONENTIAL FIT ***************



182.....................................................................

***** REGRESSION ANALYSIS *****

ERROR DISTRIBUTION: POISSON   LINK FUNCTION: LOG
      Y-VARIATE: CEN
 OFFSET VARIATE: ZZ1
```

```
*** REGRESSION COEFFICIENTS ***
                                ESTIMATE        S.E.         T

CONSTANT                          -1.545       0.361       -4.28
PAIR 2                            -0.846       0.302       -2.80
PAIR 3                            -0.810       0.312       -2.60
PAIR 4                             0.047       0.287        0.16
PAIR 5                             0.227       0.287        0.79
PAIR 6                            -0.328       0.291       -1.13
TIME 2                            -1.955       0.477       -4.10
TIME 3                            -2.275       0.533       -4.27
TIME 4                            -1.543       0.429       -3.59
TIME 5                            -1.126       0.409       -2.75
TIME 6                            -0.591       0.418       -1.41
TIME 7                            -0.441       0.418       -1.05
TRT 2                              0.141       0.410        0.34
TIME 2 .TRT 2                      0.889       0.641        1.39
TIME 3 .TRT 2                      1.592       0.672        2.37
TIME 4 .TRT 2                      1.349       0.592        2.28
TIME 5 .TRT 2                      0.984       0.579        1.70
TIME 6 .TRT 2                      0.408       0.587        0.69
TIME 7 .TRT 2                      0.007       0.584        0.01
* STANDARD ERRORS BASED ON SCALE PARAMETER WITH VALUE   1.000
 183  'R'


DEVIANCE      DF       SHAPE PARAMETER

  1006.319   149  1.0000




DEVIANCE      DF       SHAPE PARAMETER

   921.688   148  2.0722


DEVIANCE      DF       SHAPE PARAMETER

   921.562   148  2.0346


DEVIANCE      DF       SHAPE PARAMETER

   921.556   148  2.0261
```

```
DEVIANCE      DF       SHAPE PARAMETER

   921.556   148  2.0243


DEVIANCE      DF       SHAPE PARAMETER

   921.556   148  2.0239


DEVIANCE      DF       SHAPE PARAMETER

   921.556   148  2.0238


DEVIANCE      DF       SHAPE PARAMETER

   921.556   148  2.0238

182.....................................................................

***** REGRESSION ANALYSIS *****

ERROR DISTRIBUTION: POISSON   LINK FUNCTION: LOG
      Y-VARIATE: CEN
 OFFSET VARIATE: ZZ1


*** REGRESSION COEFFICIENTS ***
                                ESTIMATE        S.E.         T

CONSTANT                          -3.531       0.381      -9.27
PAIR 2                            -1.318       0.322      -4.09
PAIR 3                            -1.186       0.337      -3.52
PAIR 4                             0.136       0.311       0.44
PAIR 5                             0.688       0.301       2.29
PAIR 6                            -0.340       0.307      -1.11
TIME 2                            -3.291       0.482      -6.83
TIME 3                            -3.610       0.537      -6.72
TIME 4                            -2.796       0.431      -6.49
TIME 5                            -2.213       0.410      -5.40
TIME 6                            -1.223       0.421      -2.91
TIME 7                            -0.947       0.422      -2.24
TRT 2                              0.091       0.415       0.22
TIME 2 .TRT 2                      1.460       0.645       2.27
TIME 3 .TRT 2                      2.332       0.679       3.44
TIME 4 .TRT 2                      2.542       0.593       4.28
TIME 5 .TRT 2                      2.103       0.585       3.60
TIME 6 .TRT 2                      0.876       0.600       1.46
TIME 7 .TRT 2                     -0.005       0.585      -0.01
```

```
* STANDARD ERRORS BASED ON SCALE PARAMETER WITH VALUE   1.000


CAUTION, THESE S.E. S ARE UNDERESTIMATED


****** S.E. OF SHAPE PARAMETER *****

        SHSE  1.2862E -1


******* ADJUSTED STANDARD ERRORS *****


       TSE
 4.6008E -1
 3.2678E -1
 3.3968E -1
 3.1135E -1
 3.0700E -1
 3.0692E -1
 5.0837E -1
 5.6132E -1
 4.5685E -1
 4.3162E -1
 4.2922E -1
 4.2755E -1
 4.1514E -1
 6.4826E -1
 6.8352E -1
 6.1040E -1
 6.0046E -1
 6.0216E -1
 5.8487E -1

184  'CLOS'
```

Acknowledgements

Clearly, this macro owes a great deal to the GLIM version by Aitkin and Francis and this debt is readily acknowledged. I am also grateful to Mr. Mengher and Dr. Pandher of the Oxford Eye Hospital for letting me use their data in the example.

References

[1] Aitkin, M. and Clayton, D.
The fitting of exponential, Weibull and extreme value distributions to complex censored survival data using GLIM.
Appl. Statist., **29**, 156-163, 1980.

[2] Aitkin, M. and Francis, B.
A GLIM macro for fitting the exponential or Weibull distribution to censored survival data.
GLIM Newsletter, **2**, 19-25, 1980.

[3] Roger, J.H. and Peacock, S.D.
Fitting the scale as a GLIM parameter for Weibull, extreme value, logistic and log-logistic regression models with censored data.
GLIM Newsletter, **6**, 30-37, 1982.

Fitting and Assessing a Non-Linear Response Curve with Genstat

P A Nunn
Inland Revenue
Statistics Division
West Wing
Somerset House
Strand
London
United Kingdom *WC2R 1LB*

Sigmoidal curves are widely used to represent many different biological responses to stimuli, especially growth of organisms over time. This article is concerned with modelling a particular system, namely the response of cereal grain protein content (measured as the percentage of nitrogen in dry matter) to applied fertilizer nitrogen. Some new methodology is developed which may be of more general interest. First, a method is presented for calculating a confidence region for the fitted curve and this is used to estimate the value of applied nitrogen at which we are 95% confident of attaining a given minimum protein content in the grain. Second, two measures of non-linearity are proposed to help in choosing between competing non-linear models. The smaller the measure of non-linearity, the closer the behaviour of the parameters of the non-linear model resembles that of a linear model. Thus for a close-to-linear model, its parameter estimates are almost unbiased, approximately Normal and have variances close to minimum; furthermore, the algorithms used by `OPTIMISE` will converge faster and more reliably.

After investigating a number of possible models, the following function was chosen:

$$\begin{aligned} Y &= \eta(X,\underline{\Theta}) \\ &= \alpha - \beta exp[-exp[-\gamma](X-\rho)^2]. \end{aligned}$$

This model, when fitted to a number of winter wheat datasets, has low correlation between parameters, low non-linearity and good convergence properties. Figure 1 shows a range of typical curve shapes. The model is also sufficiently flexible to model curves showing a depletion region, as in Figure 1b.

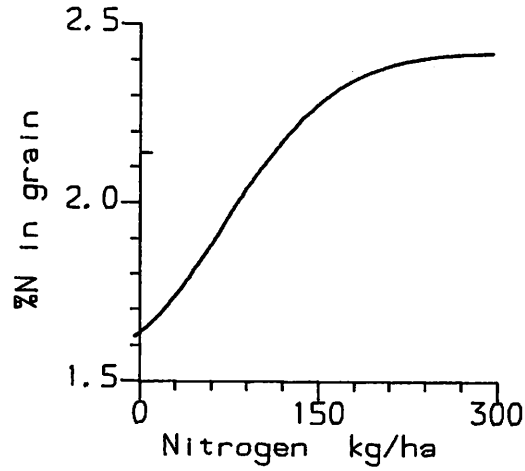


Figure 1a

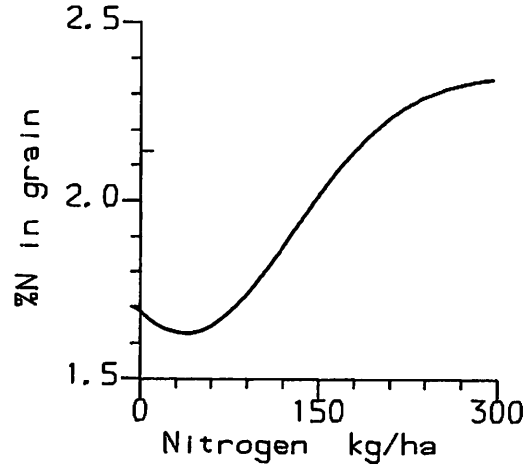


Figure 1b

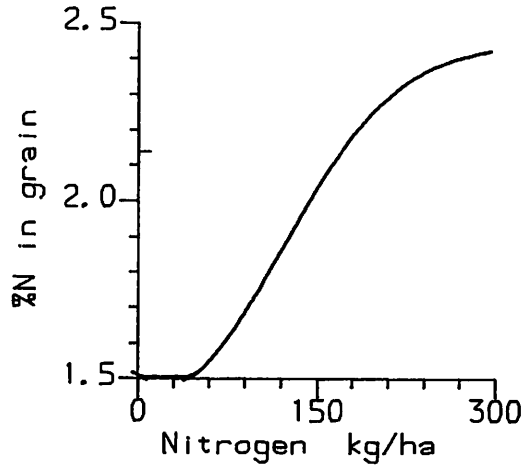


Figure 1c

This model has the shape of an upside down Gaussian curve with a variable origin, thus modelling the depletion region as in Figure 1b. It must be noted that, in cases where the depletion region is not apparent, it is possible to fit the curve the other way up, giving rise to local optima. This may be avoided by restricting β in the model to be positive.

Approximate confidence limits for the estimated response

Approximate confidence limits for the response may be estimated by using a first-order approximation to the model function. Thus:

$$\begin{aligned} Y &= \eta(X,\underline{\Theta}) \\ &\simeq \eta(X,\underline{\Theta}) + \sum_i (\Theta_i - \hat{\Theta}_i) D_i, \end{aligned}$$

where

$$D_i = \left.\frac{\partial \eta(X,\underline{\Theta})}{\partial \theta_i}\right|_{\underline{\Theta} = \hat{\underline{\Theta}}}$$

and

$$\underline{\Theta} = (\alpha,\beta,\gamma,\rho).$$

Then

$$Var(Y) = \sum_i D_i^2 \, var(\Theta_i) + 2 \sum_{i<j} D_i D_j \, cov(\Theta_i \Theta_j).$$

The effect of this approximation is to replace the solution locus by its tangent plane in the parameter space at the point $\hat{\underline{\Theta}}$ and simultaneously to impose a uniform coordinate system, on that tangent plane. Confidence limits may now be calculated for any level of applied nitrogen, using the expression for variance given above, since the D_i can be calculated and $var(\Theta_i)$ and $cov(\Theta_i \Theta_j)$ are given by Genstat. Often, we require lower one-sided confidence limits, since we may be concerned to avoid values below a certain threshold. Lower 95% confidence limits can be calculated for each level of applied nitrogen given in the data. A smooth curve can then be plotted by Genstat using the splining routines in GRAPH by setting the appropriate HEADING data structure to have the value S. Unfortunately, we do not have any analytic form for this curve and thus we cannot invert it to find the level of fertilizer nitrogen corresponding to our desired threshold value of protein content.

The following simple algorithm is proposed to overcome this difficulty:

(1) given $y_c = 2.14\%$ N in dry matter (this is the threshold value for wheat to be of bread-making quality), calculate x_c by inverting the fitted model,

(2) estimate $var(y_c)$ using the formula given above,

(3) find the lower 95% confidence limit, y_d, at the point x_c,

(4) shift the whole fitted curve down by an amount $(y_c - y_d)$ by subtracting this from the estimate of α,

(5) given $y_c = 2.14$, and using the new shifted response curve, calculate a new value x_c',

(6) repeat steps (2) to (5) using the new value x_c' until $y_c - y_d \leq 0.001$,

(7) the final value of x_c' is the desired 95% confidence limit.

For all the data sets used in this investigation (and for other potential models) the procedure converged rapidly, usually between four and six iterations were all that was required.

Measures of non-linearity

No standard criteria exist for the choice between competing non-linear models. Inspection of residuals, sum of squares, Student's t and correlations between parameters is always useful. Measures of non-linearity have been proposed to assess how far from linear are the properties of parameter estimates of non-linear models. Such a measure would provide additional information to guide our choice, as we would prefer a model whose parameter estimates behave in a manner which is close to linear. The smaller the value for non-linearity, the more justified is our replacement of the solution locus by its tangent plane with a uniform coordinate system, which we do in order to estimate confidence limits as described above.

Beale (1960) and Bates and Watts (1980) have suggested measures of non-linearity, but both are rather complex and require a considerable amount of code to program. Two simple measures on non-linearity are presented here:

(a) Radii of curvature, as given by

$$\texttt{CURV\_1} = \frac{|\ddot{\eta}_\gamma|}{|\dot{\eta}_\gamma|^2} \text{ and } \texttt{CURV\_2} = \frac{|\ddot{\eta}_\rho|}{|\dot{\eta}_\rho|^2}.$$

The larger these radii, the less non-linear the behaviour of estimates of parameters. For further detail see Bates and Watts (1980).

(b) The quantity `TOTAL = A_VAL + B_VAL + 2 * C_VAL`, obtained from the matrix of second derivatives of η calculated at a given level of applied nitrogen, say x_c, (the lower confidence limit for the level of fertilizer nitrogen to give the threshold protein content), with

$$\texttt{A\_VAL} = \frac{\partial^2 \eta(x_c, \underline{\Theta})}{\partial \gamma^2},$$

$$\texttt{B\_VAL} = \frac{\partial^2 \eta(x_c, \underline{\Theta})}{\partial \rho^2}$$

and

$$\texttt{C\_VAL} = \frac{\partial^2 \eta(x_c, \underline{\Theta})}{\partial \rho \partial \gamma}$$

The reasoning behind this runs as follows. We assume that the second order approximation (in the parameters) is a very good representation of the model function. For the first order approximation to be a good representation of the second order approximation, and therefore of the model function, we require the elements of the matrix of second derivatives of the model function with respect to the parameters, evaluated at x_c, to be as small as possible.

In cases where blocks or replicates are used, the curve may be fitted to the means over replications, thus increasing the speed of convergence. The variation between replicates can be used to estimate error and may thus be used to examine the model for goodness of fit.

Program Listing

```
'REFE/NUNN=400' Philip_Nunn
''
Program to fit Normal-type model (with fitted origin), plot the curve
together with its 95 % lower confidence limit, estimates the recommended
level of applied nitrogen and calculates the values A,B and C, and the
measures of curvature for the two non-linear parameters.
''
'UNIT'$ 12
'FACTOR' BLOCK $ 2= 6(1,2)
:        PLOT $ 6= (1...6)2
'VARI'             N_LEVELS=0,60...300
```

```
'FACTOR' NITROGEN$ N_LEVELS
'HEAD' HY=''%N in grain at 100% DM''
:      HX='' Nitrogen applied  kg/ha''
:      HD='' in grain at 100% DM''
:      HTN=''
  95% lower confidence limit of applied nitrogen for minimum 2.14 %N in DM''
:      HCUR=''
  First measure of non-linearity:
  (radii of curvature)''
:      HNL=''
  Second measure of non-linearity:
  (closeness of second-order to first-order approximation of model
function)''
:      HDI=''
   Values for D(i)''
:      SSP=''SSP''
'INPUT' 2
'HEAD' TITLE 'READ' TITLE 'PRINT' TITLE
'READ/P' NITROGEN,%N
'INPUT' 1
'DESC' %N$ 2; HD
'CALC' N=VARFAC(NITROGEN)
:      X=N/10
'SCALAR' ALPHA,BETA,GAMMA,RHO,TX,TN,TTX,ADX,DX,DY,TVY,DDY,DC,A_VAL,
B_VAL,C_VAL, DDD,TOTAL,V11,V12,V22,V13,V23,V33,V14,V24,V34,V44,XC,
D(1...4),VY1,VY2,VY3,CURV_1,CURV_2
'VARI' STEPLEN$ 2
:      VY,CLOW,FITS,MEANS$ N_LEVELS
'MODEL' IMPERIAL$ EY=-EXP(-EXP(-GAMMA)*(X-RHO)**2)
''
set initial values for non-linear parameters
''
'EQUA' GAMMA,RHO=5,2
'CALC' ELEM(STEPLEN;1)=0.02*GAMMA
:      ELEM(STEPLEN;2)=0.02*RHO
'OPTI/PRIN=PS,LIK=3,NPAR=2,CONST=Y' IMPERIAL; Y=%N; Z=EY;
PARAM=GAMMA,RHO,BETA,ALPHA; STEP=STEPLEN; FVAL=FITTED; VCOV=COVMAT
'EQUATE' V11,V12,V22,V13,V23,V33,V14,V24,V34,V44=COVMAT
:        XC=0
'FOR' NTIMES=1...6
'CALC' D(4)=1
:      D(3)=-EXP(-EXP(-GAMMA)*(XC-RHO)**2)
:      D(2)=BETA*2*(XC-RHO)*EXP(-GAMMA)*D(3)
:      D(1)=0.5*(XC-RHO)*D(2)
:      VY1=D(1)**2*V11+D(2)**2*V22+D(3)**2*V33+D(4)**2*V44
:      VY2=D(1)*D(2)*V12+D(1)*D(3)*V13+D(1)*D(4)*V14
:      VY3=D(2)*D(3)*V23+D(2)*D(4)*V24+D(3)*D(4)*V34
:      ELEM(VY;NTIMES)=VY1+2*(VY2+VY3)
:      XC=XC+6
```

```
'REPEAT'
'TABLE' TAB$ NITROGEN
'TABU' FITTED; MEAN=TAB
'EQUA' FITS=TAB
'TABU' %N; MEAN=TAB
'EQUA' MEANS=TAB
'CALC' CLOW=FITS-1.65*SQRT(VY)
'GRAPH/HY,HX' FITS,CLOW,MEANS; N_LEVELS $SSP
'EQUATE' DC,TTX=0
'LABEL' BEGIN_LOOP
'CALC' TX=RHO+SQRT(EXP(GAMMA)*LOG(BETA/(ALPHA-2.14)))
:      D(4)=1
:      D(3)=-EXP(-EXP(-GAMMA)*(TX-RHO)**2)
:      D(2)=BETA*2*(TX-RHO)*EXP(-GAMMA)*D(3)
:      D(1)=0.5*(TX-RHO)*D(2)
:      VY1=D(1)**2*V11+D(2)**2*V22+D(3)**2*V33+D(4)**2*V44
:      VY2=D(1)*D(2)*V12+D(1)*D(3)*V13+D(1)*D(4)*V14
:      VY3=D(2)*D(3)*V23+D(2)*D(4)*V24+D(3)*D(4)*V34
:      TVY=VY1+2*(VY2+VY3)
:      DY=1.65*SQRT(TVY)
:      DDY=DY-DC
:      ALPHA=ALPHA-DDY
:      DC=DY
:      DX=TX-TTX
:      ADX=ABS(DX)
:      TTX=TX
'JUMP' BEGIN_LOOP*(ADX.GT.0.001)
'CALC' TN=10*TX
'PRINT/C,VAR=1,LABR=1,LABC=1' HTN,TN$ 1,8.1
'CALC' DDD=BETA*EXP(-EXP(-GAMMA)*(TX-RHO)**2)
:      A_VAL=(D(1)+DDD)*EXP(-GAMMA)*(TX-RHO)**2
:      B_VAL=2*EXP(-GAMMA)*((TX-RHO)*D(2)+DDD)
:      C_VAL=2*(TX-RHO)*EXP(-GAMMA)*(D(1)+DDD)
:      TOTAL=A_VAL+B_VAL+2*C_VAL
:      CURV_1=ABS(A_VAL)/D(1)**2
:      CURV_2=ABS(B_VAL)/D(2)**2
'PRINT' HCUR
'PRINT/P' CURV_1,CURV_2$ 12.4
'PRINT' HNL
'PRINT/C,VAR=1,LABC=1,LABR=1' HDI,D(1),D(2),D(3),D(4)$ 10.4
'PRINT/P' A_VAL,B_VAL,C_VAL$ 12.4
'PRINT' TOTAL$ 12.4
'RUN'
'CLOSE'
'STOP'
```

Example data sets

Three data sets are given which correspond to the three types of curve illustrated in Figure 1a-c.

```
''Data set 1 (type a)''
    300     2.45
    240     2.41
    180     2.36
    120     2.15
     60     1.84
      0     1.63
      0     1.63
    240     2.37
    180     2.35
    300     2.43
     60     1.90
    120     2.17
   'eod'
''Data set 2 (type b)''
    120     1.91
    300     2.37
    180     2.17
      0     1.69
     60     1.62
    240     2.28
      0     1.74
    240     2.33
    120     1.92
    180     2.00
     60     1.60
    300     2.34
  'eod'
''Data set 3 (type c)''
    120     1.76
     60     1.51
    240     2.38
    300     2.43
      0     1.49
    180     2.15
    300     2.40
     60     1.56
    180     2.14
    120     1.96
    240     2.40
      0     1.56
   'eod'
```

References

[1] Bates, D.M. and Watts, D.G.
Relative curvature measures of non-linearity.
J.R.S.S.(B), **42**, 1-25, 1980.

[2] Beale, E.M.L.
Confidence regions in non-linear estimation.
J.R.S.S.(B), **22**, 41-88, 1960.

[3] Ratkowsky, D.A.
Non-linear Regression Modelling.
Macel Dekker Inc., New York, 1983.

A Program for Routine Analysis of Cereal Nitrogen Response Data

A W A Murray
Rothamsted Experimental Station
Harpenden
Hertfordshire
United Kingdom *AL5 2JQ*

Introduction

This article describes the use of Genstat to solve a problem which I am sure will have parallels in the experience of many consultant biometricians. Many of us have the task of producing analyses of a large number of broadly similar datasets and providing our clients with appropriate summaries of their data. I have tackled my problem in analysing a large number of fertilizer trials by building a general purpose program which requires minimal input from me to produce an analysis satisfactory to my clients.

I work in the statistics department at Rothamsted in a section providing a consultancy and data analysis service to the Agricultural Development and Advisory Service (ADAS). The soil science division of ADAS carries out a large number of cereal nitrogen response trials each year throughout England and Wales. These trials have from 6 to 9 levels of nitrogen application replicated in 2 or 3 randomised complete blocks. There may also be other factors, such as split or single dressings of nitrogen, fungicide or growth regulator treatments. ADAS's main interest is in predicting the optimum levels for application of nitrogen fertilizer in order to be able to provide advice to farmers. This is available through a published booklet or from ADAS local advisors.

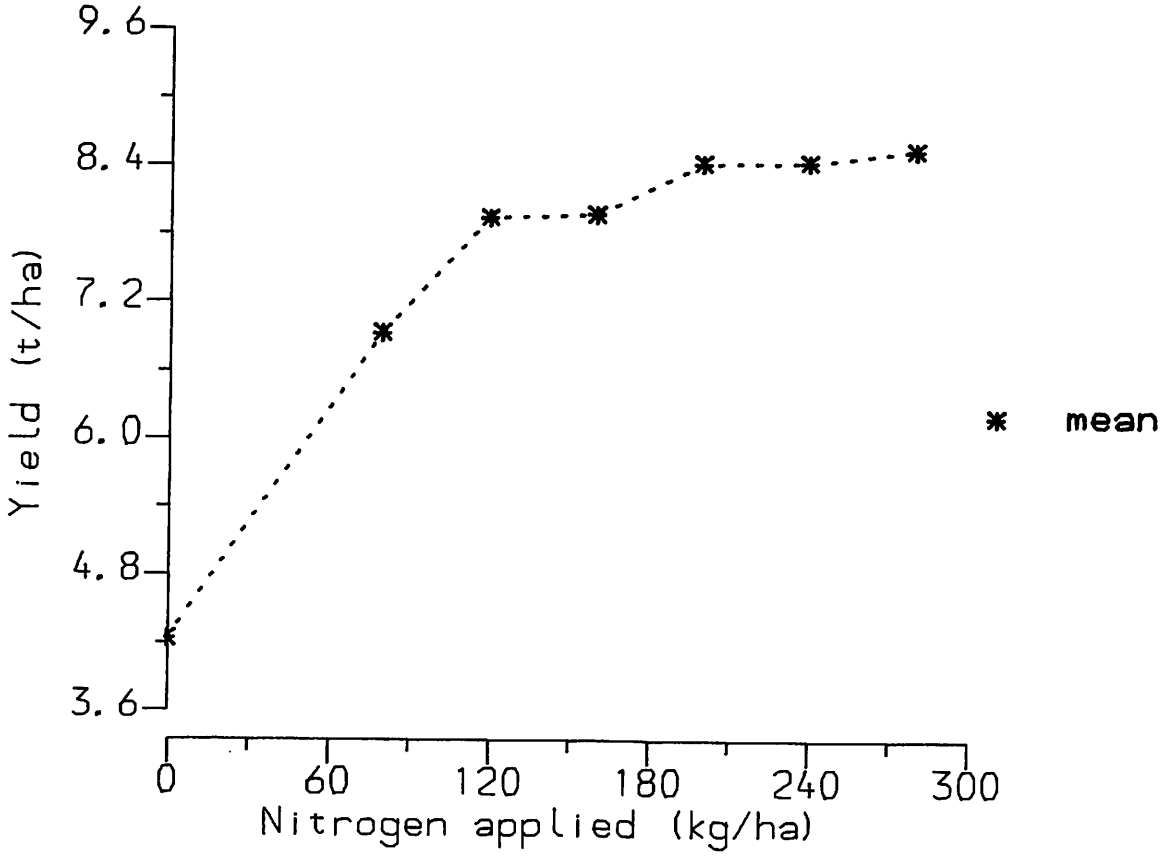
Figure 1 shows a plot of a typical winter wheat data set. Cereal crops generally show a similar pattern of diminishing returns to increasing applications of nitrogen fertilizer, so that beyond a certain point the value of the increased yield obtained is no longer sufficient to pay for the additional fertilizer required. This is the definition of *economic optimum* for the purposes of this analysis, we ignore other variable costs which may, indeed, alter according to the level of nitrogen fertilization employed. For instance, greater use of crop protection sprays might be required with high applications of nitrogen.

The results of these fertilizer trials form a large body of broadly similar data sets, differing mainly in details such as the actual levels of nitrogen applied and the number of plots employed. The program described here was initially designed to make life simpler for myself,

by providing a flexible system for the analysis of these data, and to provide appropriate summary information (such as an estimate of economic optimum) as requested by my clients. The program has now been refined to the stage where it requires very little input, other than raw data, and so it is suitable for 'self-service' use by responsible numerate scientists.

Description of the Program's Operation

A brief description follows of how the program is run and what output is produced. The bulk of the code is stored as macros in a backing store file. These are called from a short driver program. The user may edit this to alter the settings of certain options, for example to obtain additional graphical output for a pen plotter. A data file must be provided to be read on input channel 2 and this should consist of a heading to identify the data set, followed by the yields and nitrogen levels for each plot.



Boxworth after wheat 1981

Figure 1

The first stage of the analysis is to produce an analysis of variance with linear and quadratic orthogonal polynomials for nitrogen. If factors other than nitrogen are also present, then the driver program must be altered so that the values for these can be declared or read in, and the treatment and/or block formulae edited to produce the appropriate ANOVA. A table of plot residuals is printed in the order of the supplied data, which should usually be field order.

Next, a curve is fitted and a summary analysis of variance printed, as shown:

```
Boxworth NO (Backside)  (1981)
* * * * *   ANALYSIS OF VARIANCE FOR CURVE FIT   * * * * *
SOURCE OF VARIATION      DF          SS          MS          VR
BLOCK                     2      0.1902      0.0951
CURVE                     2     42.4634     21.2317      286.83
DEVIATIONS                4      0.2727      0.0682        0.92
RESIDUAL                 12      0.8883      0.0740
```

This is not standard Genstat output and I explain later how it is produced. This trial had 7 levels of applied nitrogen so there are 6 degrees of freedom available. These are broken down into 2 for the curve, and the remaining 4 associated with the deviations of the means from the fitted curve. Comparing these deviations with the residual error provides a test of whether or not the curve is a good fit to the data. The program automatically prints a warning if an F test of this variance ratio suggests that the deviations' mean square is significantly larger than the residual. Any other factors and all interactions are taken together and would appear, labelled as OTHER, in the table. This has to be set up by modifying a line in the driver program.

The curve fitted is a linear plus exponential (LPE):

general, non-linear version: $y = a + b.r^N + c.N$

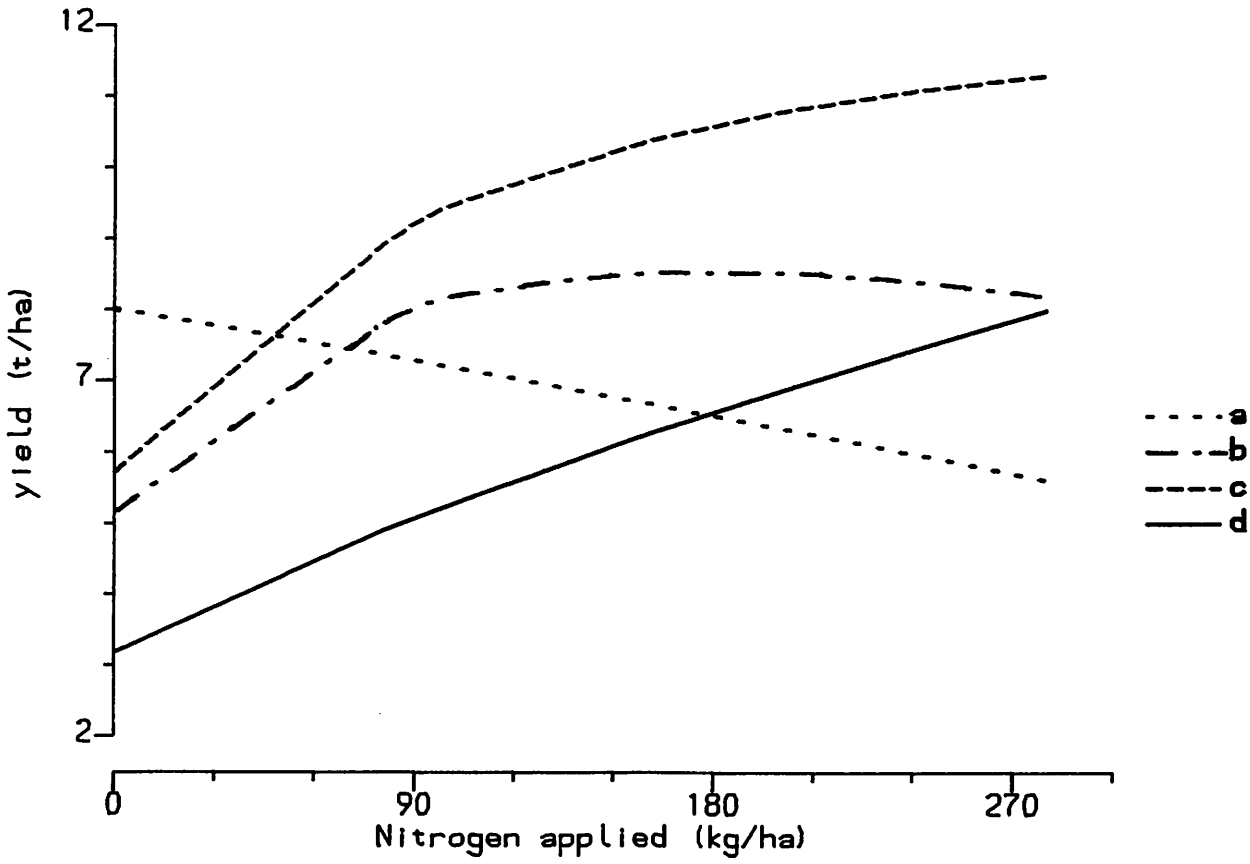
constrained, linear version: $y = a + b.0.99^N + c.N$

This curve is described in Sylvester-Bradley, Dampney and Murray (1984). Although the choice by default of a fixed value for r may appear somewhat arbitrary, this model has been found to give a good fit to a number of data sets arising from cereal nitrogen response experiments. It is, however, possible to set an option in the driver program to optimise the value of r where desired. Possible alternative models are discussed later.

The next part of the output is a table of the nitrogen levels, means, fitted values and residuals. The fitted curve is displayed on a graph.

The subsequent action of the program depends on the results of the curve fit. If the total mean square for nitrogen is not significantly different from error at $P \leq 0.1$, then fitting a curve is not really justified, so a warning message is printed and no further summary information provided.

The fitted curve can take on a variety of shapes as illustrated in figure 2.



Linear plus exponential curves
Figure 2

For instance, curve *a* has a maximum to the left of zero so that yields decrease for every level of applied nitrogen. In this case the optimum must be no application of nitrogen. Curve *b* increases for part of the range and then 'turns over'. Curves of this type must necessarily give an optimum in the range tested. In example *c*, the curve, although increasing everywhere on this range, nevertheless has a linear asymptote whose slope is less than the critical value. Curves of this type may have an economic optimum within the range of levels tested. Some curves, like example *d*, have everywhere a slope greater than the critical value so that an economic optimum cannot be calculated and must be assumed to be the maximum level tested. The program can recognise all these situations by testing the signs and magnitudes of the parameter estimates or functions of them. In each case the summary information printed is appropriate to the situation.

Four tables can be produced to summarise the information in the fitted curve. These are:–

(1) a table of parameter estimates;

(2) a table of economic optima for various price ratios of grain to fertilizer;

(3) a table of average slopes for various increments of applied nitrogen, and

(4) a table showing the nitrogen fertilizer required to achieve certain percentages of the predicted yield at optimum.

A specimen set of tables (1-4) is shown.

***** The fitted model is a linear plus exponential curve with
equation Y = A + B*(R**N) + C*N and coefficients as below *****

	A	9.7768	B	-5.5464	R	0.9900	C	-0.0033
Information useful to statisticians!			VAR_B	2.5262E -1	COV_BC	7.9407E -4	VAR_C	2.9384E -6

Table 1

***** Summary values calculated from the fitted curve ****

	Nitrogen Level	Estimated Yield	
'uncorrected' N_max	281	8.52	
'uncorrected' N_opt	202	8.38	
at an assumed fertiliser cost of	4.0	kg grain per kg N applied	
N_opt 'unbiased'	200	8.37	
at an assumed fertiliser cost of	4.0	kg grain per kg N applied	
'uncorrected' N_opt	217	8.44	
at an assumed fertiliser cost of	3.0	kg grain per kg N applied	
N_opt 'unbiased'	214	8.43	
at an assumed fertiliser cost of	3.0	kg grain per kg N applied	
'uncorrected' N_opt	234	8.48	
at an assumed fertiliser cost of	2.0	kg grain per kg N applied	
N_opt 'unbiased'	229	8.47	
at an assumed fertiliser cost of	2.0	kg grain per kg N applied	
Fitted yield at zero applied N		4.23	
Standard Error of N_opt	19.1		

N.B. Use the 'uncorrected' estimates for a one-off trial. However, in considering a series of trials, the 'unbiased' estimates are better because, in the long run, if we take this estimate we neither underestimate or overestimate the true value. A large difference between the two estimates may arise where the curve is very flat, the data are very variable, or the curve is not a good fit. A difference of about 20 kg/ha N or more would suggest that the estimates be treated with caution!

Table 2

***** Average slopes for certain increments of applied Nitrogen have been calculated from the fitted curve *****

Interval From	To	Average Slope (kg grain per kg N applied)	
217	257	1.9	N_opt to N_opt+40 kg/ha ('uncorrected')
214	254	2.1	N_opt to N_opt+40 kg/ha ('unbiased')
177	217	4.5	N_opt-40 kg/ha to N_opt ('uncorrected')
174	214	4.7	N_opt-40 kg/ha to N_opt ('unbiased')
0	217	19.4	zero to N_opt ('uncorrected')
0	214	19.6	zero to N_opt ('unbiased')
0	80	35.0	
80	120	17.2	

Table 3

***** Nitrogen applied to obtain a certain percentage of yield at economic optimum (unbiased) *****

% of yield at N_opt	Yield attained (t/ha)	Nitrogen applied (kg/ha)	% of N_opt ('unbiased')
95	8.00	145	68
90	7.58	110	52
75	6.32	52	24

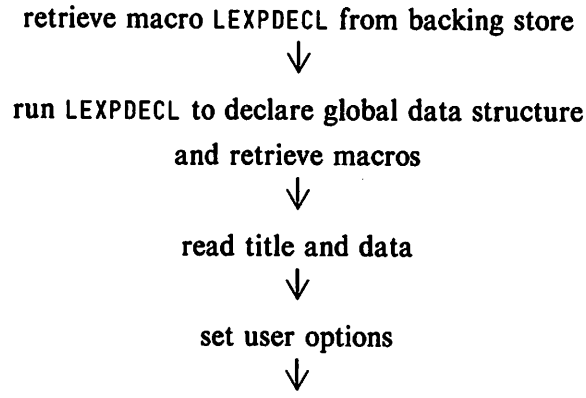
Table 4

The values in Table 4 cannot be found by solving an equation analytically but must be found by numerical methods. This is done using the direct function minimisation facility of 'OPTIMISE/LIK=1'.

Captions are printed to help the users understand the tables and to direct them to sources of further information and help, in case of difficulty.

How the Program is Written

The program has been through numerous revisions and one complete rewrite before reaching its present self-service version. The use of macros has considerably simplified the task of maintenance and modification.



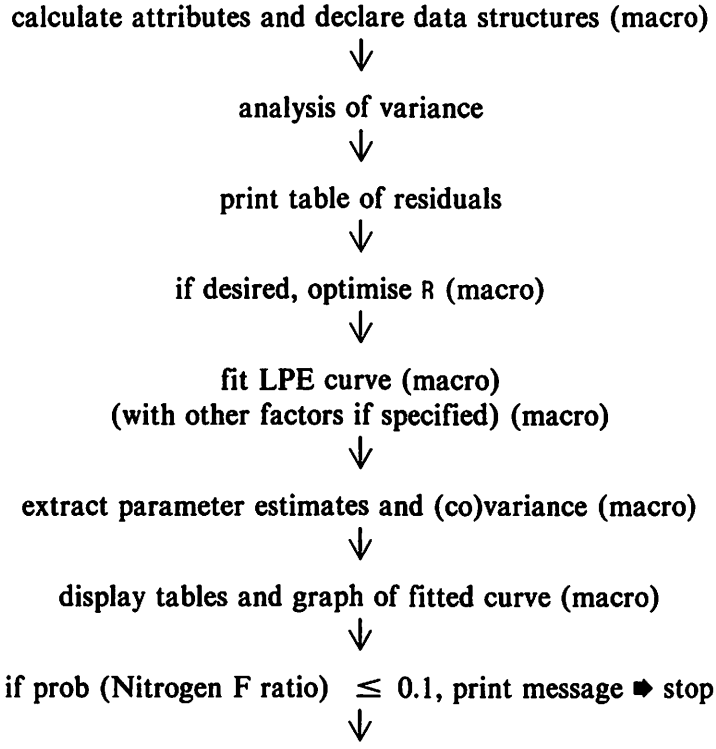
Overall structure of program
Figure 3

The structure of the program is shown in figure 3 and described below. Only two macro calls are visible to the user in the driver program. The first declares various global data structures and retrieves macros called later. The second macro, LEXP (shown in figure 4), produces the analysis described above. This macro calls various others in the course of its execution.

All macros are called by means of 'USE/R' so that they are not compiled until 'RUN' time. This makes it easy to interleave execution and compilation phases, so making it possible for the program to cope with data sets of any size. For instance, the first call is to a macro which sets up various data structures, whose dimension and other attributes depend on the data supplied to the program. This macro itself consists of several 'START' – 'RUN' blocks and, among other things, makes a factor NITROGEN from values read in as a variate and correctly labels the levels with the actual amounts of nitrogen applied. It sounds simple but, in fact, requires some tricks. This macro (which I often use in other programs) has been of great benefit to me because I no longer need to edit programs to take account of data sets of differing size and with different sets of nitrogen levels.

An important feature of the program is the tailoring of output to the clients requirements by modifying, or substituting for, Genstat standard output so as to make the analysis more easily understood by the non-statistician. An example of this is the analysis of variance table for the curve fit which I showed earlier. Standard regression output summary analysis of variance is not easy for the inexperienced to understand; that shown in the previous section is surely much clearer. This table is produced by the macro which fits the curve. In fitting the curve all standard output is suppressed by use of the PRIN=Z option with FIT directives. Residual degrees of freedom and sums of squares are saved at each stage of fitting. A few simple

calculations and a concatenated print of headings and scalars produces the customised analysis of variance.



Structure of macro LEXP

Figure 4

Macro LEXPOUT (see figure 5) produces appropriate summary information, depending on the shape of the curve as illustrated in figure 2. This shape can be determined by testing the signs and magnitudes of the parameter estimates and functions of them as shown in figure 5. This macro is some 420 lines long, including comments and captions, and was the most difficult part of the program to write. Genstat 4.04 has only fairly crude means to control the flow of command in a program. It should be much more straightforward to program the logical structure shown in figure 5 using the features planned for Genstat 5.

The above is merely an outline of the program. A full listing is available from the author on request.

print table of parameter estimates and (co)variances

↓

if B > 0 **then** print warning ➡ stop

else

∨

if C > 0 **then** goto C-POSITIVE

else

∨

if Nmax < 0 **then** print warning ➡ stop

else

∨

calculate and print summary information

↓

stop

C-POSITIVE

if C > critical value then assume Nopt is at maximum level tested

else

∨

calculate Nopt

↓

if Nopt not in range tested then assume Nopt is at maximum level tested

↓

calculate and print appropriate summary information

↓

stop

Structure of macro LEXPOUT

Figure 5

Concluding Remarks

This program is used by the ADAS section at Rothamsted and it has been distributed to certain, carefully vetted, ADAS users, who have found it quite straightforward to use. No serious bugs have (yet) been found.

I am not entirely happy with the LPE model for cereal nitrogen responses, especially with the version where r is arbitrarily constrained to be 0.99. Fitting other models to these data can fairly often give quite different estimates of the economic optimum nitrogen level for some experiments. The program is currently being extended to make available other models commonly used for the analysis of cereal nitrogen response data. I have written the code to fit the linear over quadratic model, one of the family of inverse polynomials described by Nelder (1966), and am working on the code for producing summary information from fitted curves of this type. Other models may be added in the future.

This article has described how Genstat can be used to provide powerful tools to assist the consultant biometrician in his day-to-day routine analyses. I hope it will encourage others to attempt similar solutions to their own problems.

References

[1] Nelder, J.A.
Inverse polynomials, a useful group of multi-factor response functions.
Biometrics, **22**, 128-141, 1966.

[2] Sylvester-Bradley, R., Dampney, P.M.R. and Murray, A.W.A.
The response of winter wheat to nitrogen.
Ministry of Agriculture, Fisheries and Food, Reference Book 385, The Nitrogen Requirement cereals, 151-174, London, HMSO, 1984.

A Genstat Macro for the Bivariate Analysis of Intercropping Data

*R F A Poultney**
J Riley
Statistics Department
Rothamsted Experimental Station
Harpenden
Hertfordshire
United Kingdom *AL5 2JQ*

* present address: *The Hatfield Polytechnic*
P.O. Box 109
College Lane
Hatfield
Hertfordshire
United Kingdom *AL10 9AB*

In the western world, farmers sow their crops in different fields; for instance, barley in one field and wheat in another. In the tropics and semi-arid tropics, farms are small yet the land must yield a sufficient variety of crops to feed the farmer's family for a whole year. To achieve this, farmers continue to practice a centuries-old system, called intercropping, whereby different crops are grown on the same small area of land but are mixed in their layout. For example, millet and sorghum may be grown in alternate rows, maize and beans may be sown so that the bean plants can grow up the stalks of the maize plants and grass for forage is often sown in the shade of timber and fruit trees. Agricultural researchers have realised that certain benefits can be gained from such a mixed-species system: not only can the yields be greater than they are when the species are grown alone in one field, but the soil between the plants is protected by the extra crop from the harsh tropical sun and monsoon rains, and smaller plants are shaded by the taller ones during the early parts of their life-cycles. Typical intercropping experiments are described in Riley (1985a,b).

Any intercropping experiment on two different species will provide two different sets of yields and a set of correlations between the yield values. One of the recognised analyses to deal with two correlated variables from intercropped plots was described by Pearce and Gilliver (1978). A description of this method, together with the Genstat instructions, is in Riley (1985a). Briefly, the two yield variates are transformed so that they become independent; treatment means can then be plotted between rectangular or skew axes and bivariate statistical techniques can be used to test for differences between them.

The macro described here calculates a bivariate analysis of variance for two intercrop variates from a designed experiment with a single error; modifications for designs with more than one error, such as a split-plot design, can easily be made but are unlikely to be needed since the use of such designs is rarely recommended for intercropping trials. The macro calculates and prints the bivariate analysis of variance and the bivariate *F* statistic for each of the treatment factors included in the experiment. The two variates are then transformed according to the method advocated by Pearce and Gilliver and the new treatment means are plotted between skew axes. Such a presentation displays the original treatment means with their underlying correlation eliminated. To aid the interpretation of the display even further, the graph is rotated so that both axes lie at equal angles to the vertical (Dear and Mead, 1984); this removes any likelihood of interpreting the graph in terms of horizontal and vertical distances. Finally, the radii are printed for circles representing standard errors, confidence regions and regions of non-significance.

The Macro

```
'MACRO' BIVAR$
''
      MACRO TO PRODUCE A BIVARIATE ANALYSIS OF VARIANCE AND ITS
      ASSOCIATED GRAPHS FOR INTERCROPPING DATA
       by  ROSIE POULTNEY and JANET RILEY
           STATISTICS DEPARTMENT
           ROTHAMSTED EXPERIMENTAL STATION
           HARPENDEN
           UK


*** SETTING UP LOCALS,SCALARS,VARIATES AND HEADINGS ***
                                                        ''
'LOCA' V12,TV11,TV22,Y1U,Y1L,Y2U,Y2L,A11,A12,A21,A22,R,V11,V22,VT2,VD2,
       X1,X2,X3,Y1,Y2,Y3,HX1,HX2,HY1,HY2,UY2X,MY2X,LY2X,Y1OUT,Y2OUT,
       UY1X,MY1X,LY1X,UY2Y,MY2Y,LY2Y,UY1Y,MY1Y,LY1Y,Y134,Y112,Y114,
       Y234,Y212,Y214,XX1,XUY2X,XMY2X,XLY2X,XX2,X2X,XLY1X,XMY1X,Y11,Y22,
       XUY1X,XX4,WIDTH1,XL,XU,YL,YU,XRANGE,YRANGE,ADJUST,WIDTH2,AH3,
       XAXIS,YAXIS,X,Y,BVS,VMY1,VMY2,YM,AAA,BBB,CCC,GAP,Q,AH1,AH2,AH4,
       TOTDAT,N,SER,CRR,NSR,FDF1(1...ICOUNT),FDF2(1...ICOUNT),YMAX,TOTAL,DIF,
       T12(1...ICOUNT),E12,TDF(1...ICOUNT),L(1...ICOUNT),F(1...ICOUNT),
       MOUT,SOUT,TOUT,DOUT,D1SS(1...ICOUNT),D2SS(1...ICOUNT),TSS(1...ICOUNT),
       DSS(1...ICOUNT),E1,E2,RESNOT,RESNOD,EDF,TDF,L,F,T1,T2,TSS,DSS,T12,FDF1,
       MY1(1...TABLENUM),MY2(1...TABLENUM),MY1,MY2,FORM,H1,H2,H3,FDF2,
       HPLOT,BVS,N1,N2,JY2U,JY234,JY212,JY214,JY2L,JY1L,JY114,JY112,
       JY134,JY1U,C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15,C16,
       CP1,CP2,CP3,CP4,CP5,CP6,CP7,CP8,CP9,CP10,CP11,CP12,CP13,CP14,CP15,
       ERCAPT,CORCAPT,VARCAPT,STACAPT,NSIGCAPT,CONCAPT,TNO,TCAPT,T2CAPT,
       T3CAPT,T4CAPT,LABELS,HPLUS,WL(1...ICOUNT),TABNUM,MEANNUM
```

```
'SCAL' V12,TV11,TV22,Y1U,Y1L,Y2U,Y2L,A11,A12,A21,A22,R,FDF1(1...ICOUNT),
       X1,X2,X3,Y1,Y2,Y3,HX1,HX2,HY1,HY2,UY2X,MY2X,LY2X,GAP,Q,FDF2(1...ICOUNT),
       UY1X,MY1X,LY1X,UY2Y,MY2Y,LY2Y,UY1Y,MY1Y,LY1Y,Y134,Y112,Y114,
       Y234,Y212,Y214,XX1,XUY2X,XMY2X,XLY2X,XX2,X2X,XLY1X,XMY1X,
       XUY1X,XX4,WIDTH1,XL,XU,YL,YU,XRANGE,YRANGE,ADJUST,YMAX,WIDTH2,
       T12(1...ICOUNT),E12,TDF(1...ICOUNT),L(1...ICOUNT),F(1...ICOUNT),
       TOTDAT,N,SER,CRR,NSR,WL(1...ICOUNT),TABNUM,MEANNUM
'VARI' XAXIS,YAXIS $ 3
     : X,Y $ 8
     : BVS $ 4
     : YM $ 3
'HEAD' AH1='' = DATA1 + DATA2''
     : AH3='' (DATA1 AFTER TRANSFORMATION)''
     : AH2='' = DATA1 - DATA2''
     : AH4='' (DATA2 AFTER TRANSFORMATION)''
     : FORM = ''PLPPTTTTTTTTTTTT''
     : H1 = ''Y1''
     : HPLOT = ''*''
     : H2 = ''Y2''
     : LABELS =''-''
     : H3 = '' ''
     : HPLUS = ''+''
     : C1='' error sum of products = ''
     : C2='' on ''
     : C3='' and''
     : C4='' degrees of freedom ''
     : C5='' treatment sum of products = ''
     : C6=''treatment ''
     : C7='' bivariate F-statistic for treatment = ''
     : C8='' Wilks' lambda = ''
     : C9=''==============''
     : C10='' variances after adjustment for covariance are ''
     : C11='' coefficient of correlation between original data = ''
     : C12='' radius of standard errors = ''
     : C13='' on 2 and ''
     : C14='' radius of confidence regions = ''
     : C15='' x square root of tabulated F-value on 2 and ''
     : C16='' radius of non-significance = ''
''

*** BIVARIATE ANALYSIS OF VARIANCE ***

''
'START'
'DESC' TOTAL $; AH1
     : DIF $;AH2
     : Y11 $; AH3
     : Y22 $; AH4
'CALC' TOTAL = DATA1 + DATA2
     : DIF = DATA1 - DATA2
```

```
''
    *** analysis of variance of DATA1 and DATA2 and their totals
        and differences ***
''
'ANOVA' DATA1,DATA2,TOTAL,DIF;OUT=MOUT,SOUT,TOUT,DOUT
''
    *** extraction of variances,sums of squares and degrees of
        freedom ***
''
'EXTR' MOUT;MODELT$VAR=V11
     : SOUT;MODELT$VAR=V22
     : TOUT;MODELT$VAR=VT2
     : DOUT;MODELT$VAR=VD2
     : MOUT;MODELT$SS=D1SS(1...ICOUNT)
     : SOUT;MODELT$SS=D2SS(1...ICOUNT)
     : TOUT;MODELT$SS=TSS(1...ICOUNT)
     : DOUT;MODELT$SS=DSS(1...ICOUNT)
     : MOUT;MODELB$SS=E1
     : SOUT;MODELB$SS=E2
     : TOUT;MODELB$SS=RESNOT
     : DOUT;MODELB$SS=RESNOD
     : MOUT; MODELT$ DF = TDF(1...ICOUNT)
     : MOUT; MODELB$ DF = EDF
''
    *** calculation of
        covariance,
   variances after adjustment for covariance,
   error sum of products,
   correlation coefficient,
   transformation of DATA1 and DATA2 ***
''
'CALC' V12 = (VT2 - VD2)/4
     : TV11,TV22 = V11,V22 - (V12*V12/V22,V11)
     : E12 = (RESNOT - RESNOD)/4
     : R = V12/SQRT(V11*V22)
     : A11 = 1/SQRT(2*V11*(1-R))
     : A12 = -1/SQRT(2*V22*(1-R))
     : A21 = 1/SQRT(2*V11*(1+R))
     : A22 = 1/SQRT(2*V22*(1+R))
     : Y11,Y22 = A11,A21*DATA1 + A12,A22*DATA2
'RUN'
''

*** CALCULATION OF STATISTICS ***


   *** loop containing calculations of
       the treatment sum of products,
       Wilk's lambda,
       and the bivariate F statistic ***
''
```

```
'START'
'FOR' TDF=TDF(1...ICOUNT);L=L(1...ICOUNT);F=F(1...ICOUNT);T1=D1SS(1...ICOUNT);
   T2=D2SS(1...ICOUNT);TSS=TSS(1...ICOUNT);DSS=DSS(1...ICOUNT);
   T12=T12(1...ICOUNT);FDF1=FDF1(1...ICOUNT);WL=WL(1...ICOUNT);
   FDF2=FDF2(1...ICOUNT)
'CALC' T12 = (TSS - DSS)/4
     : L=((T1+E1)*(T2+E2) - (T12+E12)**2) / ((E1*E2) - E12**2)
     : WL=1/L
     : F=(SQRT(L)-1)*((EDF-1)/TDF)
     : FDF1 = TDF*2
     : FDF2=2*(EDF-1)
'DEVA' T1,T2,TSS,DSS
'REPE'
'CALC' TABNUM = TABLENUM - 1
'RUN'
''

*** CALCULATION OF GRAPH CONTENTS ***


    *** obtaining means to be plotted ***
''
'START'
'ANOVA' Y11,Y22 ; OUT = Y1OUT,Y2OUT
'EXTR' Y1OUT;MODELT$MEAN=(*)TABNUM,VMY1
     : Y2OUT;MODELT$MEAN=(*)TABNUM,VMY2
''
    *** calculation of
       radius of standard error
       radius of confidence regions
       radius of non-significance ***
''
'CALC' TOTDAT=NVAL(DATA1)
     : MEANNUM=NVAL(VMY1)
     : N=TOTDAT/MEANNUM
     : SER=SQRT(1/N)
     : CRR=SQRT(2/N)
     : NSR=SQRT(4/N)
''
    *** calculation of transformed axes and the position for
        annotation on them  ***
''
'CALC' Y1U,Y2U = MAX(DATA1,DATA2)
     : Y1L,Y2L = MIN(DATA1,DATA2)
     : WIDTH1=INTPT(LOG10(ABS(Y1U)))+7.2
     : WIDTH2=INTPT(LOG10(ABS(Y2U)))+4.2
     : Q = 1.5+0.25*(INTPT(LOG10(ABS(Y2U))))+0.5*(R.GT.0.4)-0.5*(R.LT.-0.4)
     : X1,X2,X3 = A11*Y1L,Y1L,Y1U + A12*Y2U,Y2L,Y2L
     : Y1,Y2,Y3 = A21*Y1L,Y1L,Y1U + A22*Y2U,Y2L,Y2L
     : YMAX = MAX(VMY2)
```

```
'CALC' HY1,HY2 = (Y3,Y1 - Y2)/2 + Y2 - 1
     : HX1,HX2 = (X3,X2 - X2,X1)/2 + X2,X1 + 1,-2
     : MY2X,UY2X,LY2X = 1,3,1*(X2 - X1)/2,4,4 + X1
     : MY1X,UY1X,LY1X = 1,3,1*(X3 - X2)/2,4,4 + X2
     : MY2Y,UY2Y,LY2Y = 1,3,1*(Y1 - Y2)/2,4,4 + Y2
     : MY1Y,UY1Y,LY1Y = 1,3,1*(Y3 - Y2)/2,4,4 + Y2
     : Y134,Y112,Y114 = 3,1,1*(Y1U - Y1L)/4,2,4 + Y1L
     : Y234,Y212,Y214 = 3,1,1*(Y2U - Y2L)/4,2,4 + Y2L
     : XX1,XUY2X,XMY2X,XLY2X,XX2 = X1,UY2X,MY2X,LY2X,X2 - Q
     : X2X,XLY1X,XMY1X,XUY1X,XX4 = X2,LY1X,MY1X,UY1X,X3
'EQUA' X = X1,UY2X,MY2X,LY2X,LY1X,MY1X,UY1X,X3
     : Y = Y1,LY2Y,MY2Y,UY2Y,LY1Y,MY1Y,UY1Y,Y3
     : YM = Y1,Y3,YMAX
     : XAXIS = X1,X2,X3
     : YAXIS = Y1,Y2,Y3
'RUN'
''
    *** positioning of axes in centre of frame ***
''
'CALC' XL = INTPT(X1 - 0.5)
     : XU = INTPT(X3 + 1)
     : YL = INTPT(Y2)
     : YU = INTPT(MAX(YM))
     : XRANGE = XU - XL
     : YRANGE = YU - YL
'JUMP' AAA*(XRANGE.EQ.YRANGE) + BBB*(XRANGE.LT.YRANGE)
'CALC' ADJUST = (XRANGE - YRANGE)/2
     : GAP = XRANGE/4
     : YU = (YU+ADJUST) + GAP
     : YL = (YL-ADJUST) - GAP
     : XU = XU + GAP
     : XL = XL - GAP
'JUMP' CCC
'LABEL' BBB
'CALC' ADJUST = (YRANGE - XRANGE)/2
     : GAP = YRANGE/4
     : XU = (XU+ADJUST) + GAP
     : XL = (XL-ADJUST) - GAP
     : YL = YL - GAP
     : YU = YU + GAP
'JUMP' CCC
'LABEL' AAA
'CALC' GAP = XRANGE/4
     : YU,XU = YU,XU + GAP
     : YL,XL = YL,XL - GAP
'LABEL' CCC
'EQUA' BVS = YL,YU,XL,XU
''
    *** placing calculated values for axes annotation into
        headings to be plotted ***
''
'SET/F' N1 = WIDTH1
      : N2 = WIDTH2
```

```
'JOIN/1,FMT=N2' JY2U = Y2U
      : JY234 = Y234
      : JY212 = Y212
      : JY214 = Y214
      : JY2L=Y2L
'JOIN/1,FMT=N1' JY1L=Y1L
      : JY114 = Y114
      : JY112 = Y112
      : JY134 = Y134
      : JY1U = Y1U
'PAGE'
''
    *** forming and plotting text output from headings ***
''
'JOIN/1'CP1=E12 $ 8.4
     : CP2=EDF $ 4
     : ERCAPT=C1,CP1,C2,CP2,C4
     : CP3=R$7.4
     : CORCAPT=C11,CP3
     : CP11=TV11$8.4
     : CP12=TV22$8.4
     : VARCAPT=C10,CP11,C3,CP12
     : CP13=SER$8.4
     : CP14=CRR$8.4
     : CP15=NSR$8.4
     : STACAPT=C12,CP13,C13,CP2,C4
     : CONCAPT=C14,CP14,C15,CP2,C4
     : NSIGCAPT=C16,CP15,C15,CP2,C4
'PRIN' ERCAPT
     : VARCAPT
     : CORCAPT
     : STACAPT
     : CONCAPT
     : NSIGCAPT
'FOR'T12=T12(1...ICOUNT);TDF=TDF(1...ICOUNT);TNO=(1...ICOUNT);F=F(1...ICOUNT);
    FDF1=FDF1(1...ICOUNT);WL=WL(1...ICOUNT);FDF2=FDF2(1...ICOUNT)
'JOIN/1'CP4=TNO $ 4
      : CP5=T12 $ 8.4
      : CP6=TDF $ 4
      : CP7=F $ 8.4
      : CP8=FDF1$4
      : CP9=FDF2$4
      : CP10 = WL$8.4
      : TCAPT = C5,CP5,C2,CP6,C4
      : T2CAPT= C7,CP7,C2,CP8,C3,CP9,C4
      : T3CAPT=C6,CP4
      : T4CAPT=C8,CP10
'LINE' 4
'PRIN' T3CAPT
     : C9
'LINE' 1
```

```
'PRIN' TCAPT
     : T4CAPT
     : T2CAPT
'DEVA' TCAPT,T2CAPT,T3CAPT,T4CAPT
'REPE'
'PAGE'
''

*** GRAPH INSTRUCTIONS FOR LINE PRINTER AND PLOTTER ***

''
'GRAPH/ATY=H3,ATX=H3,BV=BVS,NRF=61,NCF=101' VMY2,YAXIS,
Y,Y2,HY2,HY1,Y1,LY2Y,MY2Y,UY2Y,Y2,Y2,LY1Y,MY1Y,UY1Y,Y3;VMY1,XAXIS,
X,X2,HX2,HX1,XX1,XUY2X,XMY2X,XLY2X,XX2,X2X,XLY1X,XMY1X,XUY1X,XX4
$ FORM;HPLOT,*,LABELS,HPLUS,H2,H1,JY2U,JY214,JY212,JY234,JY2L,
JY1L,JY114,JY112,JY134,JY1U
'OUTPUT' 2
'GRAPH/ATY=H3,ATX=H3,BV=BVS,NRF=61,NCF=101,DEVICE=1,BUFF=N' VMY2,YAXIS,
Y,Y2,HY2,HY1,Y1,LY2Y,MY2Y,UY2Y,Y2,Y2,LY1Y,MY1Y,UY1Y,Y3;VMY1,XAXIS,
X,X2,HX2,HX1,XX1,XUY2X,XMY2X,XLY2X,XX2,X2X,XLY1X,XMY1X,XUY1X,XX4
$ FORM;HPLOT,*,LABELS,HPLUS,H2,H1,JY2U,JY214,JY212,JY234,JY2L,
JY1L,JY114,JY112,JY134,JY1U
'OUTPUT' 1
'ENDM/LOCALS=DESTROY'
```

Method of Use

The user must supply the block and treatment factors for the experiment in the usual way. A factor must be created to identify the plots within blocks and the BLOCK directive must recognise this structure so that the residual sum of squares can be easily extracted. Two model statements must be declared in the following way:

'SET/M' MODELT = *treatment formula*

'SET/M' MODELB = *block formula*

where the block formula identifies the block.plot structure.The two variates for analysis must be read in and called DATA1 and DATA2. Missing values in either variate can be accommodated and restricted variates can be used. In addition, two scalars must be declared: TABLENUM, the number of the table in the ANOVA output from which the means are to be taken, and ICOUNT, the number of treatments and interactions for which bivariate F statistics are to be calculated. This can be less than the total number of treatment and interaction terms.

The macro produces two output files. The first contains the required analyses of variance and calculated statistics together with a line-printer graph. The second contains the instructions for producing the graph on a graph plotter. To produce a graph of size k inches $\times$ p inches, NRF should be set to $6k + 1$ and NCF should be set to $10p + 1$.

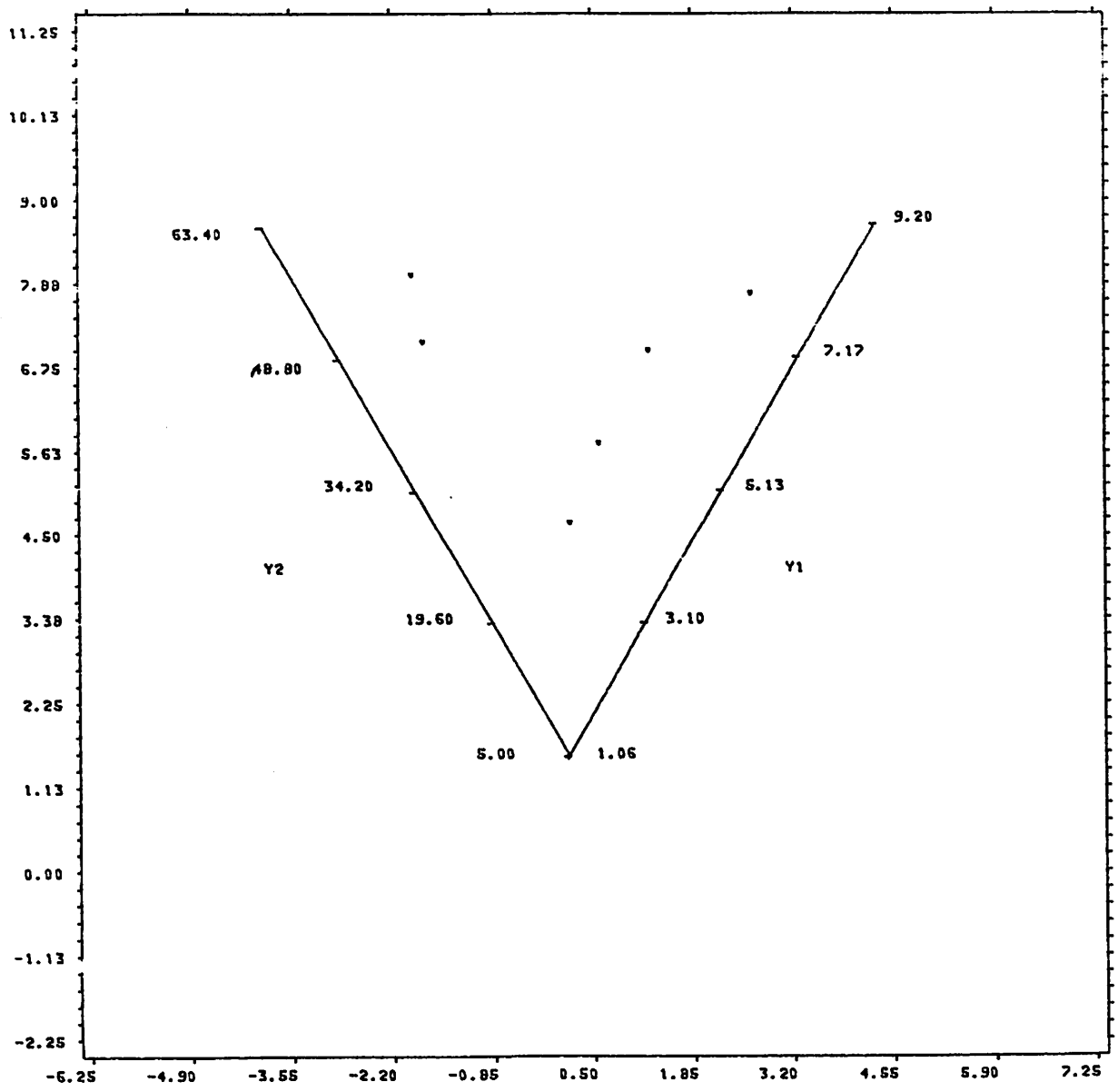
Example

The data below, yields of the crop in kg/plot, come from Dear and Mead (1983). In the experiment, two melon spacings were combined with two okra varieties in four randomised blocks. The results are from three successive cropping seasons; the consistency of the variances from individual season analyses was tested by Dear and Mead and found to be acceptable for a combined analysis.

			OKRA (Kg/plot)				MELON (Kg/plot)			
	M1	01	1.06	3.28	2.18	1.98	63.4	29.0	45.4	46.4
YEAR1		02	2.52	1.96	1.56	1.67	44.6	42.7	50.2	45.9
	M2	01	5.58	3.08	2.35	2.07	37.0	53.9	56.9	39.0
		02	1.90	2.40	2.59	4.10	53.1	47.9	28.3	20.1
	M1	01	5.4	6.2	7.0	5.8	32.0	20.0	20.2	19.0
YEAR2		02	5.6	7.8	8.3	6.9	9.5	16.5	6.8	18.5
	M2	01	4.4	4.0	5.0	4.2	18.0	20.5	12.8	14.0
		02	7.1	6.6	9.2	4.8	21.0	16.5	5.0	10.0
	M1	01	3.13	2.30	2.21	2.30	16.6	22.9	15.1	16.7
YEAR3		02	2.81	3.30	4.81	5.80	10.3	34.3	13.3	12.6
	M2	01	2.70	4.34	2.50	3.61	20.5	22.7	14.8	11.8
		02	4.70	4.90	2.30	2.44	20.1	19.4	23.3	21.1

M1 and M2 are the melon spacings, 01 and 02 are the okra varieties and YEAR1, YEAR2 and YEAR3 are the cropping seasons.

The graph produced here is of okra variety against years.



To calculate the radii of standard errors, non-significance regions and confidence regions, we first find the value of F-distribution at, say, the 5% level on the degrees of freedom given in the output. Using this we can calculate the radii as

radius of standard error = square root(0.125) = 0.3535

radius of confidence regions = square root(0.25 x 3.285) = 0.9062

radius of non-significance regions = square root(0.5 x 3.285) = 1.2816

These can be drawn on the graph with the same scale as the square plotting axes.

Example Output

```
GENSTAT V    RELEASE 4.04B
COPYRIGHT 1984 LAWES AGRICULTURAL TRUST (ROTHAMSTED EXPERIMENTAL STATION)

  1  'REFE/NUNN=200,NID=300'EXAMPLE
  2  'OUTPUT'1$80
  3  'UNIT' $ 48
  4  'SCAL' TABLENUM = 6
  5       : ICOUNT = 7
  6  'NAME' MNAM = M1,M2
  7       : ONAM = O1,O2
  8       : YNAM = YEAR1,YEAR2,YEAR3
  9  'FACT' MELON $ MNAM
 10       : OKRA $ ONAM
 11       : YEAR $ YNAM
 12       : BLOCK $ 4 = (1...4)12
 13       : PLOT $ 12 = 4(1...12)
 14  'TREA' MELON*OKRA*YEAR
 15  'BLOCK' BLOCK/PLOT
 16  'SET/M' MODELT = MELON*OKRA*YEAR
 17         :MODELB=BLOCK.PLOT
 18  'INPUT' 2
 19  'READ/S' DATA1,DATA2
 20  'INPUT' 1
 21  'GENE' YEAR,MELON,OKRA,4
 22  'RUN'


  IDENTIFIER   MINIMUM      MEAN   MAXIMUM    VALUES   MISSING
       DATA1     1.060     4.015     9.200        48         0

  IDENTIFIER   MINIMUM      MEAN   MAXIMUM    VALUES   MISSING
       DATA2      5.00     26.24     63.40        48         0


 26  'MACRO' BIVAR$
```

```
 338  'USE' BIVAR$


VARIATE: DATA1

SOURCE OF VARIATION               DF          SS       SS%
          MS        VR

BLOCK STRATUM                      3       1.264      0.67       0.421

BLOCK.PLOT STRATUM
  MELON                            1       0.020      0.01       0.020     0.016
  OKRA                             1       7.833      4.18       7.833     6.085
  YEAR                             2     114.748     61.17      57.374    44.575
  MELON.OKRA                       1       0.020      0.01       0.020     0.016
  MELON.YEAR                       2       7.589      4.05       3.795     2.948
  OKRA.YEAR                        2       9.436      5.03       4.718     3.666
  MELON.OKRA.YEAR                  2       4.214      2.25       2.107     1.637
  RESIDUAL                        33      42.475     22.64       1.287

TOTAL                             44     186.337     99.33       4.235


GRAND TOTAL                       47     187.601    100.00

GRAND MEAN                              4.02
TOTAL NUMBER OF OBSERVATIONS              48


***** TABLES OF MEANS *****

VARIATE: DATA1


    GRAND MEAN     4.02

         MELON       M1       M2
                   3.99     4.04


         OKRA        01       02
                   3.61     4.42


         YEAR     YEAR1    YEAR2    YEAR3
                   2.52     6.14     3.38


         OKRA        01       02
        MELON
           M1      3.57     4.42
           M2      3.65     4.42
```

YEAR	YEAR1	YEAR2	YEAR3
MELON			
M1	2.03	6.63	3.33
M2	3.01	5.66	3.44

YEAR	YEAR1	YEAR2	YEAR3
OKRA			
O1	2.70	5.25	2.89
O2	2.34	7.04	3.88

OKRA	O1			O2		
YEAR	YEAR1	YEAR2	YEAR3	YEAR1	YEAR2	YEAR3
MELON						
M1	2.13	6.10	2.48	1.93	7.15	4.18
M2	3.27	4.40	3.29	2.75	6.93	3.59

***** STANDARD ERRORS OF DIFFERENCES OF MEANS *****

TABLE	MELON	OKRA	YEAR	MELON OKRA
REP	24	24	16	12
SED	0.328	0.328	0.401	0.463

TABLE	MELON YEAR	OKRA YEAR	MELON OKRA YEAR
REP	8	8	4
SED	0.567	0.567	0.802

***** STRATUM STANDARD ERRORS AND COEFFICIENTS OF VARIATION *****

STRATUM	DF	SE	CV%
BLOCK	3	0.187	4.7
BLOCK.PLOT	33	1.135	28.3

.
.
.

Similar output produced for DATA2, TOTAL (=DATA1 + DATA2),
DIFF (=DATA1 - DATA2) and Y11 (DATA1 after transformation).

.
.
.

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

VARIATE: Y22 (DATA2 AFTER TRANSFORMATION)

SOURCE OF VARIATION	DF	SS	SS%	MS	VR
BLOCK STRATUM	3	7.392	6.24	2.464	
BLOCK.PLOT STRATUM					
MELON	1	0.421	0.36	0.421	0.421
OKRA	1	1.204	1.02	1.204	1.204
YEAR	2	54.046	45.60	27.023	27.023
MELON.OKRA	1	0.000	0.00	0.000	0.000
MELON.YEAR	2	6.419	5.42	3.210	3.210
OKRA.YEAR	2	9.005	7.60	4.503	4.503
MELON.OKRA.YEAR	2	7.029	5.93	3.515	3.515
RESIDUAL	33	33.000	27.84	1.000	
TOTAL	44	111.125	93.76	2.526	
GRAND TOTAL	47	118.517	100.00		

GRAND MEAN	6.71
TOTAL NUMBER OF OBSERVATIONS	48

***** TABLES OF MEANS *****

VARIATE: Y22 (DATA2 AFTER TRANSFORMATION)

GRAND MEAN 6.71

MELON	M1	M2
	6.80	6.62

OKRA	O1	O2
	6.55	6.87

YEAR	YEAR1	YEAR2	YEAR3
	7.55	7.37	5.21

OKRA	O1	O2
MELON		
M1	6.64	6.96
M2	6.46	6.77

YEAR	YEAR1	YEAR2	YEAR3
MELON			
M1	7.35	7.98	5.08
M2	7.74	6.76	5.35

YEAR	YEAR1	YEAR2	YEAR3
OKRA			
01	8.00	6.98	4.67
02	7.10	7.75	5.75

OKRA	01			02		
YEAR	YEAR1	YEAR2	YEAR3	YEAR1	YEAR2	YEAR3
MELON						
M1	7.45	8.12	4.34	7.26	7.83	5.81
M2	8.54	5.84	5.00	6.94	7.67	5.69

***** STANDARD ERRORS OF DIFFERENCES OF MEANS *****

TABLE	MELON	OKRA	YEAR	MELON OKRA
REP	24	24	16	12
SED	0.289	0.289	0.354	0.408

TABLE	MELON YEAR	OKRA YEAR	MELON OKRA YEAR
REP	8	8	4
SED	0.500	0.500	0.707

***** STRATUM STANDARD ERRORS AND COEFFICIENTS OF VARIATION *****

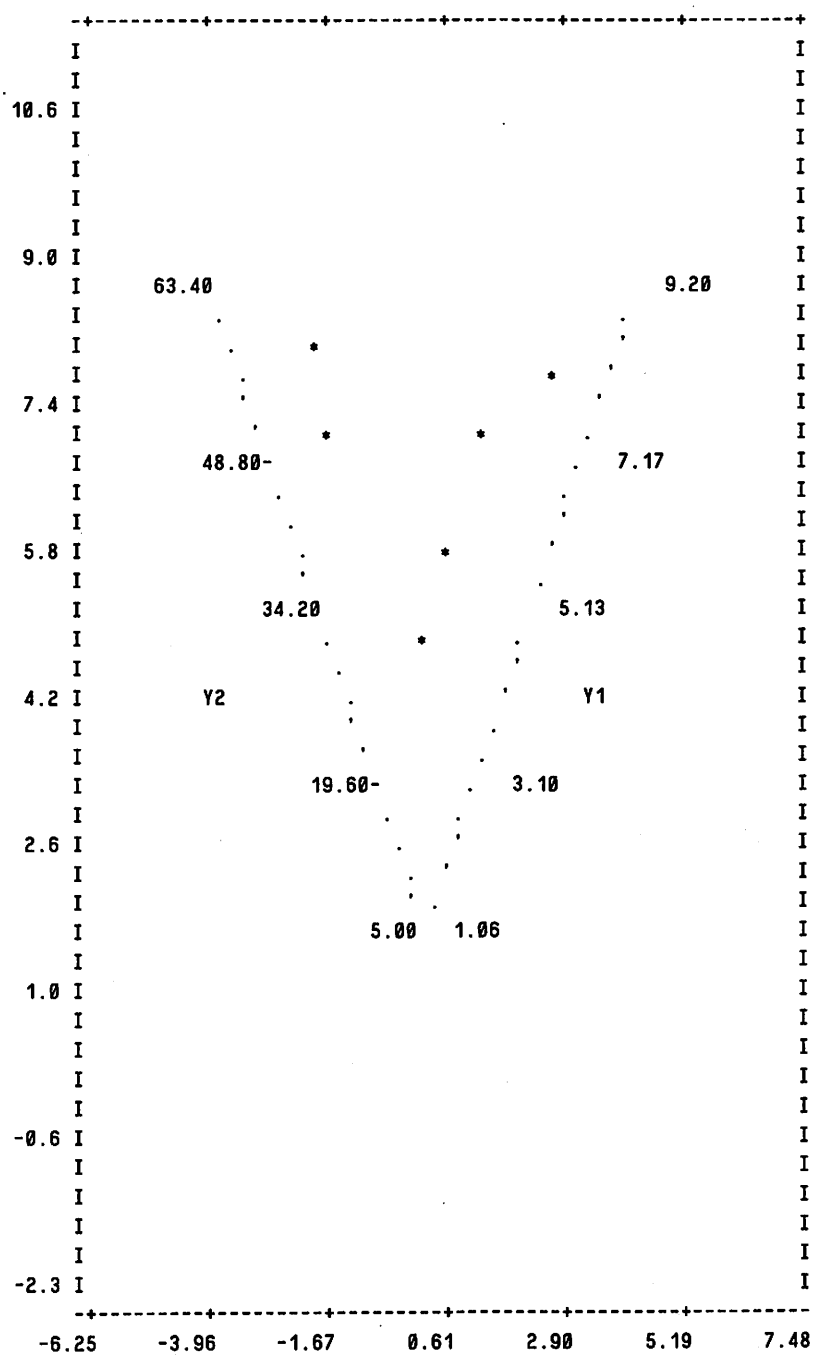
STRATUM	DF	SE	CV%
BLOCK	3	0.453	6.8
BLOCK.PLOT	33	1.000	14.9

```
 339  'RUN'
```

```
 error sum of products =-152.3318  on  33 degrees of freedom

 variances after adjustment for covariance are  0.9707 and 50.7860

 coefficient of correlation between original data =-0.4958

 radius of standard errors =  0.3536 on 2 and  33 degrees of freedom

 radius of confidence regions =  0.5000 x square root of tabulated F-value on 2
 and  33 degrees of freedom

 radius of non-significance =  0.7071 x square root of tabulated F-value on 2 and
   33 degrees of freedom


treatment   1

===============

 treatment sum of products = -0.9116  on   1 degrees of freedom

 Wilks' lambda =  0.9794

 bivariate F-statistic for treatment =  0.3346  on   2 and  64 degrees of freedom


treatment   2

===============

 treatment sum of products =-31.3472  on   1 degrees of freedom

 Wilks' lambda =  0.8437

 bivariate F-statistic for treatment =  2.8376  on   2 and  64 degrees of freedom
```

```
treatment   3
==============

 treatment sum of products =-686.4392  on   2 degrees of freedom
 Wilks' lambda =  0.0847
 bivariate F-statistic for treatment = 38.9696  on   4 and  64 degrees of freedom


treatment   4
==============

 treatment sum of products = -0.1114  on   1 degrees of freedom
 Wilks' lambda =  0.9995
 bivariate F-statistic for treatment =  0.0084  on   2 and  64 degrees of freedom


treatment   5
==============

 treatment sum of products = -2.0094  on   2 degrees of freedom
 Wilks' lambda =  0.7837
 bivariate F-statistic for treatment =  2.0738  on   4 and  64 degrees of freedom
```

```
 treatment   6

 ===============

  treatment sum of products = -2.2524  on   2 degrees of freedom

  Wilks' lambda =  0.7171

  bivariate F-statistic for treatment =  2.8945  on   4 and  64 degrees of freedom



 treatment   7

 ===============

  treatment sum of products =  7.8725  on   2 degrees of freedom

  Wilks' lambda =  0.7949

  bivariate F-statistic for treatment =  1.9463  on   4 and  64 degrees of freedom
```




```
340  'CLOSE'

******** END OF EXAMPLE.  MAXIMUM OF 20518 DATA UNITS USED AT LINE 338 (12250 LEFT)
```

Acknowledgement

The authors produced this macro while they were funded by the U.K. Overseas Development Administration.

References

[1] Dear, K.B.G. and Mead, R.
The use of bivariate analysis techniques for the presentation, analysis and interpretation of data.
Statistics in Intercropping Technical Report No.1, Department of Applied Statistics, University of Reading, U.K, 1983.

[2] Dear, K.B.G. and Mead, R.
Testing assumptions and other topics in bivariate analysis.
Statistics in Intercropping Technical Report No.2, Department of Applied Statistics, University of Reading, U.K, 1984.

[3] Pearce S.C. and Gilliver, B.
The statistical analysis of data from intercropping experiments.
J. Agric. Sci., Camb., **91**, 625-632, 1978.

[4] Riley, J.
Genstat by Post.
Genstat Newsletter, **15**, 14-19, 1985a.

[5] Riley, J.
A Genstat Analysis of Intercropping Stability.
Genstat Newsletter, **16**, 29-37, 1985b.

The Use of Pseudo-Factors when Treatments were Superimposed in an Orchard Experiment

D A Preece
East Malling Research Station
East Malling
Maidstone
Kent
United Kingdom *ME19 6BJ*

In experimentation in orchards, a set of trees often has to be used in successive years to test successive sets of treatments. Superimposing a new set of treatments on the layout for a previous set can be a difficult design problem, if residual effects of the previous set are to be allowed for. Such superimposition is usually attempted only if the possible residual effects can be assumed not to interact with the effects of the new treatments. The residual effects may be of little or no interest in themselves, so long as they can be eliminated in an analysis providing efficient estimates of current effects. Indeed, the past treatments can then well be regarded as constituting the levels of a blocking factor rather than of a treatment factor. This view of things can be a guide on how to randomise a new set of treatments subject to the superimposition having the desired design properties. However, a purist approach to randomisation often has to be abandoned for non-orthogonal superimpositions.

In early 1985, forty-eight apple trees were available at East Malling Research Station for use

as single-tree plots for testing 12 fungicides (1, 2, ..., 12). These trees had been used in 1984 to test 6 treatments (A, B, ..., F), with 8 trees per treatment. The design for the treatments A to F was itself part of a larger design obtained by superimposing A to F on an even earlier design. However, by early 1985 it was believed that the only past blocks or treatments which needed to be taken into account for the forty-eight trees were the treatments A to F.

The 1985 design was required at short notice and was produced in a hurry, more by intuition than by logical reasoning. The combinations of new treatments with old were as indicated by the 1s in the following table (the incidence matrix):

	D	F	A	C	E	B
7	1	1		1		1
5		1	1	1	1	
12	1		1		1	1
6	1		1	1	1	
8	1	1			1	1
4		1	1	1		1
9	1	1		1		1
2		1	1	1	1	
10	1		1		1	1
11	1		1	1	1	
3	1	1			1	1
1		1	1	1		1

The dotted lines indicate the structure of the design; the orders of the numerals and the letters were obtained by randomisation.

The 'moment of truth' came when data from this design were presented for analysis. However, Genstat analyses were easily obtained by defining blocking factors (BLOCK1 and BLOCK2) and pseudofactors (PF1, PF2 and PF3) as follows, where FUNGICD denotes the fungicides factor:

				BLOCK1	1	1	1	2	2	2
				BLOCK2	1	2	3	1	2	3
				1984 Treatment	D	F	A	C	E	B
PF1	PF2	PF3	FUNGICD							
1	1	1	7		1	1		1		1
1	1	2	5			1	1	1	1	
1	1	3	12		1		1		1	1
1	2	1	6		1		1	1	1	
1	2	2	8		1	1			1	1
1	2	3	4			1	1	1		1
2	1	1	9		1	1		1		1
2	1	2	2			1	1	1	1	
2	1	3	10		1		1		1	1
2	2	1	11		1		1	1	1	
2	2	2	3		1	1			1	1
2	2	3	1			1	1	1		1

As the factor FUNGICD is orthogonal to the factor BLOCK1, only 4 of the degrees of freedom for the 1984 treatments can have components of FUNGICD confounded with them. By inspection,

these components are PF3, with 2 d.f., and the interaction PF2.PF3, also with 2 d.f. The BLOCK and TREATMENT codes can be neatly specified as follows:

```
'BLOCK' BLOCK1.BLOCK2
'TREATMENT' TREAT//(PF1*PF2*PF3)
```

Alternatively, the BLOCK code can be given as either BLOCK1/BLOCK2 or BLOCK1*BLOCK2; the three possibilities produce analyses with, respectively, 2, 3 and 4 strata. If the analysis with 4 strata is chosen, the INFORMATION SUMMARY in the ANOVA output gives the following for the bottom stratum:

MODEL TERM	E.F.	NON-ORTHOGONAL TERMS
PF3	0.938	BLOCK2
PF2.PF3	0.812	BLOCK1.BLOCK2

Here, the final line is to be read as meaning that the interaction between the pseudofactors PF2 and PF3 is confounded with the interaction between the blocking factors BLOCK1 and BLOCK2, the efficiency factor for the treatment component being 0.812.

Thus, Genstat has not only produced a satisfactory analysis of variance but has also reported the confounding as clearly as one could wish – and the hurriedly produced design served its purpose well.

Survey of Genstat Users – Preliminary Report

M G Richardson
NAG Central Office
Mayfield House
256 Banbury Road
Oxford
United Kingdom *OX2 7DE*

About 600 copies of two related questionnaires on Genstat usage were distributed at the Fourth Genstat Conference (at York) and with Genstat Newsletter 15. 159 responses had been received by the end of April and these have been analysed in part. A summary of the findings to date appears below and it is hoped that a full analysis will appear in the next Newsletter.

Respondents were asked to estimate the typical number of jobs run per day, week, month or year (by themselves or at their site). Converted to annual figures (with daily, weekly and monthly figures being multiplied by 220, 44 and 11, respectively) the totals given correspond to a grand total of 213,000 Genstat jobs run per year. As individual and site responses were received in a ratio of about 4:3 and as, on average, only one response has been received for every two Genstat sites, the total throughout the world is probably far in excess of this figure.

On the second version of the questionnaire, respondents were also asked to estimate the typical number of unsuccessful Genstat jobs submitted before a successful run. Replacing answers expressed as a range by the midpoint of that range gives a mean value of 3.3, a median of 2.25, an interquartile range of 1.5 to 4 and a range of 0 to 20.

Areas of Genstat application fell into 34 categories, the top scores being for biology (67 respondents), agriculture (62) and statistical research (53). A wide range of other areas were indicated by between 1 and 31 respondents.

Respondents were asked which of 26 Genstat facilities they regularly used. The most popular were linear regression, with 133 respondents, and ANOVA with one error term, with 123. The least used was Procrustes rotation, with 7. Most respondents use a fairly extensive range of Genstat facilities. (The mean was 9.4 categories, the median 8.)

To see the usage of Genstat in terms of broader categories, the responses were reclassified as follows (the numbers indicate how many respondents checked one or more facilities in the group).

(a – d)	Regression and GLMs	139
(f – i)	Analysis of designed experiments	129
(t – u)	Language features	100
(w – x)	Macros	82
(l – p)	Multivariate analysis	72
(v)	Quality graphics	65
(e)	OPTIMIZE	53
(y – z)	Backing store	39
(j – k)	Cluster analysis	34
(q – s)	Time series	30

Of the 70 respondents who answered the question on usage of library macros, 27 use MANOVA. The next most widely used macro was CANCOR, with 16 users, and the least popular were ALIAS, GPROCPLT and GPROCLAB, which were not checked by any respondents.

60 other statistical packages were used by respondents or at their sites; 42 of these were mentioned by only one respondent. The most widely used were Minitab (71 respondents), GLIM (69), SPSS/SPSS-X (41), BMDP and MLP (33 each) and SAS (32). A preliminary analysis of the needs which respondents indicated as being met by these packages reveals that the commonest category of answer is simplicity (of teaching, learning or use), which was mentioned by 71 respondents. A wide range of fairly specialised facilities were mentioned by a similar number of respondents and the next commonest category was interaction, mentioned 42 times.

Respondents were asked to indicate the order of importance which they attached to possible enhancements in various aspects of Genstat. The most popular first choice was documentation, (51 out of 145 unique first choices). To summarise the responses, a score of 3 was ascribed to each (unique) first choice, 2 to a (unique) second choice and so on, down to −3 for a seventh choice, producing the following scores for each category:

Documentation	+232
Simplified syntax	+ 93
Statistical facilities	+ 73
Interaction	+ 56
Graphics	+ 49
Training	− 130
Consultancy	− 171

Other Questions inviting Comments

Much work remains to be done in analysing these. However, one outstanding feature of the comments received is general dissatisfaction with the documentation. In a sub-sample of 40 replies, 30 either criticised the documentation or suggested that it could be improved. This tallies with the very high score which documentation received above.

Case study – a Fortran influence

J Bryan-Jones
Department of Science
Cambridgeshire College of Arts and Technology
East Road
Cambridge
United Kingdom CB1 1PT

A recent request for advice reminded me of the sort of code which can result when programs are designed with Fortran in mind and then implemented in Genstat.

Example 1

The code below is an extract from a student's Genstat program which contained a macro to print one of five titles; the macro is called from within a 'FOR' 'REPEAT' loop. The design of the macro, with its use of 'ASSIGN' is reminiscent of a computed GOTO statement in Fortran.

```
'for' INDEX=1...5
  'use' HEAD $
  '' followed by various calculations & printing''
  'repeat'
```

with the macro defined as:

```
 'macro'     HEAD $
   'local'   L1,L2,L3,L4,L5,ENDLAB
   'assign'  L=L1,L2,L3,L4,L5 $ INDEX
   'jump'    L
   'label' L1 'caption' ''                  STAPLE''
              'caption' ''                  ------''
              'jump'  ENDLAB
''etc''
   'label' L5 'caption' ''                  ALL ISLANDS TOGETHER''
              'caption' ''                  --------------------''
   'label'   ENDLAB
'endmacro'
```

An alternative method, which I think is more suited to Genstat, employs a separate macro for each title:

```
'for' HEAD=HEAD(1...5)
  'use/r' HEAD $
  '' followed by various calculations & printing''
  'repeat'
```

with the macros defined something like:

```
'macro'     HEAD(1) $
  'local'   HEAD,DASH
    'heading' HEAD=''STAPLE''
              :DASH=''------''
    'print/lhm=20,squa=y' HEAD,DASH
'endmacro'
```

(The 'squa=y' removes a blank line which would otherwise be inserted between the heading and its underlining.)

There are several variations on this theme. For isntance, the centering of the heading can be handled by spaces in a format statement – which should present no problem to the Fortran programmer – the associated Genstat print statement is

```
'print/form=c' HEAD,DASH $ 20x,1,/
```

Example 2

Another example illustrates the tendency to employ the function ELEM. Consider how to shift the values of a vector K+1 places to the right, inserting 0, say, in the first K places. A typical solution employed by someone used to accessing individual array elements is

```
'for'  I=1...K    'calc' elem(Y;K)=0  'repeat'
'for'  I=K...N    'calc' elem(Y;K)=elem(V;I-K+1)   'repeat'
'calc' V=Y
```

A solution employing 'equate' is much simpler:

```
'equate' Y=(0)K,V
'calc'   V=Y
```

Does it matter?

It can be argued that the choice of directives and the appearance of the Genstat code does not matter at all, as long as the final program works correctly. I agree that this correctness is essential, however I think the ease and elegance with which the results are achieved is also of importance. The programmer has to make his program work – this involves debugging and testing the program, perhaps working with it for several days if not weeks, and simple code is easier to work with.

As computing develops we can expect people to use special purpose languages like Genstat for the kind of tasks which previously would have been handled in general purpose languages such as Pascal, Fortran or BASIC. However, this cannot be productive if the programmer is still thinking in another language be it Fortran, BASIC or Pascal!

Acknowledgement

To the keen student who provided the code that prompted this article.

Fourth Genstat Conference

The remaining papers from the Fourth Genstat Conference for which manuscripts have been received appear below. All four papers have, to some extent, been revised since the conference.

Modelling the Feeding Pattern of Rabbits with Cox's Regression Model

E de Turckheim-Lesquoy
O Pons
Biométrie
Institut National de la Recherche Agronomique
F78350 Jouy-en-Josas
France

Cox's Periodic Regression Model

The data consist of the feeding times and weights of food intakes of rabbits during weeks 6, 9, 12, 15 and 18 of life. The counting process N which counts the feeding times $T_1, T_2, \ldots$ observed between time zero and time t has been studied by Jolivet and others (1983) using a Poisson process with a periodic (24-hour period) density: this means that the number of food intakes between the time t_1 and time t_2, $N(t_2) - N(t_1)$ has a Poisson distribution with parameter $\mu = \int_{t_1}^{t_2} h(t)dt$, where h is the periodic density of the process. This also means that the probability of observing a point in the interval $(t,t+dt)$ conditionally on the past does not depend on that past:

$$P\{N(t+dt) - N(t) \neq 0 \mid past\} \simeq h(t)dt$$

If we replace $h(t)$ by a random quantity $\lambda(t{:}\omega)$, we use the stochastic intensity of the point process:

$$\lambda(t^+;\omega) = \lim dt \downarrow 0 \; P\{N(t+dt)-N(t) = 1 \mid \mathbf{F}_t\}$$

where $\mathbf{FF} = (\mathbf{F}_t)_{t\geq 0}$ is an increasing sequence of sub σ-algebras representing the relevant history.

Cox's model requires that the intensity should have the form:

$$\lambda(t) = h(t)\, exp\beta Z(t)$$

where $\beta Z(t)$ stands for the regression terms $\beta_1 Z_1(t) + \ldots + \beta_q Z_q(t)$, h and $Z_1,\ldots,Z_q$ are predictable (i.e. known before t) processes, $\beta_1 \ldots \beta_q$ are the regression coefficients to be estimated and $h(t)$ is a periodic unspecified non-negative function.

In our case we chose the two following groups of regressors Z:

(i) $x_1,\ldots,x_6$ are the lengths of the intervals between t and the previous feeding times: if $T_i \leq t < T_{i+1}$,

$$x_1(t) = t - T_i$$
$$\vdots$$
$$x_6(t) = t - T_{i-5}$$

and $q_1,\ldots,q_6$ are the weights of food intakes at time $t-x_1,\ldots,t-x_6$. These processes define models of type *Mx*, *Mq*, and *Mxq*.

(ii) $N_1,\ldots,N_8$ are the number of observed points in the intervals $[t-30',t),[t-60',t-30'),\ldots,[t-240',t-180')$ and $Q_1,\ldots,Q_8$ are the total weights eaten in the same intervals. These define *MN*, *MQ* and *MNQ* models.

As the mean of the intervals between two successive points is 46 minutes, $x_1,...,x_6$ and $q_1,...,q_6$ summarize on average about 4 hours of the past simply as $N_1,...,N_8$, $Q_1,...,Q_8$. Comparing the coefficients of these regressors to zero will tell us whether the past feeding times have an effect on the distribution of the following ones (models *Mx* and *MN* compared to the minimal model *Mm* with no regressors i.e. a Poisson model). It will also tell us whether the quantities eaten are relevant in explaining the future of the process *N* (comparing *Mxq* models to *Mx* models and *MNQ* to *MN*).

Estimating the βs

Like Cox (1972) we propose to estimate β minimizing the following

$$W(\beta) = \prod_{T_i} \frac{exp\{\beta Z(T_i)\}}{\sum_k \exp\{\beta Z(T_i + kp)\}}$$

where the product is taken over all observed points T_i during the interval of observation (t_0,t_1); the summation is over the integers k such that $T_i + kp$ lies in (t_0,t_1) and p is the length of the period. Such estimates have classical properties when $n = (t_1 - t_0)/p$ tends to infinity: consistency of $\hat{\beta}$ and weak convergence of $N^{\frac{1}{2}}(\hat{\beta}-\beta)$ to a guassian distribution under reasonable assumptions (Pons and de Turckheim, (1985)). Following Johansen (1983) these estimators can also be considered as maximum likelihood estimators in an extended model where the measure H, $H(t) = \int_0^t h(s)ds$, is replaced by a general measure which has a continuous and a discrete part. The likelihood to maximize is then

$$V(\beta) = \prod_{T_i} \Lambda[T_i] \exp\{-\Lambda[T_i]\} \prod_{U_j \neq T_i} \exp\{-\Lambda[U_j]\}$$

where $\Lambda[t] = H[t] \exp\{\beta Z(t)\}$, the points U_j are the periodic translates over (t_0,t_1) of the T_i, that is $U_j = T_i + kp$ for some i and k. $V(\beta)$ is then easy to maximize when it is considered as the likelihood of a sample of Poisson variables P_j where P_j is associated with U_j, takes the value 1 if there exists a T_i such that $U_j = T_i$ and the value 0 if $U_j = T_i + kp$ for $k \neq 0$. P_j has expectation $\Lambda_j = H[U_j] \exp\{\beta Z(U_j)\}$ and thus

$$\log \lambda_j = \beta Z(U_j) + \log H[U_j]$$

Then, if one knows, for each U_j, the value of $Z(U_j)$, the value of P_j and the number i of the T_i of which U_j is a translate (that is the value of a factor `TIME`), minimizing V is straightforward with regression commands:

```
'vari' Z(1...q) $ nn
'vari' P $ nn                    (to be calculated)
'fact' TIME $ n1. nn
'regre' P + Z(1...q) + TIME
'Y/error=poisson' P
'fit' TIME                       (fits to Poisson model)
'add' Z(1)
'add' Z(2)
    .
    .
    .
```

Below, we give a program to calculate Z, P and TIME for a series of observations starting at 7am on the first day and finishing at 7am on the eighth day : the estimation period (t_1,t_2) is shorter, since we do not know the value of Z during the first 4 hours (at least) of the experiment. The regression terms $Z_1 \ldots Z_q$ are those defined in the first section of this article, called x(1...nx), q(1...nx), N(1...nQ) and Q(1...nQ).

Genstat program to define the Poisson Sample and the Covariates

```
'refe/nid=300,nunn=400'calculZ
''
INPUT : TIMES t AND QUANTITIES q

''
'file' uf12 =12
'fetch/list=all'uf12
'get'thxq$t,q
'run'
''
NOW DEFINE :
- NUMBER OF PARAMETERS OF MODELS Mxq AND MNQ          :   nx , nQ
- LENGTH OF INTERVALS DEFINING N(1...) AND Q(1...)    :   lintQ
- SIZE OF ONE PERIOD                                  :   lper
- NUMBER OF OBSERVED PERIODS , POSSIBLY INCOMPLETE    :   nper
THE EXPERIMENT STARTS AT TIME elem(t;1) AND ENDS AT TIME nper*lper
''
'scal' nx=2 : nQ=2 : nper=7 :lintQ=30 : lper=1440
''
''
'scal' n1,nnx,nnQ
'calc' n1=nval(t)
'calc' nnx=nx+1 :nnQ=nQ + (nQ.eq.0)
'run'
''
DEFINITION OF THE VARIATE U
''
'scal'nn,nuu,nh,nstart,nstart1,t1,tnx,tstart
'calc' t1,tnx=elem(t;1,nnx)
: tstart=t1+nQ*lintQ
: tstart=tstart*(tstart.gt.tnx)+tnx*(tstart.le.tnx)
'scal' tt(1...n1)
'equa' tt(1...n1)=t
'valu'nstart=1
'for' ttt=tt(1...n1)
'goto' ok*(ttt.gt.tstart)
'calc' nstart=nstart+1
'repeat'
'label' ok
'deva' tt(1...n1)
'run'
```

```
'print' t1,tnx,tstart, nstart $ 3(8.2),4
'calc' nstart1=nstart-1
: nh=n1-nstart1 : nuu=nh*nper
'run'
'vari' h$nh
'equa' h= t$nstart1!x,nh
'calc' h=h-intpt(h/lper)*lper
'vari' uu$nuu
: h(1...nper) $h
'calc' h(1...nper)=h+ ((1,2...nper)-1)*lper
'equa' uu = h(1...nper)
'calc' uu=order(uu)
'deva' h(1...nper)
'fact' ttime $ nh,nuu=(1...nh)nper
''
SUPPRESSING TIES
''
'vari' uu1$uu
'equa' uu1=0,uu
'calc' uu1=uu-uu1
: uu1=(uu1.eq.0)
'fact' ties $2,uu
'group' ties=intpt(uu1)
'rest' uu,ttime $ ties=1
'calc' nn=nval(uu)
'run'
'unit' $nn
'equa' u=uu
'fact' time $h,nn
'equa' time=ttime
'deva' uu,uu1,ttime
''
COMPUTATION OF VARIATES P, N(1...nQ),Q(1...nQ),x(1...nx),q(1...nx)
''
'scal' dt(1...nnQ),tcrt,qcrt,tcrt(1...nx),qcrt(1...nx),i(1...nx)
'calc' dt(1...nnQ)=lintQ*(1 ...nnQ)
'equa' N(1...NNQ),Q(1...NNQ)=0
'equa' x(1...nnx),q(1...nnx)=0
'vari' P=nn!(0)
'for' i=nstart1,nstart...n1
'calc' tcrt=elem(t;i)
: qcrt=elem(q;i)
'calc' Pcrt=(u.eq.tcrt)
: P=P + Pcrt
'goto' suite2*(nQ.eq.0)
'calc' logic(1...nnQ)=
       ((u-tcrt).le.dt(1...nnQ))*((u-tcrt).gt.(dt(1...nnQ)-dt(1)))
: Q(1...nnQ)=Q(1...nnQ) + logic(1...nnQ)*qcrt
: N(1...nnQ)=N(1...nnQ) + logic(1...nnQ)
```

```
'label' suite2
'goto' suite3*(nx.eq.0)
'equa' x(1...nx),q(1...nx)=0
'calc' i(1...nx)=(i+1)-(1,2...nx)
'calc' tcrt(1...nx)=elem(t;i(1...nx))
'calc' qcrt(1...nx)=elem(q;i(1...nx))
'calc' logic=tcrt(1).lt.u
: x(1...nx)=x(1...nx)*(1-logic) + (u-tcrt(1...nx))*logic
: q(1...nx)=q(1...nx)*(1-logic) + qcrt(1...nx)*logic
'label' suite3
'repeat'
'deva'Pcrt,qcrt,tcrt,logic(1...nnQ),logic
'run'

''SAVING THE VARIATES''

'goto' suite4*(nx.ne.0)
'put'ZCOX$P,u,N(1...nQ),Q(1...nQ),lP,lper,nper,inth,time
'goto'suite6
'label' suite4
'goto' suite5*(nQ.ne.0)
'put'ZCOX$P,u,x(1...nx),q(1...nx),lp,lper,nper,inth,time
'goto' suite6
'label' suite5
'put'ZCOX$P,u,N(1...nQ),Q(1...nQ),x(1...nx),q(1...nx),lP,lper,nper,inth,time
'label' suite6
'file' uf13=13
'save'uf13$ZCOX
'disp/list=all'uf13
'run'
'close'
'stop'
```

Interpretation of the Genstat Output

Most of the Genstat output can be used in a sensible way (Pons and de Turckheim, (1986)):

(i) the deviance is a goodness-of-fit criterion in the class of all models whose intensity has the form: $\lambda(t) = h(t)\, F(Z(t))$ for a given choice of Z and any function F. We have $DEV(M) = -2log\, W(\hat{\beta})$ on a chosen model M and $DEV = 0$ when F is non-parametric. DEV is maximum for the Poisson model Mm: it is thus convenient to calculate $dD(M) = DEV(Mm) -)DEV(M)$: $dD(M)$ represents the gain of goodness-of-fit when adding regressors to the Poisson model. It is asymptotically distributed as a χ^2 variable and the tests for nested models are easy to use.

(ii) standard-errors calculated by Genstat are consistent estimators and the Wald's statistic for each β (named T in Genstat output) is asymptotically distributed as a normal $N(0,1)$ variable.

(iii) the residuals are also useful: for each P_j the expected value $\hat{\lambda}_j$ is

$$\hat{\lambda}_j = \frac{\exp\{\hat{\beta}Z(U_j)\}}{\sum_k \exp\{\hat{\beta}Z(U_j+kp)\}} = \hat{\Lambda}[U_j]$$

Considering the n points U_j which are periodic translates of the same T_i, the sum of the corresponding $\hat{\lambda}_j$ is 1: these n points are correctly fitted if $exp\{\hat{\beta}Z(T_i)\}$ is large and $exp\{\hat{\beta}Z(T_i+kp)\}$ is small for $k \neq 0$; that is, if $\hat{\Lambda}[T_i]$ is near 1. So instead of considering each residual d_j^2 given by Genstat

$$d_j^2 = -2(log\hat{\lambda}_j + 1 - \hat{\lambda}_j) \quad \text{if } Y_j = 1$$

$$d_j^2 = 2\hat{\lambda}_j \quad \text{if } Y_j = 0$$

it is convenient to consider the cumulated residuals

$$D_i = \sum_k d_j^2 = -2log\,\hat{\Lambda}[T_i]$$

and to compare them to the corresponding ones in the Poisson model Mm, where they are equal to $-2log\frac{1}{n}$ when $(t_0,t_1) = (0,np)$. We, of course, still have $DEV = \sum_i D_i$.

A graph of these residuals can therefore be used and might give intelligible information. Possibly, for some bad D_i, the n values d_j^2 could also be displayed for detailed examination.

Conclusion

Cox's regression model turned out to be quite useful in studying periodic processes. In the rabbit experiment, it gave satisfactory results: for example, a significant effect of the last food intake through its weight and the time elapsed since. Also, cumulated quantities Q were much more informative then the cumulated numbers of intakes N and the slowing effect of large quantities of food eaten lasts for about one hour for any of the 7 adults (18 week old rabbits).

However, the proposed method of estimating β is not recommended when the number n_1 of observed points is too large: in the example it was reasonable for most of the rabbits, say for $n_1 \leq 250$, but it took excessive computing time to fit the data to the different models for rabbit number 8, for which $n_1 = 397$.

In the case of large data sets, a parametric version of Cox's periodic model should be used. In these models, the function h is piecewise constant on R intervals of $(0,p]$. When $Z_1,...,Z_q$ are jump processes, Genstat may still be used to fit the data and a test is available to choose R, the number of values of h (Pons and de Turckheim, (1986b)).

References

[1] Cox, D.R.
Regression models and life tables.
J.R.S.S.(B), **34**, 187-220, 1972.

[2] Johansen, S.
An extension of Cox's regression model.
Internat. Statist. Review, **51**, 165-174, 1983.

[3] Jolivet, E., Reyne, Y. and Teyssier, J.
Approche méthodologique de la répartition nycthémérale des prises d'aliment chez le lapin.
Reproduction, Nutrition, Développement, **23**, 13-24, 1983.

[4] Pons, O. and de Turckheim, E.
Cox's periodic regression model.
Prépublication Université Paris-Sud Mathématique n° 85T36, 1985.

[5] Pons, O. and de Turckheim, E.
Modele de Régression de COX Périodique et Etude d'un Comportement Alimentaire.
Cahiers de Biométrie n° 1 INRA-Versailles (to appear), 1986a.

[6] Pons, O. and de Turckheim, E. Modelling a feeding pattern with point process: Cox's periodic regression model.
Submitted to Biometrics, 1986b.

Genstat 4.03E

J Coursol
Equipe de Recherche associée au C.N.R.S. 532
Statistique Appliquée
Mathématique
Bâtiment 425
Université de Paris-Sud
91405 Orsay cedex
France

Introduction

Genstat 4.03E is a version of Genstat designed for use on microcomputers based on the Intel 8086/8 series of processors, such as the IBM PC, Victor (Sirius), Olivetti M24 and Apricot, running the PC-DOS or MS-DOS operating system.

The program has been overlaid and is supplied in two versions, which vary in their degree of overlaying. Each has a maximum internal storage space of 8000 data items.

The minimum system requirements to run Genstat 4.03E are 210 Kbytes of free memory and either a floppy disc drive with a capacity of at least 600 Kbytes or a hard disc. 320 Kbytes of free memory allow the faster version of the program to be used and performance is greatly improved by the use of an 8087 mathematical coprocessor. Use of a hard disc greatly improves the convenience of use and should be the norm.

This version of Genstat is based on release 4.03, with changes in some directives (`INPUT`, `OUTPUT`, `FILE` and `GRAPH`) and incorporating some features of 4.04 (the directives `R`, `HELP` and `FOURIER` and error messages). A `DOS` directive returns control temporarily to the DOS level so that MS-DOS commands or other programs can be executed. The Genstat `CALL` directive, which may be used to add new functions written in Fortran to Genstat, is formally documented in this version.

Modified Directives

INPUT, OUTPUT and FILE

The INPUT, OUTPUT and FILE directives perform the usual Genstat operations, as described in the Genstat 4.03 manual, and also the secondary channel allocations to the files. For this, it is necessary to declare a heading structure containing the name of the required file. Using the option FILE=*heading* with any of the three directives will close the file previously allocated to the relevent channel, open the file named in the heading structure and allocate that file to the channel.

The first INPUT and OUTPUT directives, for any other than the primary channels, and the first FILE will necessarily contain the FILE option (otherwise fault FI 5 is reported). Thereafter, the FILE option will be used only when it is necessary to reallocate the channel to another file.

Note that the FILE option has no effect for primary input or output channels since these allocations are made definitively at the start of a job.

The syntax of the three directives is:

'INPUT [/FILE = *heading*]' *scalar*

- *heading* describes an existing file, otherwise an error is reported.
- *scalar* (= 1, 2 or 3) refers to the input channel.

'OUTPUT [/FILE = *heading*]' *scalar1* [$ *scalar2*]

- *heading* describes a file to be created or overwritten.
- *scalar1* (= 1, 2 or 3) refers to the output channel.
- *scalar2* gives the number of columns on output.

'FILE [/FILE = *heading* [,MODE = *name*]]' *user-file* = *scalar*

- *heading* describes an existing file, when *name* is O (for old), or a file to be created, when *name* is N (for new). The default for *name* is O.
- *scalar* (= 9 or 10) refers to the logical channel number.
- *user-file* is a user-file identifier.

Examples

In the following example, a subfile, SF, which has been created in the workfile, is copied to the external file B:KEEP, which has been assigned to unit 9, with the internal file-name UF1.

```
      .
      .
'HEADING'                G1=''B:KEEP''
'FILE/FILE=G1,MODE=N' UF1=9
'PUT'            UF1,SF
      .
      .
```

In a second Genstat program, the subfile, SF, is retrieved from B:KEEP, and put into the workfile.

```
      .
      .
'HEADING'                G2=''B:KEEP''
'FILE/FILE=G2'   UF2=9
```

```
'FETCH'          UF2 $ SF
   .
   .
```

In the next example, the file B:INDAT contains two records:

```
1 2 3 4 5  'EOD'
6 7 8 9 10 'EOD'

'REFERENCE' EXAMPLE2
'VARIATE'        X,Y,Z $ 5
'HEADING'        FF=''B:INDAT''
'INPUT/FILE=FF' 2                ''allocate B:INDAT and change the input
                                   stream''
'READ'           X
'INPUT'          1               ''return to the primary input stream''
'RUN'
'INPUT'          2               ''change the input stream''
'READ'           Y
'INPUT'          1               ''return to the primary input stream''
'RUN'
'INPUT/FILE=FF' 2                ''allocate B:INDAT and change the input
                                   stream''
'READ'           Z               ''Z and X have the same values''
'PRINT/P'        X,Y,Z $ 8
'INPUT'          1               ''return to the primary input stream''
'RUN'

   X        Y        Z
   1        6        1
   2        7        2
   3        8        3
   4        9        4
   5       10        5

'CLOSE'
'STOP'

GRAPH
```

The additional option CM = 1 of the GRAPH directive will give a high-resolution plot (assuming that a graphics screen is available).

If the output stream is the primary channel (allocated to the screen), the plot is displayed on the screen. Hardcopy can be produced on graphics printers which are IBM compatible and have a GRAPHICS interface loaded.

If the output channel is allocated to a file, the graph is stored in vectorial mode (similar to Tektronics) and may be displayed (using the supplied PRINTG.EXE program) by entering the command:

PRINTG *fileout*

where *fileout* is the output file created by Genstat. PRINTG interprets the control characters for form feeds etc. and displays the high-resolution graph on the screen.

Genstat 4.04 Features

R, HELP and FOURIER

The R, HELP and FOURIER directives and extended error messages are implemented as in Genstat 4.04 and are described below.

R is a synonyn for RUN and may be used in exactly the same way. It is particularly useful in interactive running.

HELP provides on-line information about Genstat. The syntax is:

'HELP' [*names*]

where names are directive names or indices. Omitting names gives a basic index.

FOURIER is used in time-series analysis for calculating cosine and Fourier transforms of real or complex series. The syntax is described in the Genstat 4.04 Manual.

Error Messages

As in Genstat 4.04, extended error messages are printed when faults are detected. The extended message is printed immediatly after the formal message.

Other Features

DOS

The command sequence 'DOS' 'R' causes the message:

*** COMMAND LEVEL (CR comes back to GENSTAT) ***

to appear, followed by the prompt >*x*>, where *x* is the current volume.

The user is returned to the MS-DOS environment and may execute MS-DOS commands or programs (whose maximum size will depend of the available memory). Sending a void line (by entering a carriage return) returns the user to Genstat, with all structures preserved.

CALL

This directive is for use by experienced MS-FORTRAN programmers who wish to interface external modules directly with Genstat. Its syntax is

'CALL' *routine* $ *a1*; *a2*; ...

where routine is the name of the external module and *a1*; *a2*; ... are Genstat structures with assigned values.

Note that the external module does not generally know the dimensions of *a1*, *a2*, etc. (unless these are explicitly defined within it), so it will be necessary to transfer these dimensions in scalar structures.

Example

```
   .
   .
'VARIATE'       V
   .
   .
'SCALAR'        NV
'CALCULATE'     NV=NVAL(V)
'CALL'          TOTO $ NV;V
   .
   .
```

The external module is a Fortran subroutine with name MAIN and arguments corresponding to the structures *a1*, *a2*, etc. of the CALL directive.

The Fortran data type equivalents of the structure modes described in Part II, section 4.1.2 of the Genstat 4.03 manual are listed below:

Structure Mode	Data Type
real	REAL*4
integer	INTEGER*4
name	REAL*8 (Hollerith data)
identifier	INTEGER*2
text	INTEGER*2 (Hollerith data)

The subroutine must be compiled with the MS-FORTRAN compiler release 3.13 or higher.

Example

```
      SUBROUTINE MAIN(RX,X)
C
C LISTING OF RX VALUES OF X WHERE THE STRUCTURE OF X IS ONE OF
C SCALAR, VARIATE, MATRIX, SYMMAT, DIAGMAT, TABLE OR DSSP
C
      REAL*4 RX,X(1)
      INTEGER IX,I
C
      IX=RX
      DO 1 I=1,IX
    1 WRITE(*,*) X(I)
      RETURN
      END
```

The ASM source given below (and also in file NBPARAM.ASM) must be altered as appropriate and compiled using the MASM Microsoft compiler.

```
NAME NBPARAM
DATA segment public 'DATA'
public NB_PARAM
NB_PARAM DW 2
DATA ends
END
```

The 2 in the fourth line is the number of structures passed by `'CALL'` in this example.

The two resulting object modules, together with the ENTXGG.OBJ module, should then be linked using the MS-LINK linker.

The available memory size must be greater than

170 Kbytes + routine size + routine heap size + routine stack size

(the heap size depends on the number of files opened by the routine module; allow between 2 and 10 K-bytes approximately for heap + stack).

The routine module must have extension `GGG` and must be present in the Genstat volume and directory.

The Genstat Macro Library

J Bryan-Jones
Department of Science
Cambridgeshire College of Arts and Technology
East Road
Cambridge
United Kingdom *CB1 1PT*

Abstract

The following topics are considered: why there should be a Macro Library for Genstat, a strategy for writing re-usable software in Genstat and quality control for the library.

Introduction

An obvious place to start is by answering two questions 'What is The Genstat Macro Library?' and 'Why do we need it?'

The Genstat 4 Macro Library (G4ML) is a collection of procedures or part-programs in the Genstat language, which is special in several ways. First, it can be used by anyone who has access to Genstat. Second, there is good documentation in the form of a user-manual, so that it is easy to find out what techniques are available and how to use them. Third, expert contributions to the library come from many sources and the programming of the macros has been checked and tested. Finally, the library is easy to obtain from NAG, or rather, it soon will be!

We need the G4ML because, although Genstat offers a lot, it does not offer everything: for example it offers `'ANOVA'` but not `'MANOVA'`. A library is a convenient way of providing pre-tested programs for some analytical techniques, so that we can save time in designing, writing and testing programs. This is because we do not need to know the full statistical or algorithmic details in order to use the technique; nor do we need to learn how to implement the algorithm using the Genstat language. Also, there is a high probability that the answers are correct. In short, macros make Genstat extensible.

Examples from the Genstat 4 Macro Library

As an example of a statistical technique whose details we may not know, I have chosen to consider the calculation of an interval estimate for the ratio of two estimated parameters. The variance of such a ratio cannot be evaluated exactly but we can get an approximation using Fieller's theorem (Fieller, 1940, 1944). This procedure is particularly relevant to bioassay (Finney, 1971) and calibration (Davis and Goldsmith, 1972).

For example, in a quantal bioassay to estimate the dose that gives 50% effectiveness, $ED50$, the classical procedure is to use weighted regression to fit a straight line where the dependent variable is the probit of the response and the explanatory variable is $ln(dose)$. In Genstat we can get the weighting automatically by using the regression directives with a binomial error; this also takes care of the probit transformation for us (and indeed offers the alternative logit transformation). If the equation is $E(y|x)=mx+c$, where y is the proportion responding and x is the $ln(dose)$ then the regression or glm analysis provides estimates of m and c and their covariance matrix. The point estimate of $ln(ED50)$ is the value of x when $y=0.5$ and is given by

$$\hat{x} = 0.50/\hat{m} - \hat{c}/\hat{m}$$

which is sometimes called 'back-prediction'. To obtain an interval estimate for $ln(ED50)$ we must first estimate the variance of $\hat{x}$; which involves the variance of $\hat{c}/\hat{m}$. This cannot be evaluated exactly, because the denominator is an estimate with a standard error. The library macro FIELLER will provide an interval estimate if it is given the results of the regression analysis. Alternatively, the library macro PROBCOMP will fit the regression (using a probit transformation) as well as providing the interval estimate; indeed with PROBCOMP it is very straightforward to do a standard probit analysis.

By using one of these library macros I can gain in three ways. First, I do not actually need to know the formula. Second, I will almost certainly save time, even if I have not used the library before. Third, there is a higher probability that the answers are correct.

As a second example I shall consider the display of three dimensional data, to illustrate a technique for which an algorithm may not be known or the Genstat implementation of the algorithm may seem difficult. Most Genstat users are familiar with the 'GRAPH' directive which offers good plotting facilities in 2 dimensions, including the possibility of using different symbols to make the display more useful. This is fine for 3 variables of which one is essentially a grouping factor. However, if the 3 variables are quantitative then the library macro D3PLOT may prove useful – it produces a perspective plot in two dimensions.

The gains from using the library for this graphical display are threefold. I do not need to think about the algorithmic details, nor how to implement the algorithm in Genstat, so I will almost certainly save time. Also, there is a higher probability that the graph is as accurate as the printer or screen allows.

The thing that these two examples have in common is that each requires some skill or knowledge which I may not have at my finger-tips. The use of library software can simplify the work that I have to do in order to progress with the data-analysis; it can also make it easy for me to try different methods of analysis for the data.

Macros for Standard or Complex Techniques

A different way of classifying Library software is to consider the complexity of the technique. A macro may simplify a standard technique or it may provide a new and probably complex technique.

Essentially, the 'standard-technique macro' saves time. Its justification is that it is used a lot and is useful to many people. Since Genstat is a high-level statistical language, some standard statistical techniques are already provided by its directives, but some are not. For example, I have already mentioned that there is no directive 'MANOVA'; instead, we need to 'USE' MANOVA $ (after setting up the parameters).

There are at least two reasons for having standard technique macros in the library. First, macros that simplify everyday or routine tasks could be particularly useful for inexperienced users of Genstat. Second, simple tools presented as macros are potentially useful in teaching, especially in service courses, where we may want to separate the task of data analysis from the task

of learning to program. An obvious example that might appeal to ex-MINITAB users is to provide the equivalent of NRANDOM, BRANDOM, PRANDOM etc., which provide pseudo-random samples of specified size from particular distributions.

The other type of macro is the 'complex technique macro'. Often, the technique is new and the Genstat macro facility provides a means of making it quickly available to the statistical public. The justification is the dissemination of research and development. It is probably fair to assume that, at least initially, these techniques will be used by experienced statisticians, who will either be familiar with Genstat or will have good support services. Also, we should recognise that the technique may not gain general credence, and may even disappear.

It seems to me that most of the current macros are of the second type, and are aimed at the experienced Genstat user who wants to extend his repertoire of analyses. Perhaps there is more motivation and satisfaction for authors when the macro applies some recent research. Nevertheless I see a role for macros of both types.

In summary: using the library lets us use techniques without knowing them in detail and without having to spend time programming them. So the library helps to control the complexity of our work.

Writing Re-usable Software

I now want to consider what is involved in writing software for general release. The phrase may sound rather grand and professional, but we should recognise that a library macro is not merely 'just another little bit of Genstat code'; it is published as software for general release to the international statistical computing community and must be 're-usable' by all.

Clearly, a software library must attain an acceptable professional standard. Let us consider what this standard is and how it may be achieved. There are certain attributes that good statistical software should have. It should be correct (numerically and statistically), not inefficient (though this is inevitably a compromise), reliable (with safeguards for incorrect or inappropriate input), easy to use (concise but straightforward interface), flexible (works for different sizes and shapes of data) and easy to maintain (comprehensible code).

Before considering how such a standard is achieved, let us see what other standards there are. In the last few years I have met many Genstat users who have written macros for their own use; often their comment is 'Well, it worked for my job but to make it good enough for the library is far too much work'. Such comments imply that, in some sense, users do not ordinarily write 'good software'. We can assume that the user achieves correctness and, to some extent, efficiency and reliability; however, he may not attempt to achieve flexibility or ease of use and maintenance.

Before continuing, a few words about terminology – I use the term 'one-off' for the user's macro and refer to the potential library macro as a 'product'. Each 'one-off' program can be considered as an initial working draft of a potential final product. The difference in approach to the two types of software is well illustrated by an example.

Example of a one-off program

Many of you will be familiar with the macro RYL for producing graphs involving residuals and, in particular, the macro QQNORM which standardises the residuals

and finds the expectations of the corresponding normal order statistics. Suppose these macros were not available and that we were setting out to write their equivalent.

First, we need to be sure that we understand the term 'normal order statistic'. We can get a definition and a formula from one of many texts: the expected value of the rth order statistic in a sample of size n from a standard normal distribution is given by

$$\frac{r+a}{n+2a+1}$$

where a is constant. However, we might discover that there is some disagreement about the value of the constant a. Cox and Hinkley (1974) suggest $a=-0.375$ from Blom (1958); Harter (1961) recommends $a=-0.363$, while Alvey, Galwey & Lane (1984) use $a=-0.5$. These numbers do not seem too different, certainly two of them are quite close. If we are worried we can always try them all – as long as we set a as a variable it is easy to change its value. Perhaps we should read the literature more carefully to find out why they differ and what other values have been recommended. Note that this is the first point where our behaviour may differ, depending on whether we are writing software to be used once or as a re-usable product.

Returning to the formula, we know that Genstat can

```
'calculate' NSCORE = NED((R + A)/(nval(RESID) + 2*A + 1))
```

where the values in R are the integers 1 to N, N is the number of values in the variate of residuals, RESID, and A is a scalar. We may pause to wonder whether this will work if we have missing values, or if the variates are restricted. However, for our 'one-off' we can ensure that these difficulties do not arise. Note that this is the second point where our behaviour may differ depending on whether we are writing one-off software or a product.

If N is known, we can set the values of R by

```
'variate' R $ RESID = 1 ... N
```

or, if N is not known, we could use the function CUM and write:

```
'variate' R $ RESID   'calc'  R = cum(R=1)
```

The standardisation of residuals is easily done by:

```
'calc'  STDRES = (RESID-mean(RESID)) / sqrt(var(RESID))
```

Then we can simply order the values in STDRES and plot the required graph:

```
'calc'  STDRES = order(STDRES)
'graph' NSCORE ; STDRES
```

All that remains is to collect together the fragments and we have a workable one-off solution.

Development into a Product

Suppose that we now want to consider developing our code into a product. We must consider the scope of potential use and, further, it is also worth considering how the code may be improved.

We have already identified two possible problems – A may have the wrong value and

the residuals may have no values, missing values or may be restricted. Clearly, we want to consider possible modification of A and we want to be sure that the code does sensible things if the vector of residuals is incomplete. A third possible problem is that we may be introducing too many new variables – we need to weigh up clarity and space considerations.

First, let us consider the value of A. If we want to run the 'one-off' with a different A, we only need to change one line (which shows that we have already introduced some generality into the code). For a product we would use a standard technique to handle default parameters. We can use the function TYPE which returns an integer or the missing value indicator; if the result is not missing then we can use 'position' or a single expression to test its value. An example of this is given by Simpson (1981) who utilises the statement

```
'equate' CONTROL $ P,/ = PARAMS
```

to put into the variate CONTROL the values which the user has set into the variate PARAMS (P = nval(PARAMS)). Any missing values in CONTROL are then replaced by the DEFAULT values that the macro writer has incorporated by

```
'calc' CONTROL = repmv(DEFAULT)
```

Second, let us consider the problem of missing or restricted residuals. The algorithm we use and its implementation must be robust. Clearly we will need to make use of the functions nval and nmv. An elegant revision of QQNORM which handles missing values utilises the fact that the function ORDER puts missing units at the beginning (Sackville-Hamilton, 1984). The code is

```
'calc'  R = cum(R=1) - nmv(STDRES,X)
 :    X,R = order(R;STDRES)
 :      R = R + 0/(R.gt.0)
 :      R = ned((R+A)/(nval(R) - nmv(R) + 2*A + 1))
```

Third, let us consider whether the product should create new data structures or overwrite the values in the users' data structures. If the role of the library is to provide techniques then it is reasonable to hide the code from users and concentrate on efficiency. If, on the other hand, the role of the library is to give ideas to the user, then he should be able to understand the code and the writer should concentrate on its clarity. It is perhaps appropriate to note that the revision of QQNORM has reduced NID from 5 to 1 and NUNN from 12 to 4.

The one-off solution ordered the values in STDRES. This is not the same as the macro QQNORM which retains the ordering in STDRES and instead re-orders the new variate NSCORE. It may not be immediately obvious how to do this, if you are not familiar with the technique. It can be done using ORDER twice, or ORDER and POSITION. However, it is not clear that the macro needs this complication – perhaps it should be an optional extra.

This brings us to another important point about software – defining what it is going to do. Clearly, this will be fairly well defined for 'one-off' software. However, for software products, we have to define a set of problems which it is useful to solve. Indeed, the product does not necessarily have to solve the original problem which suggested it. It must cater for a wider class of problems which other people may want to solve – and this level of design is not easy.

It is relevant here to add something else about products. Once the problem has been defined, we know the purpose of the product. The next extremely important feature of 'product' design is the partitioning of the problem. Consider, for example, the library macro QQNORM, This does three things – standardises residuals, calculates expected normal order statistics and re-orders the expected values; it could be argued that these functions should be performed by three separate macros. Consider, also, bioassay analysis for which there are two macros, written by different authors. One author has chosen to tackle only a small aspect of the analysis. The second has chosen to provide the whole analysis in a single macro (which does not call the other author's macro).

By considering how to do a normal plot for residuals we have seen that some design and programming decisions may be different in the one-off and the product situations. This raises an interesting question which could be debated at some length: 'How much software is truly "one-off"?'. With the current emphasis on modular design and structured programming we should always think of partitioning our current problem in a way that will provide code which we can re-use. However, sometimes a lot of work is needed to ensure that a software product is general, safe, and easy to use and maintain. Brooks (1975) argues that the cost of a programming product is at least 3 times the cost of a one-off program; this probably gives a useful guide for preparing macros for the Macro Library.

In summary: macros for the library are software products, and they must be good; this can mean more costly development but with savings for the Genstat community at large. Care is required in defining what a suite of macros is going to do, in partitioning this among separate macros and in making design and programming decisions which ensure that the software is both useful and re-usable. Perhaps we should always plan software with re-usable macros in mind.

Quality Control

The last topic I want to address is *refereeing* or *quality control.* First, some relevant facts:

1. It was agreed that each macro should be subjected to a refereeing process. Each submitted macro is sent to one or more independent referees who are asked to comment on the suitability of the technique and its implementation.
2. There are no funds to pay referees or even to reimburse expenses, so that we rely on the goodwill and interest of the more experienced users in the Genstat community.

Now, some conjectures:

1. Quality of assessment will greatly affect the library.
2. To referee a macro description is not a daunting task; however, assessing the macro code is rather different.
3. Refereeing is not identical to quality control.

Clearly, if the library has good referees who are prompt and constructive in their comments and criticisms, then we will be able to maintain a high standard in our library, with the result that authors will feel it is worthwhile to publish in the library and their macros will be used by a grateful and respectful public. This would be an ideal situation! However, if the refereeing process lets through some poor macros that are perhaps inadequately described, incorrectly programmed, tedious to use or employ

out-of-date methods, then users will suffer and be put off from using other, good macros. Also, if the refereeing process is slow then authors may lose their enthusiasm.

I have used two terms: *refereeing* and *quality control.* I suggest that a library of software needs quality control rather than refereeing. Usually, when we publish we are expounding some new knowledge; when we referee papers we are assessing the contribution of this knowledge. We do not usually get paid for refereeing but we do get an early opportunity to see another person's original ideas, and this can be of great interest. However, when we assess a library macro we are not assessing its contribution to the body of knowledge but, instead, we are checking that the quality is of the required standard: we need to be sure that it will pass all reasonable tests which could be devised. The ability to control quality is quite different from the abilities needed to assess knowledge. In analogy, we can think of buying a house and first commissioning a structural survey; we want to know that the house is sound and that we can expect to use it for a number of years – not that it is a building of intrinsic novelty or architectural interest!

In summary: macros submitted for the G4ML are independently refereed. The quality of refereeing will have a great affect on the success of the library, but refereeing code is a non-trivial task. Quality control is more appropriate to library software than refereeing.

Summary

There is a role for macros and a Macro Library. However, macros can only be useful if they are good, and there are problems in establishing a workable mechanism for ensuring that a macro is good.

References

[1] Alvey, N., Galwey, N. and Lane, P.
An introduction to Genstat.
Academic Press, 1982.

[2] Blom, G.
Statistical estimates and transformed beta-variables.
Wiley, 1958.

[3] Brooks, F.P.
The Mythical Man Month: essays on software engineering.
Addison-Wesley, 1975.

[4] Cox, D.R. and Hinkley, D.V.
Theoretical Statistics.
Chapman and Hall, 1974.

[5] Davis, O.L. and Goldsmith, P.L. (Eds)
Statistical Methods in Research and Production.
Longman Group, Ltd, for ICI, 1972.

[6] Fieller, E.C.
The biological standardisation of insulin.
J.Roy.Statist.Soc., Suppl 7, 1-64, 1940.

[7] Fieller, E.C.
A fundamental formula in the statistics of biological assay, and some applications.
Quarterly J. Pharm. & Pharm., **17**, 117-123, 1944.

[8] Finney, D.J.
Probit analysis.
Cambridge University Press, 1971.

[9] Harter, H.L.
Expected values of normal order statistics.
Biometrika, **48**, 151-165, 1961.

[10]Sackville-Hamilton, N.R.
Personal communication, 1984.

[11]Simpson, H.R.
Comment – points of general interest to macro writers.
Genstat Newsletter, **8**, 19-20, 1981.

Genstat and Workstations

K I Trinder
NAG Central Office
Mayfield House
256 Banbury Road
Oxford
United Kingdom *OX2 7DE*

Introduction

The use of Genstat on mainframe and mini computers is well established and it is anticipated that its use on computers of these types will continue to increase.

There is also, however, a growing usage of less powerful computers and consequently a growing demand for Genstat on such computers. Computer manufacturers are offering more power and facilites in single- or few-user systems, thus giving more freedom to individuals. The addition of networks provides a means of using the resources available on other computer systems. There is clearly a wide market open to Genstat which is, as yet, largely untouched.

The aim of this article is to examine one specific type of computer, the workstation, in relation to Genstat. In particular, there will be, first, a discussion of what is meant by the term 'workstation' and, second, a description of the problems encountered when implementing Genstat on one such workstation (the ICL Perq running PNX), with reference to how similar problems might be found in other workstation implementations.

What is a Workstation?

In order to discuss workstations, it is useful to get some idea of where they come in the whole spectrum of computer power and then to examine their principle characteristics.

At the top end of the spectrum there are super-computers (for example, the Cray), followed by mainframes (dominated by IBM) and mini-computers (notably the PDP and VAX ranges from DEC, particularly the VAX for Genstat use).

At the bottom end there are home computers (for example, those made by Commodore), PCs (again dominated by IBM) and workstations.

As mentioned above, Genstat is established on a considerable number of mainframe and mini computers and so it is appropriate to give more attention to the lower end of the spectrum.

There is little to offer owners of home computers (unless someone would like to write a Genstat game: perhaps an adventure based on the manual?)

There has certainly been considerable interest in providing Genstat on PCs although, strictly, this is outside the scope of the article. However, an MS-DOS version of Genstat is now available and details are given elsewhere in this Newsletter.

The next step up is the workstation, although it is not always clear what is actually meant by the term workstation. It would be nice to simply say that a workstation is smaller than a mini-computer and bigger than a PC and leave it there, but that would be very unsatisfying and not at all helpful. What is really required is a concise and formal definition of a workstation in a few sentences based on attributes which are either common to or special to all workstations. Unfortunately, this appears to be a most difficult thing to find; that is, it is almost impossible to give a staightforward definition which applies exclusively to workstations. It is therefore necessarily only possible to talk in terms of the general characteristics of workstations.

We can now identify the main attributes of workstations:

1) They are single- or few-user systems (perhaps four at most) with an emphasis on considerable computing power being available to individual users.
2) 16/32 or 32/32 bit processors (i.e. 16 or 32 bit data paths with 32 bit computations).
3) Good networking capabilities; i.e. the ability to link workstations to other workstations and to mainframes for file transfer, file sharing or terminal emulation.
4) Normally one megabyte of memory or more, with half a megabyte as a minimum.
5) A large internal disc; usually having between 30 and 70 megabytes capacity.
6) A floor standing (and usually noisy) processor and disc cabinet.
7) Frequent use in software development for scientific and engineering applications requiring Fortran, Pascal and C compilers.
8) Unix (or one of its many lookalikes), either as the only operating system or as an alternative to a proprietary operating system.
9) 'Convenience' software: for example, bit-mapped graphics, window management system, screen editor.
10) 'Convenience' hardware: for example, tablet and puck, mouse.
11) A current price in the approximate range £5,000 to £40,000.

Having decided on what we might expect to find in a workstation, it is informative to list the computers which are generally thought of as workstations:

ICL Perq
Apollo Domain
Sun
IBM RT PC
Whitechapel MG1
Hewlett Packard 9000
Acorn Cambridge
PCS Cadmus
Sage
Tadpole Leonardo
Orion
DEC MicroVAX
IBM PC/AT

The last three of these are, perhaps, somewhat questionable as workstations: the Orion because it is a multi-user system termed by its manufacturers a 'supermicrocomputer', the MicroVAX because it is effectively a scaled down VAX mini-computer and the IBM PC/AT because it would more usually be thought of as a powerful PC.

Unfortunately, Genstat is available on few these computers. The VAX/VMS version of Genstat will transfer to the MicroVAX running MicroVMS. The PC version of Genstat, mentioned above, will run on machines which offer MS-DOS either as the operating system (such as the IBM PC/AT) or as a subsystem (such as the IBM RT PC). There was interest some time ago in putting Genstat on the ICL Perq running PQOS and the Sage but little has come of it. The ICL Perq running PNX and the Whitechapel MG1 running Unix both have Genstat implementations in progress and it is expected that work on an implementation for the PCS Cadmus will begin soon.

Mounting Genstat on the ICL Perq

The ICL Perq is one the earlier workstations and runs the PNX operating system which is based on Bell Laboratories' Unix Version 7. The author has been working on an implementation of Genstat 4.04 for several months, though not continuously.

The relevent experience of the author before beginning the implementation was, on the positive side, ten years of using Fortran (on and off) and, on the negative side, little knowledge of Genstat, little knowledge of Unix and no knowledge of the Perq. It was therefore necessary that the early days were spent getting to know as much as possible about Genstat, the Perq and PNX, and this was done mainly by reading the available documentation.

The first real problem was getting the Genstat source code onto the Perq, since the Genstat source was on magnetic tape and the Perq has no tape drive. However, the computer network being used allowed the Perq to function as a terminal into a VAX with a tape drive. It was then a simple, though very slow, job to transfer the files from the VAX to the Perq. The slowness was a consequence of the large volume of the Genstat source code and the inability of the Perq to cope with a high bit transfer rate: the practical limit was 1200 baud since anything faster invariably resulted in corrupt files.

The large volume of the Genstat source code presented a futher problem at this stage, in that the disc on the Perq did not have enough space free. However, space was easily created by deleting files belonging to other people and even deleting parts of the operating system.

Each of the above problems might easily apply to other implementations on small computers which have low capacity discs, no tape drive (the usual means of reading the Genstat source) and less than ideal file transfer facilities.

One further cause of problems was that the implementation was begun using the Fortran source which had been adapted for another Unix environment. This appeared to be a sensible approach at first, although it later proved to be a considerable mistake due to the subtle variations between the different 'flavours' of Unix. Two major difficulties were encountered on the Perq. First, the other Unix version made a number of calls to system subroutines which were not allowed in PNX. Second, the bit and byte manipulation routines (which all implementors have to provide), although superficially acceptable, could not be used on the Perq due to differences in the way it orders bytes within words. It was eventually decided to restart the implementation using the base version of Genstat; however the time spent prior to that was not entirely wasted, in that valuable knowledge was gained, about the Perq, Unix and Genstat.

Implementors of Genstat on other Unix based systems will need to consider which source code should be used in their implementation. It might be wise to use the source from another Unix system in some cases but, in general, it is probably safer to stay with the base version.

Sadly, the author has not been able to give as much time to the implementation on the Perq as would be desirable, though the work is progressing, albeit at a slow pace.

Conclusion

It is comforting to note that Genstat 5 is being written to Fortran 77 standards, given the increasing adherance of Fortran compilers to those standards. Although there are still differences between Fortran compilers, it is generally hoped that implementors of Genstat on all computers systems will have an easier task in the future.

The need for versions of Genstat on workstations and other small computers is recognised and a start has been made on meeting that need.

Readers interested in implementing Genstat on workstations are invited to contact the author.

Notice

Genstat Primer

Edward Arnold (Publishers) Ltd
41 Bedford Square
London
United Kingdom *WC1B 3DQ*

Edward Arnold Ltd is pleased to announce the Genstat Primer, by A.J. Weekes, lecturer in the Department of Economics and Related Studies at the University of York.

Weekes has written an excellent primer for Genstat users: its individuality is attributable to its usefulness to beginners, not shared by competing texts. Teaching establishments in the United Kingdom and overseas, which have Genstat among their computing facilities, will find this text of great value.

July 1986. 144 pages.
£5.95 net paper 0 7131 3607 3

Available from Edward Arnold Ltd.

Trade Department: Edward Arnold (Publishers) Ltd
Woodlands Park Avenue
Woodlands Park
Maidenhead
Berkshire
United Kingdom SL6 5BS

Telephone: Littlewick Green (062 882) 3104