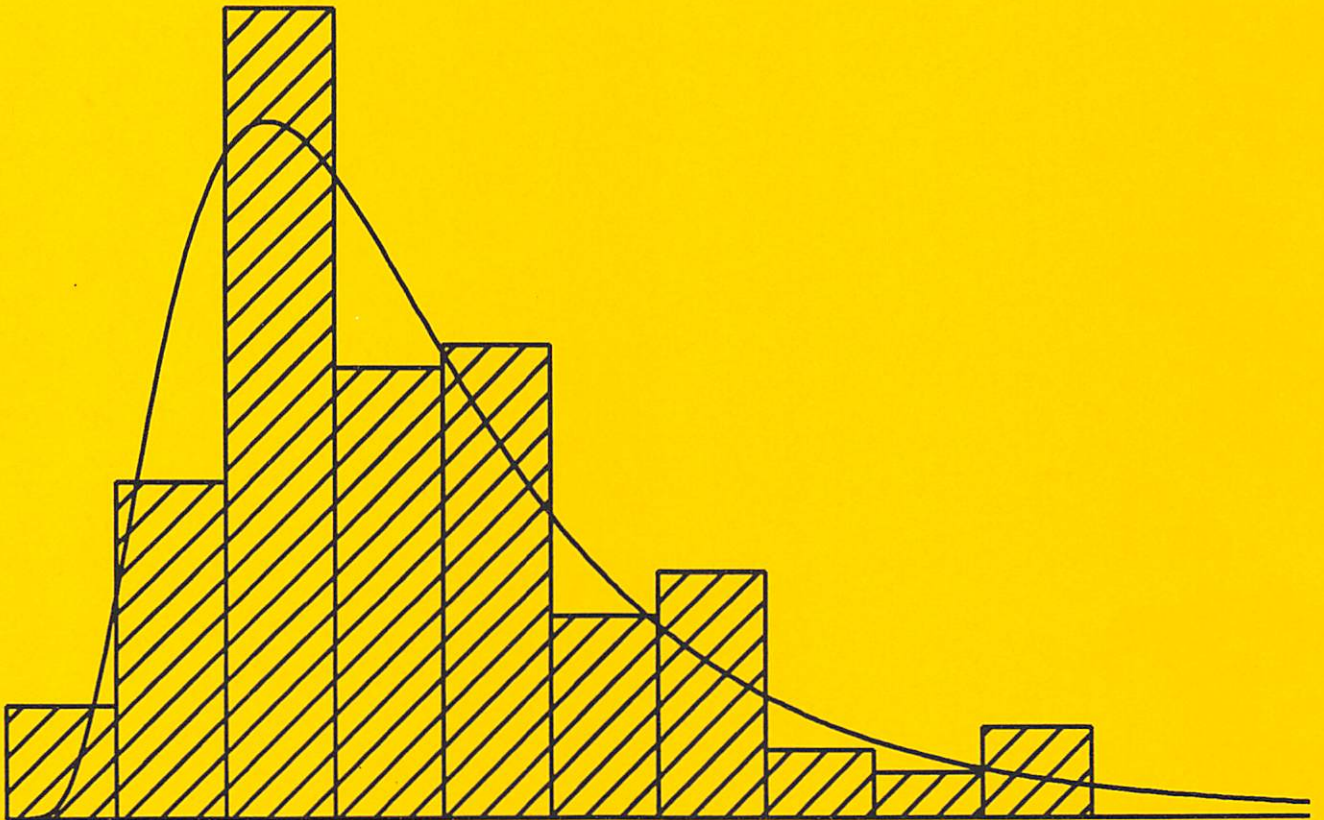


GENSTAT

Newsletter

Issue 31



Editors

Sue Welham
AFRC Institute of Arable Crops Research
Rothamsted Experimental Station
HARPENDEN
Hertfordshire
United Kingdom AL5 2JQ

Anna Kane
NAG Ltd
Wilkinson House
Jordan Hill Road
OXFORD
United Kingdom OX2 8DR

©1994 The Numerical Algorithms Group Limited

All rights reserved. No part of this newsletter may be reproduced, transcribed, stored in a retrieval system, translated into any language or computer language or transmitted in any form or by any means, electronic, mechanical, photocopied recording or otherwise, without the prior permission of the copyright owner.

Printed and Produced by NAG[®]

NAG is a registered trademark of:

The Numerical Algorithms Group Ltd

The Numerical Algorithms Group Inc

The Numerical Algorithms Group (Deutschland) GmbH

Genstat is a trademark of the Lawes Agricultural Trust

ISSN 0269-0764

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

NAG Bulletin Board:

Gopher: Name=NAG Gopher Server, Type=1, Port=70, Path=1/, Host=www.nag.co.uk

Mosaic: <http://www.nag.co.uk:70/>

Genstat Newsletter

Issue 31

Contents

	Page
1. Editorial	3
2. Genstat Talk	4
3. Some recent developments in descriptive multivariate analysis W J Krzanowski	9
4. Using Genstat to develop a taxonomic classification from ribosomal DNA sequences G M Arnold, J A Bailey, C Sheriff and M J Whelan	15
5. The jack-knife and the bootstrap P W Lane and R W Payne	20
6. Efficiency factors for some balanced hyper-Graeco-Latin superimpositions of Youden squares . E D Gardiner and D A Preece	24
7. Solving the depletion equation: an example of inverse nonlinear regression P Brain and L R Saker	30
8. Rungen – a user-friendly Genstat interface D Kilpatrick	35
9. An interface between Genstat and the Brief editor on PC P W Goedhart	41
10. A Genstat procedure to calculate a kappa coefficient of agreement for nominally scaled data . . A J Rook	45
11. Analysis of unbalanced multi-stratum trials using ANOVA and REML R F A Poultney	48

Published by
The Rothamsted Experimental Station Statistics Department
and The Numerical Algorithms Group Ltd

Editorial

It is with deepest regret that the editors must report the death of Pete Digby following a fall at his home in Harpenden. Pete joined Rothamsted in 1979 from the AFRC Unit of Statistics at Edinburgh to take over the development of the multivariate section of Genstat. He also contributed to the design of Genstat 5 and to its Procedure Library. Pete was involved in the general scientific research programme at Rothamsted and will be remembered for his innovative use of multivariate techniques, for example in the analysis of oil-seed data and in his book "Multivariate Analysis of Ecological Communities" (with Rob Kempton). His friends will also remember his out-of-work interests in photography and cricket. In recent years Pete suffered from epilepsy and this may have been a factor in his fall. Pete's lively contributions at Genstat conferences will be greatly missed, as will his contribution to Genstat generally.

This issue of the Newsletter is the second from the new editorial team, and the editors would like at this point to describe once again the kind of articles accepted for publication in Genstat Newsletters. Many Newsletter articles are written by the Genstat developers, to describe the uses of new Genstat facilities and procedures, but the editors are also keen to receive papers from users who have found interesting and innovative applications for the Genstat system. The Genstat Newsletter is for all Genstat users, hence papers are welcomed from any users who have ideas and / or procedures they wish to share. In the first instance, papers should be directed to Sue Welham of Rothamsted Experimental Station.

Included with this issue is the first call for papers for the Ninth International Conference of Genstat Users, which is to be held in Dublin in July 1995. Full details and addresses can be found on the enclosed flyer. A convenient index of directives and associated manual reference pages is also enclosed, which is designed to fit inside the Genstat Manual cover, to provide a quick user reference.

Another helping of Genstat Talk is provided, together with the usual details on how to join the Genstat electronic discussion list. A broad selection of topics are dealt with by the articles in this issue, beginning in earnest with an overview of some of the recent developments in descriptive multivariate analysis, followed by a paper concerning the use of Genstat in developing a taxonomic classification from ribosomal DNA sequences.

Next comes a discussion of how the new Genstat procedures JACKKNIFE and BOOTSRAP can be used in forming jackknife and bootstrap estimates for any statistic that can be calculated in Genstat. Two involved articles then follow, dealing firstly with the superimposition of Youden squares and efficiency factors, and secondly with the solving of the depletion equation in Genstat using inverse nonlinear regression and the FITNONLINEAR directive.

Continuing from a topic introduced in Newsletter 30, this issue describes two new interfaces for the Genstat system. The first, RUNGEN, provides a user-friendly interface to writing Genstat programs for non-statisticians and also offers facilities which simplify the input of spreadsheet data. The second discusses an interface between Genstat and the Brief editor on a PC.

The final article in this issue introduces a Genstat procedure to calculate a kappa coefficient of agreement for nominally scaled data. As usual, the code for this and any other procedure appearing in any Genstat Newsletter will be posted on the NAG bulletin board.

GENSTAT TALK

Extracts from the Genstat electronic discussion list, November 1993 to April 1994, summarized and edited by Peter Lane, Rothamsted. To join the discussion, send the message:

SUBSCRIBE Genstat *first-name last-name*
to the address: `LISTSERV@IB.RL.AC.UK`

The opinions expressed here are not necessarily endorsed by either NAG or Rothamsted, and statements may not have been checked for accuracy. However, members of the Genstat development team and of NAG's Statistics Section are contributors to the discussion.

Maxima of matrices

Query: What is the most elegant way of identifying the row and column containing the largest value in a symmetric matrix?

tjc1@phoenix.cambridge.ac.uk

Reply 1: The POSITION function can be used to find the elemental position, but this needs converting to row and column numbers. The following solution is not necessarily the most elegant, and does not find multiple solutions, but it is fast and efficient.

```
SYMMETRIC dists ...  
CALC n = NROW(dists)  
& index = CUM(1(1...n)) - \  
  POSITION(MAX(dists); 1(#dists))  
& row = POSITION(1; index.GE.0)  
& col = row - index$[row]
```

simon.harding@afrc.ac.uk

Further replies: Successively briefer pieces of code were supplied to provide all solutions and row and column index vectors for the matrix, by *rod@tui.marc.cri.nz*; *anon*; and *ruth.butler@afrc.ac.uk*

PostScript graphics

Query: Does anyone have any experience of transferring a PostScript file generated by Genstat into a wordprocessor package? I use Genstat on a Vax and Word on a Macintosh.

ian.wakeling@afrc.ac.uk

Reply: I have transferred both PostScript and HPGL files from Genstat to WordPerfect on a PC with no problems. You probably need to transfer them as binary or data files (not as text or ASCII). Usually I use one file per diagram otherwise the wordprocessor gets confused. I prefer HPGL because I can see them when inside WordPerfect.
peteb@prospect.anprod.csiro.au

REML and ANOVA

Query: I recently posted a query about differences between output from REML and regression. It turned out that I asked the wrong question and I should have been asking about the differences between ANOVA and REML. I want to use a fixed effects structure *area/region/family*, but get space problems with REML so have used *area+region+family*. Why does REML give different answers to ANOVA? (Program attached.)

jeff@canopy.biom.csiro.au

Reply: The model you want to fit is:

```
area + area.region + area.region.fam  
3 levs 3x9=27 levs 27x210=5670 levs
```

ANOVA is clever enough to realize that you have only three regions for each area, and so on, but REML sets up the full matrix with 6000 rows before finding this out, hence your space problems. If you use redefined factors, numbering regions within areas, for example, this problem will be much less severe.

To get means, REML forms a full table of predicted means for each factor combination and then takes marginal means to get the tables of predicted means that it wants. So if you fit *area/region/family* your means are formed as you would expect (assuming no missing cells). But with *area+region+family*, the means for area 1, for example, are formed using an average over all regions, not just the ones in area 1. The difference between ANOVA and REML output is really just this question of actual versus predicted means – you need to be aware of how a different model specification can affect the way in which the means are formed.

In Release 3.2 space-saving measures will be implemented to avoid including unnecessary rows in the mixed model equations.

sue.welham@afrc.ac.uk

Restructuring input data

Query: I have some count data classified by two factors, but excluding combinations of levels with zero counts. I would like to have the full set of combinations in Genstat. I can define the full set of levels for the factors and generate their values, then set up a variate with all zeroes; how do I insert the non-zeroes?

r.a.reese@ucc.hull.ac.uk

Discussion: Several suggestions were made, and deep thoughts expressed about conditioning caused by different packages' approaches. The final reply seemed to answer the question, and advertized a little known feature of Genstat's approach.

Reply: There is a feature in Genstat that allows you to read numbers indexed by a key (the units structure) and it will fill out the design. Here is an example, reading seven non-zero counts and filling the other eight combinations with zeroes.

```
VARIATE [VAL=1...15] unit
UNIT [15] unit
FACTOR [LEV=3] f1
& [LEV=5] f2
GENERATE f1,f2
VARIATE count
READ unit,count
  1 10
  2 11
 11 24
  4 10
  3  5
  7 15
 15  3 :
CALC count = MVREP(count; 0)
fillmore@nsrske.agr.ca
```

Variance of linear predictor

Query: Does anyone know how to get the variances of linear predictors after fitting a GLM?

alastair@sass.sari.ac.uk

Reply: The linear predictor is Xb where X is the design matrix and b is the vector of estimates. So its variance is XVX' where V is the covariance matrix of the estimates.

```
RKEEP VCOV=V; DESIGN=X
CALC vlinpred = QPRODUCT(X; v)
In the linear model, the variance is simply leverage*dispersion, but I'm not sure whether there is a simple relationship involving the leverage in a GLM.
```

peter.lane@afrc.ac.uk

Simulating binomial

Query: Does anyone know how to simulate an ' n -units' binomial variate (I have an algorithm that is very slow)?

alastair@sass.sari.ac.uk

Reply 1: The new GRANDOM procedure in Library 3[1] does this, along with many other distributions. The code used for the binomial $B(n,p)$ is as follows:

```
CALC t[1...n] = URAND(0; seed) <= p
& random = VSUM(t)
```

peter.lane@afrc.ac.uk

Reply 2: This solution is OK for fixed n and p , but it would be useful for regression problems to be able to generate variates of binomial random variables in which n and/or p vary from unit to unit. The solution above would be slow; does anyone know of a better approach?

martin.ridout@afrc.ac.uk

Forming variates by replication

Query: I have a list of values and a list of the numbers of repeats of each value. How can I convert this into a variate or factor containing each value repeated the right number of times?

ruth.butler@afrc.ac.uk

Reply 1: The following will form a factor *res* from variates *reps* and *vals*:

```
CALC nv = SUM(reps)
& nl = NVAL(vals)-1
VARIATE [VAL=#reps${1...nl}] lim
& [VAL=1...nv] vfac
CALC lim = CUM(lim)+0.5
SORT [INDEX=vf; GROUPS=res; \
LIMITS=lim]
FACTOR [MODIFY=y; LEV=vals] res
p.w.goedhart@glw.agro.nl
```

Reply 2: In the above solution, the FACTOR statement gives an error if one of the variate values is missing. I have modified the solution to cope with this, and have a procedure called REPEAT.

tod@maths.marc.cri.nz

Reply 3: Here is another way to do the repetition, perhaps a bit simpler, but not much! It works for missing values in *vals*, and forms a variate.

```
CALC a = CUM(reps)-reps+1
& p = SUM(reps)
VARIATE [VAL=#p(0)] b
CALC b${a} = 1
& b = CUM(b)
& res = vals${b}
```

peter.lane@afrc.ac.uk

Reply 4: The EXPAND function simplifies this job.

```
CALC n = NVAL(reps)
CALC w[1...n] = \
0*expand(#reps; #reps)+#vals
VARIATE [VAL=#w] res
snspted@reading.ac.uk
```

Predictions from ordinal models

Query: I have applied the ORDINALLOGISTIC procedure to a data set on the reproductive success of swifts, but I cannot interpret the resulting parameter estimates. How do I get the predicted probabilities for each category derived from the model?

David Thomson

Reply: We have written up an interpretation of the procedure output including both the equations relating the cumulative probabilities to the covariates and the probabilities of belonging to a particular category. We would be happy to fax you a copy.

constat@durras.anu.edu.au

Estimates for ARIMA models

Query: I have used Genstat and another package to fit autoregressive processes of order 1. The results are sufficiently different to make me wonder which results to use. For example, an autoregression coefficient of 0.88 with s.e. 0.098 from Genstat compared with 0.70 from the other package. The other package uses Yule-Walker equations for the fitting, and in Genstat I have used the exact setting. Is the difference in fitting methods enough to explain the differences or are there other differences I need to take account of?

jeff@canopy.biom.csiro.au

Reply: The difference between the exact likelihood estimate of 0.88 and the Yule-Walker estimate of 0.70 is almost certainly due to the difference in method. Genstat should give you Yule-Walker estimates if you use the PTSM directive to form preliminary estimates of the model parameters. The difference is usually not so great unless the series is short.

maa016@central1.lancaster.ac.uk

Combine R from N

Query: I am looking for a fast Genstat program, preferably with few loops, to generate all subsets of size R from N things.

p.w.goedhart@glw.agro.nl

Reply 1: I have a procedure called COMBNR which evaluates all combinations of size R from the set (1..N). It is very fast, but uses a lot of space for large R (order N^R). This is OK for small R but may be prohibitive. For suitable N and R this method may be preferable to an algorithm which runs in less space but more time.

```
PROC 'combnr'
PARAM 'N','R','OUT'; \
  TYPE='scal','scal','poin'
CALC nr = N**R & r1 = R-1
FACTOR [nr; LEV=N] f[1...R]
GENERATE f[]
CALC lt[1...r1] = \
  f[1...r1]<f[2...R]
RESTRICT f[]; \
  VSUM(lt).EQ.r1; SAVE=ss
MATRIX [ss; R] x
CALC x$[*; 1...R] = \
  f[1...R]$\[ss]
& nout = NVAL(ss)
VARIATE [R] OUT[1...nout]
EQUATE x; OUT
ENDPROC
```

anon

Reply 2: Here is a less demanding, but less general, solution to the procedure, using loops. This example solves for N=6 and R=3.

```
SCALAR [VAL=1] i
FOR k1=1...4
  CALC k11 = k1+1
  FOR k2=k11...5
    CALC k21 = k2+1
    FOR k3=k21...6
      CALC x[i] = 1(k1,k2,k3)
      & i = i+1
    ENDFOR &
  ENDFOR &
```

fillmore@nsrske.agr.ca

Covariate for logit analysis

Query: I'm modelling a set of binomial data with a logit model. Some of the interactions looked a bit funny, and someone pointed out that the data have a rough trend in them, roughly exponential. If I estimate this trend with FITCURVE, how can I then use it in the logit analysis? If I just subtract the fitted values from the actual counts or proportions, I won't be able to do a logit analysis on the residuals: some will be negative, and what is the corresponding NBIN?

duncan.hedderley@afc.ac.uk

Reply: I think you want to use an offset. If f is the fitted proportion:

```
CALC o = LOGIT(f)
MODEL [DIST=bin; OFFSET=o] ...
```

This will effectively remove the effect of f on the scale of the linear predictor.

rod@maths.marc.cri.nz

Elements of tables

Query: Is it possible to get at the numbers in a (one-dimensional) table, say using element operators? I know you can with EQUATE, but I would like a more direct method. (Normally to this kind of question I'd say RTFM, but the latest version of the manual seems to be at the printers!)
duncan.hedderley@afrc.ac.uk

Reply: Unfortunately, it is not possible in Releases 2 or 3 to use the \$[] notation or the ELEMENTS function to access individual values of tables. The quickest way for a one-dimensional table is probably to put the values in a variate and then use \$[]:

```
TABLE [CLASS=f; VALUES=...] t
CALC s2,s5 = 1(#t)$[2,5]
```

BTW, the Release 3 manual is now available from OUP.

Translation for bewildered readers:

BTW = By The Way

RTFM = Read The Forgotten Manual
(censored translation)

IMHO (one that had me puzzled for months)
= In My Humble/Honest Opinion

OUP = Oxford University Press

peter.lane@afrc.ac.uk

All-in-one analysis

Query: I've had two people in here this morning with what feels like the same problem; I wonder if anyone can suggest a technique to solve it. In one case we have a number of treatment groups, each consisting of about 30 people. They were asked to rate how easy they found certain dietary changes on a scale 1 to 9. We want to see if the dietary changes that were seen as difficult differed between groups. We could do ANOVA on the scores, or use a chi-squared test to compare proportions of people who found the change a problem. What I'm wondering is, is there a single analysis we can do that will tell us something about both the numbers in each group and the strength of their feelings?

duncan.hedderley@afrc.ac.uk

Reply: Turn the problem round, and, rather than treating the ratings as being something that they are not, analyse the frequencies of each rating. This leads to McCullagh's ordinal logistic regression model as one possibility. Within Genstat, the library procedure ORDINAL-LOGISTIC is available in Release 2, and the MODEL directive has new options to fit this model in Release 3.

llefkovi@ccs.carleton.ca

Three-way PCA

Query: Does anyone out there have experience of doing three-way (or three-mode) PCA; in other words, a principal components analysis when the data can be classified by three different factors? We have someone who wants to get a map of the relationship between topics and aspects, based on the answers from a survey of 300 people. At present, she is averaging the scores for each topic-by-aspect combination over the 300 and doing an ordinary PCA on that. It seems to me that this way discards all the information about the different people.

duncan.hedderley@afrc.ac.uk

Summary: Thanks to everyone who sent suggestions. I'm still following them up, but in the meantime, here are the references I was given.

Williams & Gillard (1971) Pattern analysis of a grazing experiment. *Australian Journal of Agricultural Research* 22, 245-260.

Several people mentioned Peter Kroonenberg (Leiden University) to me. He has written several papers, including Kroonenberg & DeLeeuw (1980) Principal component analysis for three-mode data by means of alternating least-squares algorithms. *Psychometrika* 45, 69-97; also a PhD thesis, and a PC program.

duncan.hedderley@afrc.ac.uk

Manipulation of strings

Query: I need to print single strings extracted from a text during successive passes through a loop. I have a solution using RESTRICT that seems too complicated; does anyone have a better solution?

```
VARIATE (VALUES=1...nv) subs
FOR i=1...nv
  RESTRICT text; subs.EQ.i
  PRINT text
```

...
john@marc.cri.nz

Reply: There is no simple solution because Genstat's syntax for referring to elements is designed for numerical structures and has not yet been extended to texts. One solution is to use READ:

```
TEXT [1] lab
FOR i=1...nv
  READ [*; CHAN=text; END=*] lab
  PRINT lab
```

A second uses EQUATE, but is messy.

peter.lane@afrc.ac.uk

Counting runs of zeroes

Query: I wish to find the frequency distribution of consecutive runs of zeroes in a long variate. All the data are non-negative. Does anyone have some slick code to do this sort of thing?

sassjm@scri.sari.ac.uk

Reply 1: Here is a solution, starting from variate *v*, using the fact that 0/0 gives a missing value.

```
CALC cumv = CUM(v)+0/(v.EQ.0)
SORT [INDEX=cumv; GROUP=dist]
TABULATE [COUNT=runs; CLASS=dist]
SORT [INDEX=1(#runs); \
      GROUP=fruns; LEV=1runs]
TABULATE [count; CLASS=fruns]
```

anon

Reply 2: Here is another solution. I assume that missing values can be regarded as non-zero.

```
VARIATE [VAL=1, #v] z
CALC z = CUM(MVREP(ABS(z); 1))
SORT [INDEX=z; GROUP=f; LEV=1]
TABULATE [CLASS=f] z; NOBS=n
CALC z1 = 1(#n)-1
& z1 = MVINSERT(z1; z1==0)
SORT [INDEX=z1; \
      GROUP=fruns; LEV=1runs]
TABULATE [nobs; CLASS=fruns] z1
```

d.c.van.der.werf@ibn.agro.nl

Storing correlation coefficients

Query: On Page 535, the 1987 Manual states that **CORRELATE** can only display the correlations; you have to use the **FSSPM** directive and the **CORRMAT** function to store them. I've tried, but can't see how it can be done.

callinan1@rust.agvic.gov.au

Reply: The answer is simple if you understand that an **SSPM** structure is a compound structure made up of three simple structures: the sums (symmetric matrix), the means (variate) and the number of units (scalar).

```
SSPM [TERMS=x, y, z] ssp
FSSPM ssp
CALC cor = CORRMAT(ssp[1])
```

j.currall@compserv.gla.ac.uk

Aliased effects in ANOVA

Query: A colleague has analysed an experiment carried out at several orchards in several regions. We need region means as well as orchard means, but neither **REML** nor **ANOVA** seem to be able to cope. Does the problem occur only when one factor is completely aliased with the other? Has it been fixed in the latest release?

```
VARIATE [VAL=1...10, 0.5, 1.5...9.5] y
FACTOR [LEV=5; VAL=2(1...5)2] orchard
FACTOR [LEV=3; VAL=(4(1,2), 2(3))2] \
      region
```

```
TREAT region+orchard
```

```
ANOVA [PRINT=mean] y
```

gives means for region: 2.25, 6.25, 9.25

and for orchard: 4.25, 6.25, 4.25, 6.25, 5.25

which are nonsense. If we use

```
TREAT orchard+region
```

then no means are produced for region at all.

john@marc.cri.nz

Reply 1: I haven't actually checked on the computer, but I would imagine that what you want is the nesting operator (*/*), so that you can specify

```
VCOMP [FIXED=region/orchard]
```

to get the region effect and the interaction.

duncan.hedderley@afrc.ac.uk

Reply 2: The problem occurs because there is partial aliasing between the two treatment terms, so perhaps the specification isn't particularly sensible! In fact, orchards are nested within regions, so if you use

```
TREAT region/orchard
```

all will be well. It's actually more efficient if you number the orchards within the regions (Manual, Page 413). There is nothing wrong with the means reported above: the formula **region+orchard** is expected to give means of orchards eliminating regions, which is what appears. However, in Release 3.1 you will get a warning about partial aliasing in cases like this, at which point your colleague will be prompted to come and see you so that you can explain the difference between factorial and nested designs.

roger.payne@afrc.ac.uk

Convex hull

Query: Does anyone know how to determine whether a point falls within a convex hull, found using the **CONVEXHULL** procedure? I understand that this might be a problem best solved by linear programming.

mo_ths@vaxa.nerc-monkswood.ac.uk

Reply 1: A long time ago I wrote a procedure called **INSIDE** that determines which elements of a variate lie within a specified polygon. I wrote it for use with **DREAD** but it will obviously work with **CONVEXHULL** as well. It seems to fail occasionally; feel free to try it out and send remedies to me so I can sort out bugs and submit it to the Procedure Library! (Procedure attached.)

simon.harding@afrc.ac.uk

Reply 2: The **INSIDE** procedure works only if the origin is an outside point of the polygon. A small amendment to the code makes it work. (Amendment attached.)

anon

Some recent developments in descriptive multivariate analysis

W J Krzanowski

Department of Mathematical Statistics and Operational Research

University of Exeter

Laver Building, North Park Road

EXETER EX4 4QE, UK

Abstract

Principal component analysis, principal coordinate analysis, and canonical variate analysis are popular descriptive multivariate analysis features of Genstat. Various extensions, developments and generalisations of these techniques have been proposed in recent years, giving the user potentially much more scope at the expense of relatively little extra effort. This article gives a brief (and selective) overview of some of these developments.

1. Introduction

The starting point for many descriptive multivariate techniques is often an $n \times p$ data matrix X , the (i, j) th element x_{ij} giving the response for the j th observed variable on the i th sample individual ($i = 1, \dots, n$; $j = 1, \dots, p$). The most useful summary statistics for such a sample of data are the mean vector

$$\bar{x}' = (\bar{x}_1, \bar{x}_2, \dots, \bar{x}_p),$$

and the $p \times p$ covariance matrix S containing the variances of the p variables down the leading diagonal and the $p(p-1)/2$ covariances between pairs of variables in the off-diagonal positions. If X has been mean-centred (by subtracting the mean vector from each row) then

$$S = \frac{1}{n-1} X'X,$$

while if the variables have been standardised (by dividing each element of the mean-centred X by the appropriate standard deviation) then S is the correlation matrix R . We assume in the following that X has been mean-centred but not necessarily standardised.

If the data are continuous (or, at least, numerical), a convenient geometrical model of the sample identifies the n individuals as n points and the p variables as p orthogonal axes in p -dimensional space, the coordinates of the i th point on these axes being given by the values in the i th row of X . Since X is assumed to be mean-centred, the origin of the axes is at the centroid of the points. Inspection of the data swarm in this space will reveal any interesting features that might be present, for example groupings of the points or outlying individuals or obvious relationships among them. Such a model cannot in general be viewed directly, however, as in most applications p is greater than 3. The objective of many descriptive multivariate methods is therefore to effect a reduction into a small number of dimensions in which the data swarm may be inspected for these interesting features. It will be convenient in the following to refer to the original space as the X -space, and to the reduced-dimensional space as the Y -space. Three techniques in particular are very popular for deriving an appropriate Y -space in different circumstances, so we first describe them briefly.

Principal component analysis produces a projection of the original points into a low-dimensional subspace of given dimensionality k , the chosen subspace being the one in which the overall scatter of points is maximised. Let P_i ($i = 1 \dots n$) denote the n points in the X -space, P'_i denote their projections in the Y -space, and O denote the origin of axes in both spaces. The requirements of maximising scatter whatever the value of k imply that the variance of the projections decreases along successive axes in the Y -space, and that the Y -space is the k -dimensional subspace in which $V_1 = \sum_i (OP'_i)^2$ is maximised. Simple geometric arguments establish readily that it is also the subspace projection in which $V_2 = \sum_i (P_i P'_i)^2$ and $V_3 = \sum_i \sum_j [(P_i P_j)^2 - (P'_i P'_j)^2]$ are both minimised (projection implying that $P'_i P'_j \leq P_i P_j \forall i, j$). Moreover, from V_3 we see that the Euclidean distances between points in the Y -space approximate the corresponding Euclidean distances in the X -space. The fundamental

algebraic operation underlying principal component analysis is the decomposition of S into its eigenvalues (elements of the diagonal matrix D) and eigenvectors (columns of the orthogonal matrix L).

If the rows of X have come from g *a priori* groups, a Y -space in which overall scatter of points is maximised may not be the most useful space in which to view the data. More appropriate may be a space in which the separation between groups is maximised in some way, and canonical variate analysis provides the subspace in which the ratio of between-group to within-group scatter is maximised. The fundamental algebraic operation here is extraction of eigenvalues and eigenvectors of $W^{-1}B$, where W is the covariance matrix *pooled within groups* while B is the covariance matrix *between groups*. Euclidean distances between group means in the Y -space now approximate the *Mahalanobis* distances between the corresponding means in the X -space.

Principal coordinate analysis is ostensibly a different sort of technique from either of the above, as it does not start from a high-dimensional model from which a low-dimensional approximation is to be derived. Rather, it starts from a matrix of inter-point distances (or inter-object dissimilarities) and then *constructs* a low-dimensional configuration in which the distances between points are approximated, or the dissimilarities between objects are represented, as well as possible. By 'as well as possible' is meant in the sense of V_3 above. However, in spite of these apparent differences, the technique has much in common with the two previous ones: its algebraic basis is very similar to theirs (extraction of eigenvalues and eigenvectors of a simple transformation of the input distance/dissimilarity matrix), and it produces the same results in certain special cases. If the input matrix is the $n \times n$ matrix of Euclidean distances computed from X then principal coordinate analysis yields the same Y -space as does principal component analysis of X , while if the input is the matrix of Mahalanobis distances between group means in a grouped data set then an equivalent of canonical variate analysis is achieved. (Strictly, this latter technique yields the same result as the canonical variate analysis derived from the unweighted between-groups matrix

$$B = \frac{1}{g-1} \sum_{i=1}^g (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})',$$

where \bar{x}_i is the mean of the i th group of individuals, whereas standard canonical variate analysis is based on the weighted between-groups matrix

$$B = \frac{1}{g-1} \sum_{i=1}^g n_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})',$$

where n_i is the number of individuals in the i th group. However, in many circumstances the weighted and unweighted analyses do not differ materially. Also, Ashton, Healy and Lipton (1957) have argued that the unweighted analysis is better for descriptive purposes, so in such a case the principal coordinate approach is appropriate.) The extra benefit afforded by principal coordinate analysis is that non-numerical variates can be catered for, as distances/dissimilarities can be defined and thus calculated for such variates. Hence geometric representations can be obtained in low dimensions even though no original model exists for such data.

All the above is familiar, and the three multivariate techniques outlined are very popular among users of Genstat. More detail on the basic techniques can be obtained from the Genstat manual or from a variety of multivariate texts (see, e.g., Digby and Kempton, 1987). Our purpose in the next three sections is to bring to the attention of such users a number of areas of development of these techniques, in the hope that a wider set of potential applications might thereby be opened up. Although none of these areas is currently catered for explicitly in Genstat, some can be adapted fairly easily from existing facilities. By disseminating the ideas more widely, it is also hoped that the facilities might be incorporated in future releases of the system.

2. Common principal components

Consider optimality criterion V_2 of the previous section as one of the possible characterisations of principal components. This criterion lay at the heart of one of the earliest derivations of principal components (Pearson, 1901), in terms of lines and planes of closest fit to a set of points. Such a derivation highlights connections between principal component analysis and regression analysis: the former fits lines by minimising orthogonal deviations, the latter by minimising 'vertical' or 'horizontal' deviations. Let us pursue these connections a little further. In regression analysis, the data are often grouped and comparisons of regressions between the groups are of interest. Such comparisons are conducted by constraining some or all of the regression coefficients to be

equal in the groups, and then testing whether the fit of the resulting relationships is significantly worse than if the regression coefficients had been free to vary separately in each group. Arranging the constraints in a hierarchic structure assists in the testing process.

A parallel analysis can be envisaged for principal components. Suppose that the n individuals in the sample are divided *a priori* into g groups, with n_i individuals in the i th group, each group has been described separately by its principal components, and we wish to see whether these components have common features among the groups. To effect the analysis, we first need to formulate suitable models for the populations from which the groups have been obtained, and then to impose a hierarchic system of constraints on the population parameters corresponding to the features of interest.

Let the (population) dispersion matrix for the i th group be Ω_i and its estimate from the sample be S_i ($i = 1..g$). Total homogeneity between groups is expressed by the null hypothesis

$$H_0: \Omega_i = \Omega \quad \forall i,$$

which can be tested against the general alternative H_a that at least one Ω_i is different from the rest (assuming normal data) by means of the likelihood ratio test statistic

$$T_1 = n \ln |W| - \sum_{i=1}^g n_i \ln |S_i|$$

which has a chi-squared distribution on $p(p+1)(g-1)/2$ degrees of freedom if H_0 is true.

A good intermediate between equal dispersions and arbitrary dispersion, is the *common principal component model*

$$H_c: \Omega_i = L D_i L'$$

In this model the individual dispersion matrices have the same principal components, but these components may have different variances (and hence different orderings) in the different groups. Considerable heterogeneity of dispersion matrices can be accommodated within this structure, and hence the model can cater for many practical situations. Theoretical aspects of the model have been studied in a series of papers by Flury, a unified account of which can be found in Flury (1988). Estimates of L and D_i can be found either by maximum likelihood (assuming normality) or by least squares, and likelihood ratio tests exist with statistics T_2 (for H_c versus H_a) and T_3 (for H_c versus H_0) satisfying $T_1 = T_2 + T_3$. Algorithms for the estimation are given by Flury and Constantine (1985), Flury and Gautschi (1986), and Clarkson (1988); the likelihood ratio tests are derived by Flury (1988), while some simple *ad hoc* and intuitive versions of these estimates and tests have been suggested by Krzanowski (1984). Proportional dispersion matrices are obtained as a special case of this model ($D_i = \alpha_i D$), and additional possibilities are *partial common principal components* (Flury, 1988) or *common principal component spaces* (Schott, 1991).

In addition to the direct purpose of investigating principal component structure between groups, the common principal component model has played a part in generalising some familiar multivariate techniques. The first such generalisation is that of canonical variates. Recall from Section 1 that the technique requires the eigenvalues and eigenvectors of $W^1 B$. Campbell and Atchley (1981) have shown that these quantities can be found equivalently by the following steps.

1. Find the eigenvalues (diagonal elements of the diagonal matrix E) and eigenvectors (columns of the orthogonal matrix U) of W : $W = U E U'$.
2. Transform the data: $v_i = E^{-1/2} U' x_i$, where x_i is the i th row of X .
3. Find the eigenvalues F and eigenvectors A^* of

$$C = \sum_{i=1}^g n_i (\bar{v}_i - \bar{v})(\bar{v}_i - \bar{v})'$$

where \bar{v}_i is the mean of group i and \bar{v} the grand mean with respect to variables v : $C = A^* F A^{*}$.

4. The required eigenvalues and eigenvectors are then given by F and $A = U E^{-1/2} A^*$.

This analysis requires homogeneity of population dispersion matrices, i.e., hypothesis H_0 above needs to be true. If H_0 is not true but H_c can be assumed, then Krzanowski (1990) has suggested the following generalisation of the above technique.

1. Conduct a within-group *common* principal component analysis: $S_i = \hat{L} \hat{D} \hat{L}'$.

2. Transform the data:
$$\bar{v} = \hat{D}^{-1/2} \hat{L}' \bar{x}_i$$

where \bar{x}_i is the mean of the i th group with respect to the original variables.

3. Find the eigenvalues and eigenvectors of C (defined as above), and display the group means on *these* eigenvectors as axes.

A second application of the common principal component model is in two-group discriminant analysis. Often, a linear discriminant function is not appropriate because dispersion matrices are not equal in the populations, but sample sizes may be small and a quadratic discriminant function may not be reasonable either. Intermediate discriminant functions, obtained by assuming either the common principal model or proportional dispersion matrices have been studied by Flury and Schmid (1992) and Flury *et al* (1994). In general it appears that assuming proportional dispersion matrices produces good results, but the common principal component model only provides marginal advantages in some special cases.

Finally, Flury and Neuenschwander (1995) explore the implications of assuming common principal components in the context of canonical correlation analysis, and thereby propose a generalisation of canonical correlation analysis to more than two sets of variables.

3. Distance-based methods

We saw above that problems encountered in canonical variate analysis when dispersion matrices are heterogeneous can be overcome if the common principal component model is appropriate. Problems also arise if not all variables are continuous, as between- and within-group matrices B and W are no longer obtainable. Recollect from Section 1, however, that applying principal coordinate analysis to the matrix of pairwise Mahalanobis distances between groups will yield the equivalent of a canonical variate plot of group means. Individuals can then be superimposed on this plot, if desired, by using Gower's (1968) technique for adding points to an existing configuration. This method only requires the additional distances between each added point and the group means of the configuration to be specified, and can be implemented using the ADDPOINTS directive in Genstat.

To generalise canonical variate analysis to any types of variables, we could therefore use the principal coordinate approach providing we were able to define suitable distances between groups and also between individual points and group means. Now distance between two *individuals* is a very familiar concept (e.g., in cluster analysis), and many distance functions are available to the user. For a mixture of any variable types, distance based on Gower's (1971) general coefficient of similarity is the most flexible possibility, and is the one implemented in Genstat. If we denote the distance between individuals i and j by d_{ij} and if the sample is divided *a priori* into g groups $\pi_1 \dots \pi_g$ with n_i individuals in π_i ($i = 1 \dots g$) then Rao (1982) suggests defining the squared distance between π_i and π_j by

$$D_{ij}^2 = \frac{1}{n_i n_j} \sum_{r \in \pi_i} \sum_{s \in \pi_j} d_{rs}^2 - \frac{1}{2n_j^2} \sum_{r \in \pi_i} \sum_{s \in \pi_j} d_{rs}^2 - \frac{1}{2n_i^2} \sum_{r \in \pi_i} \sum_{s \in \pi_j} d_{rs}^2 .$$

To obtain the squared distance $D_{(ij)}^2$ between an individual x_i and the group π_j , we simply need to set $n_i = 1$ in the above expression to give

$$D_{(ij)}^2 = \frac{1}{n_j} \sum_{r \in \pi_j} d_{ir}^2 - \frac{1}{2n_j^2} \sum_{r \in \pi_j} \sum_{s \in \pi_j} d_{rs}^2 .$$

Krzanowski (1994) discusses this approach to generalised canonical variates, and provides some illustrative examples. The same basic idea has been used for discrimination and classification with mixed-mode data (Cuadras, 1989, 1991, 1992) and for regression with mixed data (Cuadras and Arenas, 1990).

4. Nonlinear generalisations

Most classical multivariate descriptive techniques are concerned with seeking optimal *linear* combinations of the observed variables, use methods of *linear* algebra, and look for configurations of points in *linear* spaces and subspaces. Attention has turned increasingly, however, to nonlinear generalisations of these techniques. An early attempt in this direction was the suggestion of Gnanadesikan (1977) to augment the list of variables by including their squares and cross-products, and then to do a principal component analysis on the covariance matrix of the augmented set in the hope of detecting any nonlinear structure that may be present. Gnanadesikan illustrated this idea on an artificial example in which an exactly circular structure was detected, but Flury (personal communication) showed the solution to be extremely unstable under slight perturbations. This lack of stability perhaps explains why the technique has been so little used in practice.

More systematic attempts at nonlinear descriptive methods have been made in the past ten years by a variety of researchers. Gower and Harding (1988) propose a technique for nonlinear biplotting, Hastie and Steutzle (1989) consider the idea of principal curves as a generalisation of principal components, while Gifi (1990) and Meulman (1986, 1992) build general nonlinear multivariate systems encompassing a variety of descriptive techniques. These latter two approaches start from the same premise, namely that all classical descriptive multivariate methods can be derived by finding the low-dimensional configuration of points which minimises a suitable *loss function*, but differ in the types of function considered. Gifi bases loss functions on the concept of homogeneity of variables, while Meulman bases loss functions on the concept of distances between points. The nonlinear generalisations then follow the same patterns in both systems: an optimal nonlinear transformation of each variable is sought in conjunction with minimisation of the appropriate loss function in terms of the transformed data. Various restrictions are needed to ensure uniqueness, and in general one ends up requiring to conduct a double optimisation. Alternating least squares can generally be employed to solve the problem, but the computing involved is usually rather heavy.

5. Comment

We have briefly sketched three recent strands of development of the most popular descriptive multivariate methods. All have already proved their worth in a variety of substantive applications; see Airoidi and Flury (1988) for a common principal components application in zoology, Tyteca and Dufrêne (1993) for some distance-based analyses in botany, Banfield and Raftery (1992) for an interesting application of nonlinear principal components in image analysis, and Gifi (1990) and Meulman (1986) for a variety of other nonlinear applications. The potential scope of descriptive multivariate analysis has thus been considerably widened. Ease of implementation, however, varies somewhat between the three strands at present. The algorithms referenced in Section 2 bring the methods involving common principal components within relatively easy reach of the user; the distance methods of Section 3 all involve familiar concepts such as between-individual distances, principal coordinates and ADDPOINTS, all of which are available in Genstat and hence easily implementable; but the various nonlinear methods of Section 4 involve more elaborate iterative numerical schemes which require individually-tailored software, so at present are not readily accessible to the Genstat user. If that user is prepared to consider other systems, however, then many of the techniques in Gifi (1990) can be accessed readily within SPSS!

References

- Airoidi J-P and Flury B D (1988) An application of common principal component analysis to cranial morphometry of *Microtus californicus* and *M. ochrogaster* (Mammalia Rodentia). *Journal of Zoology London* 6 21-36.
- Ashton E H, Healy M J R and Lipton S (1957) The descriptive use of discriminant functions in physical anthropology. *Proceedings of the Royal Society, Series B* 146 552-572.

- Banfield J D and Raftery A E (1992) Ice floe identification in satellite images using mathematical morphology and clustering about principal curves. *Journal of the American Statistical Association* 87 7-15.
- Campbell N A and Atchley W R (1981) The geometry of canonical variate analysis. *Systematic Zoology* 30 268-280.
- Clarkson D B (1988) A least-squares version of Algorithm AS211: the F-G diagonalization algorithm, Algorithm ASR74. *Applied Statistics* 37 317-321.
- Cuadras C M (1989) Distance analysis in discrimination and classification using both continuous and categorical variables. In *Statistical Data Analysis and Inference* (Ed. Y Dodge) 459-473 North-Holland, Amsterdam.
- Cuadras C M (1991) A distance based approach to discriminant analysis and its properties *Mathematics Preprint Series No 90* Second version. University of Barcelona, Spain.
- Cuadras C M (1992) Some examples of distance based discrimination. *Biometrical Letters* 29 3-20.
- Cuadras C M and Arenas C (1990) A distance based regression model for prediction with mixed data. *Communications in Statistics - Theory and Methods* 19 2261-2279.
- Digby P G N and Kempton R A (1987) *Multivariate Analysis of Ecological Communities*. Chapman and Hall, London.
- Flury B D (1988) *Common Principal Components and Related Models*. Wiley, New York.
- Flury B D and Constantine G (1985) The F-G diagonalization algorithm, Algorithm AS211. *Applied Statistics* 34 177-183.
- Flury B D and Gautschi W (1986) An algorithm for simultaneous orthogonal transformations of several positive definite symmetric matrices to nearly diagonal form. *SIAM Journal on Scientific and Statistical Computing* 7 169-184.
- Flury B D and Neuenschwander B E (1995) Principal component models for patterned covariance matrices with applications to canonical correlation analysis of several sets of variables. In *Recent Advances in Descriptive Multivariate Analysis* (Ed. W J Krzanowski) Clarendon Press Oxford, England.
- Flury B D and Schmid M J (1992) Quadratic discriminant functions with constraints on the covariance matrices: some asymptotic results. *Journal of Multivariate Analysis* 40 244-261.
- Flury B D, Schmid M J and Narayanan A (1994) Error rates in quadratic discrimination with constraints on the covariance matrices. *Journal of Classification* 11 101-120.
- Gifi A (1990) *Nonlinear Multivariate Analysis*. Wiley, New York.
- Gnanadesikan R (1977) *Methods for Statistical Data Analysis of Multivariate Observations*. Wiley, New York.
- Gower J C (1968) Adding a point to vector diagrams in multivariate analysis. *Biometrika* 55 582-585.
- Gower J C (1971) A general coefficient of similarity and some of its properties *Biometrics* 27 857-872.
- Gower J C and Harding S A (1988) Non-linear biplots. *Biometrika*, 73 445-455.
- Hastie T and Stuetzle W (1989) Principal curves. *Journal of the American Statistical Association* 84 502-516.
- Krzanowski W J (1984) Principal component analysis in the presence of group structure. *Applied Statistics* 33 164-168.
- Krzanowski W J (1990) Between-group analysis with heterogeneous covariance matrices: the common principal component model. *Journal of Classification* 7 81-98.
- Krzanowski W J (1994) Ordination in the presence of group structure for general multivariate data. *Journal of Classification* 11. In Press.
- Meulman J J (1986) *A Distance Approach to Nonlinear Multivariate Analysis*. DSWO Press Leiden, The Netherlands.
- Meulman J J (1992) The integration of multidimensional scaling and multivariate analysis with optimal transformations. *Psychometrika* 54 539-565.
- Pearson K (1901) On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Sixth Series* 2 559-572.
- Rao C R (1982) Diversity and dissimilarity coefficients: a unified approach. *Theor. Population Biology* 21 24-43.
- Schott J R (1991) Some tests for principal component subspaces in several groups. *Biometrika* 78 177-184.
- Tyteca D and Dufrêne M (1993) On the use of distances in the taxonomic study of critical plant groups - case studies of Western European Orchidaceae. *Annals of Botany* 71 257-277.

Using Genstat to develop a taxonomic classification from ribosomal DNA sequences

G M Arnold, J A Bailey, C Sherriff and M J Whelan
 Department of Agricultural Sciences
 University of Bristol
 Institute of Arable Crops Research
 Long Ashton Research Station
 BRISTOL BS18 9AF, UK

1. Introduction

Colletotrichum is a ubiquitous pathogenic genus of fungi which infect a wide range of plants. Their taxonomy has traditionally been derived from morphological characters, such as conidia shape and size, and the identity of the host plant. This is unsatisfactory as some forms of the pathogen with identical morphologies have been isolated from many different hosts and attack different plant species. A new approach to developing a taxonomy for *Colletotrichum* has used ribosomal DNA sequencing. This is described in detail in Sherriff *et al.* (1994), where the relatedness of a range of isolates (27 in all) selected to represent the major morphological forms of the genus is studied.

2. The data

In this study, the data constitute rDNA sequences of 886 base positions in length for each of the 27 isolates. Each base is either a purine (G – guanine, A – adenine) or a pyrimidine (C – cytosine, T – thymine); thus, each isolate record comprises a textual string of length 886 where each individual unit is one of G, A, C and T. In order for comparisons to be made between isolates these individual sequences need to be aligned. This was a relatively simple process because the structure of the ribosomal gene is well established and, hence, approximate alignments were already known. The sequences were stored in a data file in the following format:

```
009                                     (3-digit isolate code number)
GCATGCCTGTTTCGAGCGTCATTTCAACCCCTCAAGCACCCTGGCGTTGGGGCTTCCACG
....                                     (13 further lines of length 60)
TTATATGCGAGTGTTTCGGGTGTCAAACCCCTACGCGTAATGAAAGT
056
GCATGCCTGTTTCGAGCGTCATTTCAACCCCTCAAGCCCTGCTTGGTGTGGGGCOCCTACG
....
TTATGTGCGAGTGTTTGGGTGTAAACCCCTACGCGTAATGAAAGT
and so on.
```

Two other characters also appear in some sequences; O denotes a base deletion which is required to maintain alignment and X a base which is present but unknown. To check the alignment, a program was written to read in the sequences from the file described above, to compare each sequence with a chosen standard and then to print all sequences in parallel showing where any differences occur. The following Genstat commands are extracted from this program:

```
SCAL nbase;886                                     "no. of bases in each sequence"
CALC line=INT(nbase/60)                             "no. of full lines (width 60)"
& last=((nbase/60)-line)*60                         "no. of bases on last line"
& n1=INT(nbase/10)                                  "no. of sets of 10 bases"
& r1=nbase-(10*n1)                                  "bases left in last set"
TEXT space;!t(' ')                                  "blank to insert after 10 bases"
SCAL i
READ [channel=2;end=*] i
READ [end=*;ch=2;layout=fix;form=1(((1)60,*)Eline,(1)Elast,*)] seq[i]
"putting in blank every 10 bases"
TEXT edit;!t('L+10 F/space/')E1,':'
EDIT [channel=edit] seq[i]

"changing label to three digits where necessary"
IF i.lt.10
PRINT [ch=t[i];iprint=*;squash=yes] '00',i;field=2,1;dec=0;skip=*,0
```

```

ELSIF (i.ge.10).and.(i.lt.100)
PRINT [ch=t[i];iprint=*;squash=yes] '0',i;field=1,2;dec=0;skip=*,0
ELSIF i.ge.100
PRINT [ch=t[i];iprint=*;squash=y] i;field=3;dec=0
ENDIF

CALC nlin=line+2                                "calculating no. of lines for output"
VARI [values=1...nlin] nline
TEXT [values=(' ')%nlin] blank

"adding labels to sequences"                    (note - within a FOR loop indexed by j)
TEXT temp[j]
PRINT [ch=temp[j];iprint=*;orient=across;width=66] seq[j]; \
      field=1;skip=0;dec=0
CONCAT [newt=temp[j]] ' ',temp[j]
CONCAT [newt=temp[j]] %t[j],temp[j]
"testing for equality of sequence with standard (here using 009)"
CALC dum=seq[%stan].eqs.seq[j]
TEXT [value=((('-')10,' ')%n1,('-')%r1] comp[j]
RESTRICT comp[j],seq[j];dum.eq.0
CONCAT [newt=comp[j]] seq[j]
RESTRICT comp[j],seq[j]
"adding label numbers to comparisons & setting up printing blocks"
PRINT [ch=temp[j];iprint=*;orient=across;width=66] comp[j]; \
      field=1;skip=0;dec=0
CONCAT [newt=temp[j]] ' ',temp[j]
CONCAT [newt=temp[j]] %t[j],temp[j]
"printing comparisons, excluding last line (blank except for label!)"
RESTRICT temp[%seqno],blank;nline.lt.nlin
PRINT [iprint=*;orient=across] temp[%seqno],blank;just=left

```

This program produces comparisons printed in the following format:

```

009 GCATGCCTGT TCGAGCGTCA TTTCAACCCT CAAGCACCGC TTGGCGTTGG GGCTTCCACG
056 -----
058 -----
060 -----
073 -----
074 --A----- --A----- -----G----- -----
076 ----- -----A----- -----
079 -----
083 -----
091 -----
093 ----- -----CT----- -----
120 --A----- --A----- -----G----- -----A-----
138 -----
141 -----
163 -----
164 -----
165 -----
167 --A----- --A----- -----G----- -----A-----
168 -----
171 -----
189 --A----- --A----- -----G----- -----A-----
216 -----
414 -----
465 -----
501 --A----- --A----- -----G----- -----A-----
503 --A----- --A----- -----G----- -----A-----
507 -----

```

The layout, with matches indicated by - and differences given explicitly by code letter, enables any minor misalignment to be picked up easily, as well as possible transcription errors which can then be checked against the original sequence gels and amended if necessary.

3. Comparing sequences

A natural measure of the distance between two isolates based on the rDNA sequences is the proportion of base positions which show changes, *p*, say. Alternative measures that have been proposed, in which evolutionary changes over time are considered such that more than one change at a given base position may have occurred, include the Jukes and Cantor (1969) distance and the Kimura (1980) distance. The Jukes and Cantor distance

is given by $-0.75 \log(1-4p/3)$ and assumes a constant rate of change from one base to another. Kimura considers different rates for transversions (changes between purines and pyrimidines) and transitions (changes within purines or pyrimidines); if the proportion of transversions and transitions are q and t , respectively, the distance is given by $-0.5 \log\{(1-2t-q)(1-2q)^{1/2}\}$. Expanding each of these gives, to first order approximation, a distance of p , identical to the simple measure. For this particular data set, the proportion p for most pairs is less than 0.10 as the sequences are highly conserved so use of this simple measure is adequate. The following extract from a Genstat program shows the steps needed to construct a similarity matrix based on this distance measure, where deletions count as differences and pairwise comparisons exclude positions where one or both bases are unknowns X.

```
"calculate pairwise similarities, storing in symmetric matrix"
TEXT slab;values=1t(1t[1seqno])
SYMM [rows=slab] seqsim
DIAG [slab;values=fb(100)] diag
CALC seqsim=diag
FOR k=1seqno;dumk=1...b
CALC dumn=seq[k].ni.1t(X)
& ncomp=SUM(dumn)
ENDFOR
FOR i=1seqno;dumi=1...b
FOR j=seqno;dumj=1...b
EXIT dumi.eq.dumj
CALC dum=seq[i].eqs.seq[j]
RESTRICT dum;(seq[i].ni.1t(X)).and.(seq[j].ni.1t(X))
CALC num=SUM(dum) "number of matches"
& ncomp=NOBS(dum) "total number of comparisons"
& %sim=100*num/ncomp
& seqsim[%dumi;%dumj]=%sim
RESTRICT dum
ENDFOR
ENDFOR
```

4. Cluster analysis

Having calculated the appropriate similarity matrix, relationships between isolates can be illustrated by a cluster analysis. Using the HCLUSTER directive with option `method=groupaverage` gives the unweighted pair group method using arithmetic means (UPGMA) with similarity between group (i,j) (formed by amalgamation of isolates i and j) and group k given by

$$s_{kij} = \frac{n_i}{n_i+n_j} s_{ik} + \frac{n_j}{n_i+n_j} s_{jk}$$

where s_{ij} are distances between groups i and j .

The clustering information may be saved using the amalgamations parameter for HCLUSTER and entered into the procedure DDENDROGRAM (here a personally customised version) to obtain a high resolution dendrogram (Figure 1). From this figure, clear groupings are apparent, some of which confirm traditional taxonomic groupings and others which have allowed new species to be identified (see Sherriff *et al.* (1994) for further details).

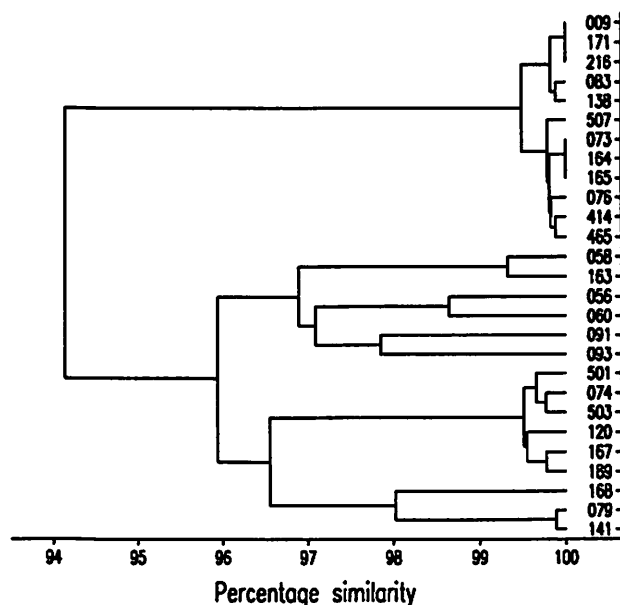


Figure 1. Dendrogram showing relatedness of 27 *Colletotrichum* isolates, from sequences of full length (886 base positions).

5. Subregion comparisons

It was also of interest to see if smaller defined subregions of the sequence could be used to obtain similar groupings. Three such regions within this sequence were ITS2 (161 bases), Domain 1 (133 bases) and Domain 2 (206 bases). Similarity matrices were constructed for the isolate sequences restricted to these regions and clustering carried out as above. Very similar patterns of groupings were apparent for both ITS2 and Domain 2. Figure 2 shows the pairwise similarities for each subregion plotted against those for the whole region, indicating that ITS2 in particular gives very similar information to the whole sequence. Correlation coefficients between the pairwise similarities for subregions and the full sequence are given in Table 1, along with those between the cophenetic distances from the equivalent dendrograms. These clearly show that the ITS2 region, and, to a slightly lesser extent, the Domain 2 region both provide information on the relatedness of the isolates which is essentially the same as from the full sequences, and can therefore provide a practically simpler and less time-consuming procedure for further similar studies.

Table 1: Correlations between measures for subregions and full sequences

Subregion	Similarities	Cophenetic distances from dendrograms
ITS2	+0.982	+0.993
Domain 1	+0.712	+0.671
Domain 2	+0.969	+0.985

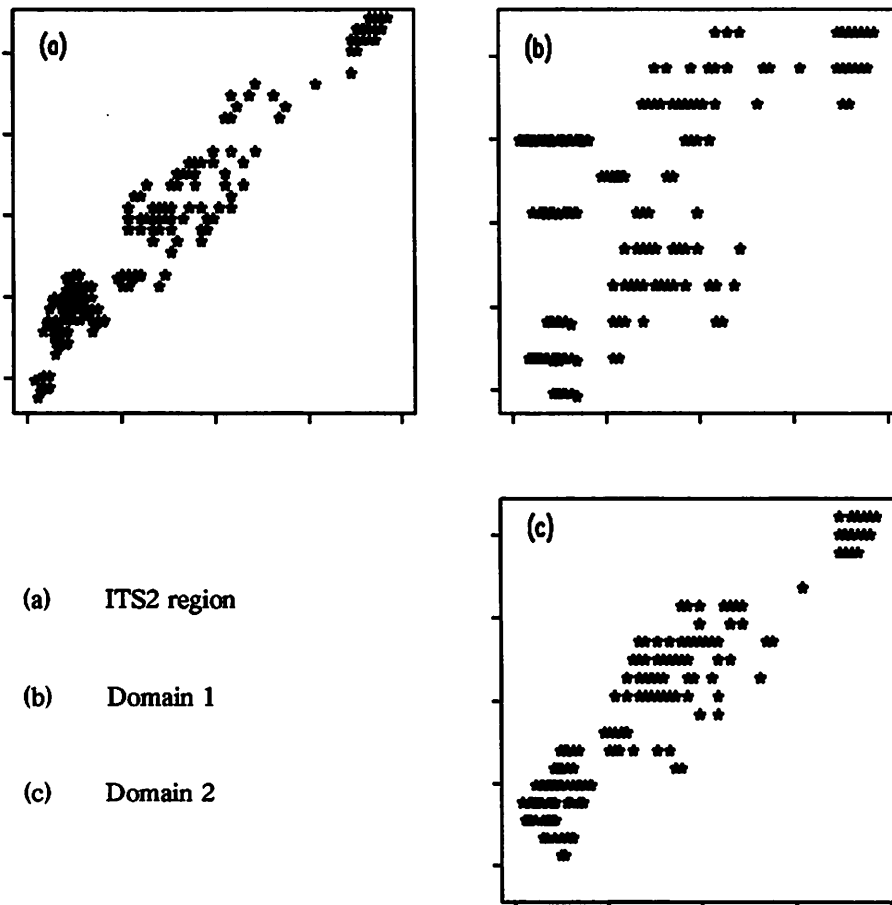


Figure 2: Plots of pairwise similarities for subregions (a) – (c) against full sequence

6. Conclusions

Genstat has proved to be a flexible tool for handling text manipulations, calculations and visual displays required in using rDNA sequence data for taxonomic investigations and gave results similar to specifically written software for sequence analysis (see Sherriff *et al.*, 1994).

Acknowledgement

Part of this work was funded by the Overseas Development Administration through a commission (X0090) by the Natural Resources Institute.

References

- Jukes T H and Cantor C R (1969) Evolution of protein molecules. In: *Mammalian protein metabolism* (Ed. H N Munro) Academic Press, New York, 21-132.
- Kimura M (1980) A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.* 16 111-120
- Sherriff C, Whelan M J, Arnold G M, Lafay J-F, Brygoo I and Bailey J A (1994) Ribosomal DNA sequence analysis reveals new species groupings in the genus *Colletotrichum*. *Experimental Mycology* 18 121-138.

The jackknife and the bootstrap

P W Lane and R W Payne

BBSRC IACR Rothamsted Experimental Station

HARPENDEN, Herts AL5 2JQ, UK

Summary

Bootstrapping is a general method for estimating properties of statistics, particularly their precision. It is computer intensive, but is increasingly being considered as an alternative to classical methods. In particular, it avoids distributional assumptions, and is applicable when the complexity of a model makes analytical methods intractable. Jackknifing is a similar but simpler technique. Procedures have been developed to provide a general facility to form bootstrap or jackknife estimates for any statistic that can be calculated in Genstat.

1. The jackknife

The jackknife was introduced by Quenouille (1949) to reduce bias in estimation. The name was coined by Tukey (1958), who suggested its use for estimation of variance. Its main properties are:

- it removes bias of order $1/N$ for estimation from a sample of size N ;
- it can form standard errors of estimates even when these are difficult to form by other methods;
- it is often a more robust method than alternative classical methods;
- it needs a lot of computation: of the order of $(N+1)$ times what is required for simple estimation;
- it has proved useful in many situations, such as maximum-likelihood estimation and variance estimation, but is not useful in others, such as the estimation of order statistics.

The jackknife works by repeating some estimation process based on a sample, leaving out each observation in the sample in turn. Suppose that we want to estimate parameter θ from a sample X_1, \dots, X_N by some method, possibly biased, yielding an estimate $\hat{\theta}$. Then we repeat the estimation N times by the same method, omitting each of the observations X_i in turn, yielding estimates $\hat{\theta}_j, j=1, \dots, N$. These estimates are combined to form the *pseudo-values*:

$$\theta_j^* = N\hat{\theta} - (N-1)\hat{\theta}_j, \quad j=1 \dots N$$

From the pseudo-values, the jackknife estimates themselves can be formed:

$$\begin{aligned} \theta^* &= \text{Mean}(\theta_j^*) \\ V^* &= \text{Variance}(\theta_j^*)/N \end{aligned}$$

The original papers describing the jackknife technique are by Quenouille (1949, 1956) and by Tukey (1958). Good expository accounts are provided by Hinkley (1983) or Bissell and Ferguson (1975).

The calculations can be carried out in Genstat using the procedure **JACKKNIFE**. It requires as input a data matrix consisting of a list of variates, factors and texts all of the same length. Each unit of these vectors will be omitted in turn during the calculations. The procedure can arrange to calculate several statistics simultaneously, and produce jackknife estimates for all of them. You need to supply a procedure called **RESAMPLE** that calculates the statistics, based on a data matrix reduced by one unit. In addition, you can supply further data to the procedure if required to calculate the statistics, using the **ANCILLARY** option. The procedure produces as output the jackknife mean and standard for each statistic, and you can extract the pseudo-values.

For example, consider the estimation of the correlation coefficient. This statistics can be calculated easily in Genstat by the **CORRELATE** directive, but there is no estimation of the variance of the estimate. Here is a procedure in the form required by **JACKKNIFE** that calculates the correlation.

```
PROCEDURE [PARAMETER=pointer] 'RESAMPLE'
OPTION 'DATA', " (I: variates, factors or texts) data vectors from which to
                calculate the statistics; no default"\
'AUXILIARY', " (I: pointers) auxiliary sets of data vectors, each of which is
```

```

                to be resampled independently"\
'ANCILLARY'; " (I: any type of structure) other relevant information needed to
                calculate the statistics"\
MODE=p; TYPE=!t(ariate,factor,text),'pointer',*; SET=yes,no,no; LIST=yes;\
DECLARED=yes; PRESENT=yes
PARAMETER 'STATISTIC', " (O: scalars) to save the calculated statistics "\
'EXIT'; " (O: scalars) to save an exit code to indicate failure (EXIT[i]=1)
                or success (EXIT[i]=0) when calculating each STATISTIC[i]"\
MODE=p; TYPE='scalar'; SET=yes

CALCULATE STATISTIC[1] = CORRELATION(DATA[1]; DATA[2])
&          EXIT[1] = STATISTIC[1]==C('missing')

ENDPROCEDURE

```

The `OPTION` and `PARAMETER` statements here can be copied from the standard example that accompanies the `JACKKNIFE` procedure: the syntax must not be changed. The `AUXILIARY` parameter is not used by `JACKKNIFE`, but is included in the procedure because it can be used with the `BOOTSTRAP` procedure described below. The `EXIT` parameter of the `RESAMPLE` procedure provides the ability to signal to the `JACKKNIFE` procedure when the calculation of the statistic fails for some reason with a particular resampling of the units; if this is not relevant, the parameter does not need to be set.

Here is the result of using the procedure, using an example from Efron (1981).

```

22 VARIATE [VALUES=576,635,558,578,666,580,555,661, \
23 651,605,653,575,545,572,594] Y
24 & [VALUES=3.39,3.30,2.81,3.03,3.44,3.07,3.00,3.43, \
25 3.36,3.13,3.12,2.74,2.76,2.88,2.96] Z
26 JACKKNIFE [DATA=Y,Z] 'Correlation'

```

***** Jackknife estimates *****

Statistic	Estimate from all data	Jackknife estimate	s.e.
Correlation	0.7764	0.7828	0.1425

Several modifications to the jackknife have been suggested. The second-order and generalized jackknife techniques are designed to remove bias of higher order than $1/N$. The *infinitesimal jackknife* uses small weights for points rather than total exclusion, and the *trimmed jackknife* uses the trimmed mean of the pseudo-values rather than the simple mean. Alternative methods have also been proposed using subsets smaller than $(N-1)$. None of these modifications are available in the `JACKKNIFE` procedure, but it should not be difficult to edit the procedure to incorporate any of them.

2. The bootstrap

The bootstrap was introduced by Efron (1979) to provide variance estimation. It has the following properties:

- it estimates bias;
- it forms standard errors of estimates even when these are difficult to form by other methods;
- it estimates the distribution of estimates, giving confidence intervals;
- it is non-parametric and robust;
- it needs a lot of computation: 100 times what is required for simple estimation, to get standard errors, or 1000 times to get confidence intervals;
- it seems more widely applicable than the jackknife.

The bootstrap method works by repeated *resampling* from the units of a data matrix, generating a series of new data matrices from which estimates of means and variance can be calculated. Resampling here means making a new sample of the same size N as the original sample, by random sampling with replacement from the original sample. So each new sample contains some of the original units of data, but is unlikely to contain all of them; several of the original units are likely to be repeated in the new sample.

The statistics to be bootstrapped are calculated for each resampled data matrix, and the bootstrap estimates are then formed by calculating the mean of these statistics. Other distributional features of the statistics, such as standard errors or confidence regions, can be estimated from the empirical distribution of the set of calculated statistics.

The name *bootstrap* is seen to be an apt description of what is happening in this process: the distributional properties of a statistic are derived from the data themselves, without reference to any theoretical model, just as a magician may attempt to raise himself off the floor by pulling on his own bootstraps. A good introduction to the bootstrap is given by Efron and Tibshirani (1986); a fuller treatment can be found in Efron and Tibshirani (1993).

To understand the justification of this as a process of estimation, it helps to consider a simple application: the estimation of the mean of a sample of measurements x_1, \dots, x_N . If each x_i has some unknown distribution F , the mean \bar{x} has standard error

$$\sigma = \sqrt{(\mu_2(F)/N)}$$

where $\mu_2(F)$ is the second moment of F . If we do not know F , we do not know $\mu_2(F)$. The classical solution to this problem is to estimate σ by

$$\bar{\sigma} = \sqrt{(\bar{\mu}_2/N)}$$

where $\bar{\mu}_2$ is an unbiased estimate of $\mu_2(F)$, such as $\Sigma(x_i - \bar{x})^2 / (N-1)$ if the x_i are Normally distributed. The bootstrap solution to the problem is to estimate σ by

$$\hat{\sigma} = \sigma(\hat{F}) = \sqrt{(\mu_2(\hat{F})/N)}$$

where \hat{F} is an estimate of F , such as the empirical probability distribution of x_i .

The bootstrap calculations can be carried out in Genstat using the `BOOTSTRAP` procedure. It is used in just the same way as `JACKKNIFE`, setting the `DATA` parameter to supply the data matrix, and providing a `RESAMPLE` procedure in exactly the same form as for `JACKKNIFE`. Here is the result of bootstrapping the correlation coefficient in the example above, using the default of 100 resamplings.

```

33 BOOTSTRAP [DATA=Y,Z; SEED=77320] 'Correlation'
*** Bootstrap estimates, from 100 bootstrap samples ***

```

Label	mean	s.e.	95% confidence interval	
			lower	upper
Correlation	0.754	0.154	0.452	0.970

The output includes a confidence interval, by default at the 95% level, derived from the empirical distribution of the 100 generated estimates of correlation. However, most reports about the behaviour of bootstrap estimates suggest that 100 resamplings are not enough to get reliable estimates of such confidence intervals. Here is a repeat of the bootstrapping, carried out 1000 times, and setting the `PRINT` option of the directive to display the distribution of the generated estimates.

```

37 BOOTSTRAP [PRINT=estimates,graph; NTIMES=1000; DATA=Y,Z; SEED=77320] \
38 'Correlation'
*** Bootstrap estimates, from 1000 bootstrap samples ***

```

Label	mean	s.e.	95% confidence interval	
			lower	upper
Correlation	0.774	0.134	0.475	0.966

The resulting picture, produced in high-resolution by default, is in Figure 1. It shows a histogram of the 1000 generated estimates of correlation, with a smoothed curve superimposed in an attempt to improve the distributional shape. The curve is produced by fitting a smoothing spline with four degrees of freedom (using the `SSPLINE` function in the `FIT` directive) to the cumulated histogram values on a logistic scale. The smoothing is not completely successful here because of the limit of 1.0 on the estimates of correlation. Vertical lines are also displayed on the graph to indicate the bootstrap estimate and the confidence interval.

Three methods of bootstrapping are provided. By default, resampling is completely pseudo-random, using Genstat's random-number generator. The generator can be initialized by setting option SEED, thereby producing reproducible results; otherwise, the initialization uses the system clock. A second alternative is balanced bootstrapping, requested by setting METHOD=balance. In this case, the resampling is constrained to ensure that each unit of the data matrix occurs the same number of times in the complete set of generated samples. The third method, specified by METHOD=permute, is simply to permute the units of the data matrix. Note that this method gives no variation in results if the statistics are independent of the order of the data, like the sample mean. However, this method provides permutation tests, a type of randomization test that can be applied to grouped data.

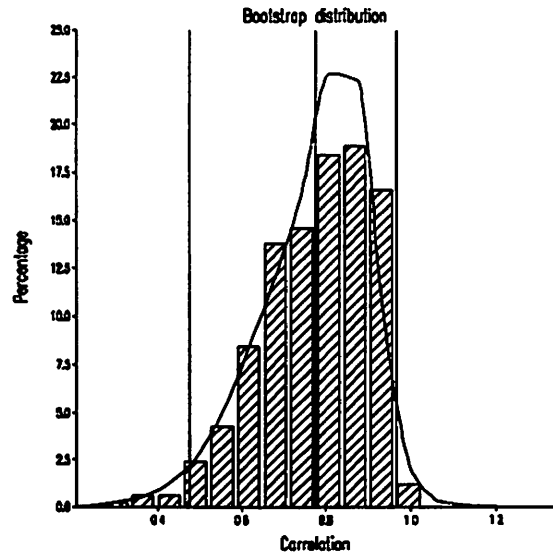


Figure 1

3. Availability of the procedures

The BOOTSTRAP and JACKKNIFE procedures have both been accepted for inclusion in the Genstat Procedure Library 3[2]. In the meantime, the procedures can be accessed from the NAG Gopher. Connection details for this Gopher were published in *Genstat Newsletter* 30, but most Gopher servers make it easy to find, as long as you know that NAG is based in the United Kingdom.

Both procedures have been written for Release 3.1 and use several of the new features, such as the new DUPLICATE directive. It would therefore require some effort to translate them to work with Release 2.2.

References

- Bissell A F and Ferguson R A (1975) The jackknife – toy, tool or two-edged weapon. *The Statistician* 24 79-100.
- Efron B (1979) Computers and the theory of statistics: thinking the unthinkable. *SIAM Review* 21 460-480.
- Efron B (1981) *The jackknife, the bootstrap and other resampling plans*. CBMS Monograph 38, SIAM, Philadelphia.
- Efron B and Tibshirani R (1986) Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science* 1 54-77.
- Efron B and Tibshirani R J (1993) *An introduction to the bootstrap*. Chapman and Hall, London.
- Hinkley D (1983) Jackknife methods. In: *Encyclopedia of statistics*, Volume 4 (Ed. Kotz, N L Johnson and C B Read) Wiley, New York.
- Quenouille M H (1949) Approximate tests of correlation in time series. *Journal of the Royal Statistical Society B* 11 18-44.
- Quenouille M H (1956) Notes on bias in estimation. *Biometrika* 61 353-360.
- Tukey J W (1958) Bias and confidence in not quite large samples (Abstract). *Annals of Mathematical Statistics* 29 614.

Efficiency factors for some balanced hyper-Graeco-Latin superimpositions of Youden squares

E D Gardiner

Department of Pure Mathematics and Statistics

The University

Cottingham Road

HULL HU6 7RX, UK

D A Preece

Institute of Mathematics and Statistics

Cornwallis Building

The University

CANTERBURY

Kent CT2 7NF, UK

The past 30 years have seen the publication of some surprising results on the efficiency factors for treatment factors from non-orthogonal experimental designs. Some of these results were discussed and exemplified by Preece (1988). The present paper takes that earlier discussion further for a class of generally balanced designs that are obtainable from certain balanced superimpositions of Youden squares.

To introduce the subject, let us consider first the following 4×7 row-and-column design for two non-interacting sets of treatments:

AA	BB	CC	DD	EE	FF	GG	
BC	CD	DE	EF	FG	GA	AB	
CE	DF	EG	FA	GB	AC	BD	(1)
EB	FC	GD	AE	BF	CG	DA	

In each cell of this design, the first letter represents a treatment from a set T1, whereas the second represents a treatment from a set T2. The design comprises 4 rows from a systematically generated 7×7 Graeco-Latin square. The four rows are chosen so that each of T1 and T2 is balanced with respect to the columns of the design, the concept of 'balance' here being that of a balanced incomplete block design; as the number of treatments in T1 or T2 is the same as the number of columns in (1), each of T1 and T2 is disposed in (1) in a Youden square. The superimposition of each Youden square on the other is such that (i) each of T1 and T2 is balanced with respect to the other, in the same sense of 'balance' as has already been used, and (ii) the design is generally balanced overall, as can be verified by submitting it to Genstat's ANOVA. By 'overall', we here mean that (a) all estimated differences in effect between two treatments from T2 have the same efficiency factor when the effects of the other factors in the design (whether block factors or treatment factors) have been eliminated, and (b) the corresponding result is true for T1. More concisely, we are saying that there is a single efficiency factor for T1 and a single efficiency factor for T2 when treatment effects are estimated after elimination of the effects of all other factors in the design. Indeed, if we now change the role of the 'columns' factor to that of a treatment factor T0, there is similarly a single efficiency factor for T0. Indeed these efficiency factors for T0, T1 and T2 are all the same for this design, namely $3/4 = 0.75$. Of course, this value is unchanged if the 'rows' factor of the design is ignored, as each of T0, T1 and T2 is orthogonal to rows.

Suppose now that the factor T2 in (1) is replaced by a factor T3 to give the following superimposition of two 4×7 Youden squares:

AA	BB	CC	DD	EE	FF	GG	
BD	CE	DF	EG	FA	GB	AC	
CG	DA	EB	FC	GD	AE	BF	(2)
EF	FG	GA	AB	BC	CD	DE	

Design (2) is in every respect as balanced as design (1), and the efficiency factors for T0, T1 and T3, calculated as before, are again equal to one another. But now the value of this common efficiency factor is $5/8 = 0.625$.

Preece (1966) described how designs such as (1) and (2) can be obtained more generally, consisting of 2 superimposed Youden squares of size $2p \times (4p-1)$ or $(2p-1) \times (4p-1)$, where $(4p-1)$ is a prime number with $p > 1$. For each of these sizes, the superimpositions have one or other of just two efficiency factors.

Suppose now that we consider the following superimposition of *three* Youden squares, with non-interacting factors T0, T1, T2, and T3 as above:

AAA	BBB	CCC	DDD	EEE	FFF	GGG	
BCD	CDE	DEF	EFG	FGA	GAB	ABC	
CEG	DFA	EGB	FAC	GBD	ACE	BDF	(3)
EBF	FCG	GDA	AEB	BFC	CGD	DAE	

As stated by Preece (1968), the efficiency factors for each of T0, T1, T2 and T3, after fitting all the other factors, are no longer all the same, but are these:

T0 and T3	T1 and T2
$\frac{7}{24} = 0.292$	$\frac{7}{20} = 0.350$

Preece (1966) also indicated that just one further similar type of 4×7 design exists; it can be obtained from (3) by replacing set T3 by a set T4 as follows:

AAA	BBB	CCC	DDD	EEE	FFF	GGG	
BCE	CDF	DEG	EFA	FGB	GAC	ABD	
CEB	DFC	EGD	FAE	GBF	ACG	BDA	(4)
EBC	FCD	GDE	AEF	BFG	CGA	DAB	

For (4), the efficiency factors analogous to those given above for (3) are different from those for (3) and follow a different pattern:

T0	T1, T2 and T4
$\frac{7}{10} = 0.700$	$\frac{7}{12} = 0.583$

The results just given for balanced superimpositions of *three* Youden squares of size 4×7 may seem strange enough. But what about analogous balanced superimpositions of three Youden squares of any size $2p \times (4p-1)$ or $(2p-1) \times (4p-1)$ for which $(4p-1)$ is a prime number with $p > 1$? We can now reveal that there

are, in general, for each size, exactly three possibilities, not just the two that are represented above by (3) and (4). The third of these possibilities has no representative of size 4×7 , as this size is too small to admit of the required combinatorial flexibility. We shall now illustrate the three possibilities by turning to the size 6×11 . As hitherto, we shall use the phrase 'efficiency factor' only in reference to efficiency calculated for a factor after fitting effects for all other factors.

Consider first the following design, analogous to design (3):

AAA	BBB	CCC	DDD	EEE	FFF	IKD	HHH	III	JJJ	KKK	
CEI	DFJ	EGK	FHA	GIB	HJC	DAF	JAE	KBF	ACG	BDH	
IFK	JGA	KHB	AIC	BJD	CKE	FEC	EBG	FCH	GDI	HEJ	
KJH	AKI	BAJ	CBK	DCA	EDB	CJB	GFD	HGE	IHF	JIG	(5)
HDG	IEH	JFI	KGJ	AHK	BIA	BHI	DKC	EAD	FBE	GCF	
GBC	HCD	IDE	JEF	KFG	AGH	GGG	CIJ	DJK	EKA	FAB	

As in all other designs in this paper, the entries in each row are generated cyclically from the entry in the first column, the cycle being (ABC ...) with $(4p-1)$ letters in the brackets. Once again, we shall use T0 for the 'blocks' factor. However, it will now be convenient to generalise our previous labelling of other treatment factors by basing a factor's labelling on the treatment that appears for that factor in column 1, row 2. In (5), we therefore base our remaining labelling on the treatments C, E and I of, respectively, the remaining three factors. If we code A,B,C,..., as 0, 1, 2,..., these treatments become 2, 4 and 8, so we denote the corresponding factors as T2, T4 and T8. The efficiency factors are then as follows:

T0 and T4	T2 and T8
$\frac{11}{18} = 0.611$	$\frac{55}{84} = 0.655$

Now consider the following design that is analogous to design (4):

AAA	BBB	CCC	DDD	EEE	FFF	GGG	HHH	III	JJJ	KKK	
CEK	DFA	EGB	FHC	GID	HJE	IKF	JAG	KBH	ACI	BDJ	
IFH	JGI	KHJ	AIK	BJA	CKB	DAC	EBD	FCE	GDF	HEG	
KJG	AKH	BAI	CBJ	DCK	EDA	FEB	GFC	HGD	IHE	JIF	(6)
HDC	IED	JFE	KGF	AHG	BIH	CJI	DKJ	EAK	FBA	GCB	
GBI	HCJ	IDK	JEA	KFB	AGC	BHD	CIE	DJF	EKG	FAH	

Here, with labelling as before, we have the respective factors T0, T2, T4 and T10, with efficiency factors as follows:

T10	T0, T2 and T4
$\frac{11}{14} = 0.786$	$\frac{11}{15} = 0.733$

But finally, amongst designs of size 6×11 , consider the following design:

AAA	BBB	CCC	DDD	EEE	FFF	GGG	HHH	III	JJJ	KKK	
CIK	DJA	EKB	FAC	GBD	HCE	IDF	JEG	KFH	AGI	BHJ	
IKH	JAI	KBJ	ACK	BDA	CEB	DFC	EGD	FHE	GIF	HJG	
KHG	AIH	BJI	CKJ	DAK	EBA	FCB	GDC	HED	IFE	JGF	(7)
HGC	IHD	JIE	KJF	AKG	BAH	CBI	DCJ	EDK	FEA	GFB	
GCI	HDJ	IEK	JFA	KGB	AHC	BID	CJE	DKF	EAG	FBH	

Here we have efficiency factors as follows:

T0, T2, T8 and T10

$$\frac{11}{15} = 0.733$$

Now we are back to the situation that we had for the balanced superimposition of just two Youden squares, namely that of a design having the same efficiency factor for each of the treatment factors.

Ignoring the 'rows' factor, our design (7) is Design 7 from p.29 of Potthoff (1963). We have thus thrown light on Potthoff's somewhat cryptic assertion that The efficiency of Design 7 is 11/15. We have, however, also shown that, in practice, our design (6) might well be preferable to (7), as design (6) provides greater efficiency for one of the factors.

The numerical values for the efficiency factors for designs (5), (6) and (7) are obtainable by taking $p = 3$ in the following general formulae for balanced superimpositions of three Youden squares of size $2p \times (4p-1)$ where $(4p-1)$ is prime:

$$1 - \frac{1}{2p} - \frac{p-1}{2p(2p+1)} = \frac{11}{14} = 0.786;$$

$$1 - \frac{1}{2p} - \frac{1}{2(2p-1)} = \frac{11}{15} = 0.733;$$

$$1 - \frac{1}{2p} - \frac{p+2}{2(2p+1)(p-1)} = \frac{55}{84} = 0.655;$$

$$1 - \frac{1}{2p} - \frac{p^2+4p-1}{2p^2(2p-1)} = \frac{11}{18} = 0.611.$$

The reader is, however, invited to obtain the numerical values by running designs (5), (6) and (7) through Genstat's ANOVA. The only factors that need to be coded are T0, T2, T4, T8 and T10. Then, for example, the efficiency factor of 0.786 for T10 in (6) can be obtained by specifying the TREATMENTSTRUCTURE for (6) as T0+T2+T4+T10.

Each of the designs (5), (6) and (7) remains a balanced superimposition of Youden squares if its first row is deleted, to give a 5×11 superimposition. If we designate the resultant designs as (5'), (6') and (7'), we have efficiency factors as follows:

(5')	T0 and T4	T2 and T8
	0.629	0.550
(6')	T10	T0, T2 and T4
	0.550	0.629

(7') T0, T2, T8 and T10

0.314

Potthoff (1963) alluded to the existence of design (7'), and indeed gave its efficiency as $11/35 = 0.314$. But we can now see that (6') is not only more efficient than (7') for all 4 factors, but indeed *twice* as efficient for 3 of the 4 factors.

By this stage in our paper, our reader will wish to know how designs such as (5), (6) and (7) are to be distinguished from one another so that the pattern of the efficiency factors can be deduced. Elementary, my dear Watson! – so long as we look carefully at some incidence matrices. For this, only one matrix n need be written down explicitly. For 6×11 designs, n is as follows, where the rows and columns have been numbered and labelled for convenience, and where zero entries have been represented by dots:

	1	2	3	4	5	6	7	8	9	10	11
	A	B	C	D	E	F	G	H	I	J	K
1A	1	1	.	1	1	1	.	.	.	1	.
2B	.	1	1	.	1	1	1	.	.	.	1
3C	1	.	1	1	.	1	1	1	.	.	.
4D	.	1	.	1	1	.	1	1	1	.	.
5E	.	.	1	.	1	1	.	1	1	1	.
$n =$ 6F	.	.	.	1	.	1	1	.	1	1	1
7G	1	.	.	.	1	.	1	1	.	1	1
8H	1	1	.	.	.	1	.	1	1	.	1
9I	1	1	1	.	.	.	1	.	1	1	.
10J	.	1	1	1	.	.	.	1	.	1	1
11K	1	.	1	1	1	.	.	.	1	.	1

This matrix satisfies the equation

$$n = I + \sum_{i \in Q} \Gamma_i \tag{8}$$

where I is the 11×11 identity matrix, Γ_i is the basic 11×11 circulant matrix with the entry 1 in the first position of column $(i + 1)$ and the entry 0 elsewhere in row 1, and Q is the set of quadratic residues in $GF(11)$, the Galois field of order 11, i.e.

$$\begin{aligned} Q &= \{ 2^0, 2^2, 2^4, 2^6, 2^8 \} \pmod{11} \\ &= \{ 1, 4, 5, 9, 3 \}. \end{aligned}$$

As the elements of T2 that appear in column 1 of (5), (6) or (7) are A, C, I, K, H, G, and the non-zero elements in column 1 of n are in the rows labelled A, C, I, K, H, G, we readily see that n is the incidence matrix for the incidence of T2 on columns of our designs, i.e. for the incidence of T2 with respect to T0. Writing n_{ij} for the incidence matrix for the incidence of T_i with respect to T_j , we thus have

$$n = n_{20}.$$

Proceeding similarly for other pairs of factors (with 'rows' ignored as hitherto), we therefore have the following for designs (5), (6) and (7):

$$(5): \quad n = n_{20} = n_{04} = n_{80} = n_{42} = n_{82} = n_{48}$$

$$(6): \quad n = n_{20} = n_{04} = n_{t0} = n_{42} = n_{t2} = n_{t4}$$

$$(7): \quad n = n_{20} = n_{80} = n_{t0} = n_{82} = n_{t2} = n_{t8}$$

where the suffix t denotes 10.

If we now examine the incidence-matrix equations for (5), we find that they contain two subsets of equations that can be written in the cyclic form

$$n_{ij} = n_{jk} = n_{ki} \tag{9}$$

namely

$$n_{04} = n_{42} = n_{20}$$

and

$$n_{04} = n_{48} = n_{80}$$

Common to these two subsets is the matrix n_{04} , relating to the factors T0 and T4, so it is not surprising that the efficiency factor for T0 and T4 in (5) is different from that for T2 and T8.

Turning now to the incidence-matrix equations for (6), we find only one cyclic subset of equations of the form (9), namely

$$n_{04} = n_{42} = n_{20}$$

The matrices in this subset relate to the factors T0, T2 and T4, but not to the other factor in (6), namely T10. So we need not be surprised that the efficiency factor for T10 differs from that for T0, T2 and T4.

The incidence-matrix equations for (7) contain no subsets of the form (9). This is the condition for there to be just a single efficiency factor for all treatment factors in the design.

For balanced superimpositions of three Youden squares of any size $2p \times (4p - 1)$ where $(4p-1)$ is prime, equation (8) needs to be generalised to

$$n = I + \sum_{i \in Q} \Gamma_i \tag{10}$$

where I is the $v \times v$ identity matrix, Γ_i is the basic $v \times v$ circulant matrix defined as before, and Q is the set of quadratic residues in $GF(v)$. The simplicity of the general formulae given earlier for efficiencies for the size $2p \times (4p-1)$ derives from the properties of the circulant matrices and the restriction of the summation to the set Q . For the size $(2p - 1) \times (4p - 1)$, similar mathematical results may be obtained by using the matrix

$$n = \sum_{i \in Q} \Gamma_i$$

As a final exercise, the reader is invited to use the 7×7 matrix n given by (10) to distinguish design (3) from design (4). For this example we have

$$\begin{aligned} Q &= \{ 3^0, 3^2, 3^4 \} \pmod{7} \\ &= \{ 1, 2, 4 \}. \end{aligned}$$

References

- Potthoff R J (1963) Some illustrations of 4DIB design constructions. *Calcutta Statist. Assoc. Bull.*, 12 19-30.
 Preece D A (1966) Some row and column designs for two sets of treatments. *Biometrics*, 22 1-25.
 Preece D A (1988) Genstat analyses for complex balanced designs with non-interacting factors *Genstat Newsletter* 21 33-45.

Solving the depletion equation; an example of inverse nonlinear regression

P Brain and L R Saker
 Department of Agricultural Sciences
 University of Bristol
 Institute of Arable Crops Research
 Long Ashton Research Station
 BRISTOL BS18 9AF, UK

1. Introduction

Regression problems can arise where the dependent variable, x , is a function of the independent variable, y , so that $x = f(y; \theta)$, where θ is a vector of unknown parameters to be estimated; an example of this was presented by Ridout (1993). This article deals with a specific example used in the analysis of experiments which measure the depletion of a radioactive nutrient by a plant, and was first presented at the 1993 Genstat Conference, Canterbury, Kent (Brain and Saker, 1993). The depletion technique was discussed by Claassen and Barber (1974), who presented a theoretically derived equation which related the nutrient concentration in the nutrient media (C) to time (t). Their equation related the rate of uptake (equivalent to the decay rate of the concentration) to the concentration in the external solution, and was of the form

$$\frac{dC}{dt} = -\frac{W}{V} \left(\frac{I_m C}{K_m + C} - E \right)$$

where W is the root weight (known), V is the volume of the solution (known), E is the efflux, I_m is the maximum rate of uptake, and K_m is the Michaelis constant. They then fitted this equation to their experimental results by solving the equation by numerical integration, then fitted to the data using least squares. This method was used by several authors, including Drew *et al* (1984), who used a simplified form of the original equation with $E=0$, but included a background level of concentration C_{\min} below which the concentration could not fall. Their equation is equivalent to the original equation but with E reparameterised in terms of C_{\min} . A further development of the procedure was introduced by McLachlan *et al* (1987), who considered the case when the original equation was inappropriate, and considered ways of investigating the relationship between dC/dt and C by fitting various empirical curves to the C versus t time courses.

None of these authors apparently recognised that the basic differential equation can be readily solved to produce a relationship between C and t ; in this paper we solve the equation and develop a method for fitting it to experimental data. The approach does not rely on numerical integration, as in previous approaches, and can be readily implemented in Genstat.

2. The model

As noted above Drew *et al* (1984) used a modified version of the equation used by Claassen and Barber (1974) by assuming that E was zero, but the depletion was dependent on the concentration above a natural background level, C_{\min} . Their equation was

$$\frac{dC}{dt} = -\frac{W}{V} \frac{I_m (C - C_{\min})}{K_m + (C - C_{\min})}$$

This can be rearranged to give

$$dt = -\frac{V}{W I_m} \left(\frac{K_m}{C - C_{\min}} + 1 \right) dC$$

which can be readily solved to give

$$t = \frac{V}{W I_m} \left[K_m \ln \left(\frac{C_0 - C_{\min}}{C - C_{\min}} \right) + (C_0 - C) \right] \quad (1)$$

where C_0 is the initial concentration. It is impossible to rearrange this equation to evaluate C given t . This equation is a more complex version of that derived by Ridout (1993) which modelled plant growth.

Under certain conditions the equation will degenerate to simpler forms; for example when K_m is very large, but $(W I_m)/(V K_m)$ is finite, then $1/K_m$ is close to zero, and equation (1) reduces to

$$C = C_{\min} + (C_0 - C_{\min}) e^{-(W I_m t / V K_m)} \quad (2)$$

i.e. the concentration decays exponentially with time. This equation is one of those considered by McLachlan *et al* (1987). Another special case occurs when $K_m = 0$, but $(W I_m)/V$ is non-zero, when equation (1) reduces to a straight line:

$$C = C_0 - \frac{W I_m t}{V} \quad (3)$$

3. Fitting the model

Equation (1) can be solved for known values of the parameters to give C for a given t using the iterative Newton–Raphson method. This can be incorporated into a series of CALCULATE commands, which can be used with the FITNONLINEAR command, in a similar way to that used by Ridout (1993). (Though as Ridout comments, there appears to be no way that a test for convergence can be carried out in a series of CALCULATES of this form.) However, when this approach was tried using equation (1) the iterative process was unstable, and could not be relied on to converge. Accordingly equation (1) was rewritten in a more suitable form:

$$(C - C_{\min}) e^{-(C_0 - C)/K_m} = (C_0 - C_{\min}) e^{-W I_m t / V K_m} \quad (4)$$

As K_m becomes increasingly large, but $(W I_m)/(V K_m)$ is constant, equation (4) reduces directly to equation (2). The case where K_m is close to zero is not covered directly by this equation but can be readily fitted using linear regression. Equation (4) has six parameters, of which only four are identifiable. However W and V are measured independently, so all parameters can be estimated. For estimation purposes $(W I_m)/(V K_m)$ was denoted by T , and $1/K_m$ was replaced by I_{K_m} . With this parameterisation equation (4) becomes

$$(C - C_{\min}) e^{-(C_0 - C) I_{K_m}} = (C_0 - C_{\min}) e^{-T t} \quad (5)$$

Equation (4) can be rewritten in the form

$$f(C) = 0$$

and the Newton–Raphson method can be applied to this to give

$$C_{i+1} = C_i + \Delta C$$

where

$$\Delta C = \frac{(C_0 - C_{\min}) e^{(C_0 - C) I_{K_m}} - T t - (C_i - C_{\min})}{1 + (C_i - C_{\min}) I_{K_m}} \quad (6)$$

This iterative sequence was set up as a series of CALCULATE statements in Genstat which were used with FITNONLINEAR. The first evaluates the first Δc (denoted by dc) using equation (6), with our initial estimates C_{\min} , T , I_{K_m} and C_0 , and estimating the fitted concentrations, C , by our observed values of c . (For reasonable data the observed concentrations are an obvious choice, and in most cases that have been tried out work well.)

The second expression updates our fitted values, fc , and our fitted values of concentration, c . Expressions E[3,5...17] and E[4,6...18] repeat this iteration, but using the fitted concentrations, c . The final expression transforms the fitted values, if necessary, so that a transform-both-sides analysis (Rudemo *et al.*, 1989) can be carried out if required (if a transform was required the MODEL statement would obviously use the transformed concentrations). The fitted concentrations are stored in C and can be used later if required. The system as set up uses several iterations; the actual number used can be altered by using more, or less, expressions. The final increment dc is available at the end of the iterative process so can be readily inspected. The CALCULATE statements are stored in the expressions given below:

```

express [value = (dc=((C0-Cmin)*exp((C0-c)*IKm-T*t)-(c-Cmin))/ \
                (1+(c-Cmin)*IKm))] E[1]
      & [value = (fc=(C=c+dc))] E[2]
      & [value = (dc=((C0-Cmin)*exp((C0-C)*IKm-T*t)-(C-Cmin))/ \
                (1+(C-Cmin)*IKm))] E[3,5...17]
      & [value = (fc=(C=C+dc))] E[4,6...18]
      & [value = (fc=log(fc))] E[19]

```

Various special cases of the equation can be readily fitted; of particular relevance are the cases where C_{\min} is zero, or when K_m is large (corresponding to $1/K_m = 0$). The increase in the residual sum of squares can then be used to test whether C_{\min} , or $1/K_m$, are significantly different from zero.

4. Finding initial parameter estimates

The fitting process outlined above needs good initial estimates of the parameters C_0 , C_{\min} , T , and $1/K_m$. An obvious initial estimate of C_0 is the observed concentration at time zero; a reasonable initial estimate of C_{\min} is often zero. Given these initial estimates two new variates x_1 and x_2 can be calculated where

$$x_1 = \ln \left(\frac{C_0 - C_{\min}}{C - C_{\min}} \right)$$

and

$$x_2 = (C_0 - C).$$

Equation (1) can then be rewritten as

$$t = \frac{1}{T} x_1 + \frac{1/K_m}{T} x_2.$$

Initial estimates of the parameters can then be readily obtained by regressing t on x_1 and x_2 . The Genstat statements below carry out this process and store the initial estimates in iC_{\min} , iT , iC_0 , and $iIKm$:

```

scal iC0,iCmin,iT,iIKm
calc iC0=c$[1]
  & iCmin=0
calc x1=log((iC0-iCmin)/(c-iCmin))
  & x2=(iC0-c)
model t
terms x1+x2
fit [con=o] x1+x2
rkeep esti=e
calc iIKm=e$[2]/e$[1]
  & iT = 1 / e$[1]

```

5. Example

The procedure was used to analyse twelve sets of data (more detail will be given in Saker and Brain (1994). The data set presented here was obtained from a 16-day old *B. campestris* plant and the concentration of radio-labelled sulphate present in the nutrient solution (counts per minute per 0.1 ml aliquot) (denoted by c) was recorded at a series of times (t).

```

read [setn=y;ser=y] t,c
0 15 29.5 45 60 75 90 105 120 135 150 165 195 225 255
284 315 375
:
11277 10221 9065 8308 7017 6445 5047 4197 3543 3019 2371 1836 1226 810 527 340
282 182
:

```

In the example below no transform was used; if a log transform had been needed *c* would have been replaced by *lc* in the MODEL statement, and expressions *E[1...19]* would have been used in the FITNONLINEAR statement. The equation can be readily fitted with *Cmin* or *IKm* (or both) set to zero.

```

model c;fitt=fc
rcyc [meth=n;maxc=50] C0,Cmin,T,IKm ; init=iC0,iCmin,iT,iIKm
fitn [calc=E[1...18];print=mon,m,s,e,f]

```

This approach was used to fit to both log-transformed data (using the transform-both-sides technique), and untransformed data; there was slight evidence that the log-transform gave a better residual plot. There was also some doubt as to whether *Cmin* was different from zero, so the equation was fitted both with, and without, *Cmin*.

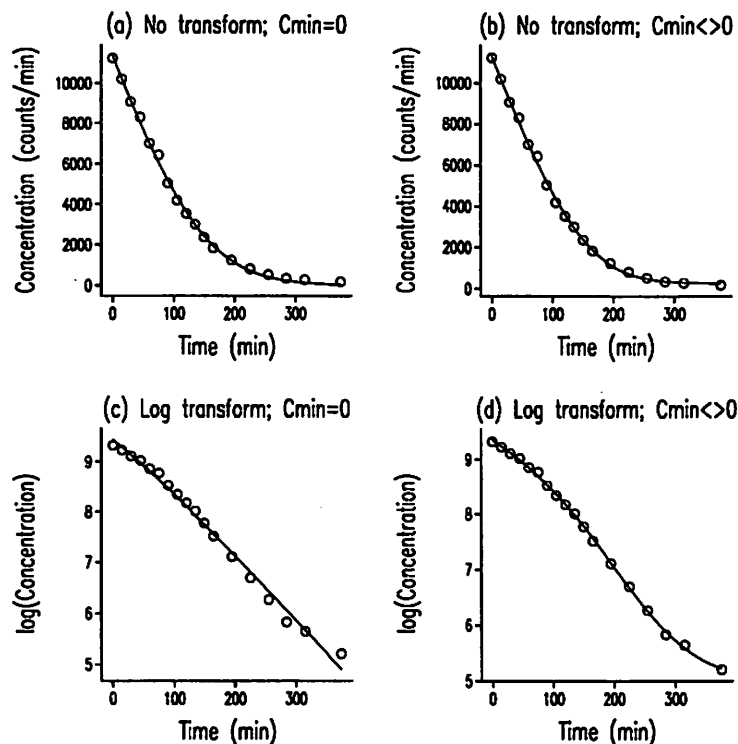


Figure 1. Observed and fitted concentrations (measured as counts per minute) for untransformed – (a) and (b) – and log transformed – (c) and (d) – counts, with C_{min} constrained to zero – (a) and (c) – and unconstrained – (b) and (d) – for the example set of data.

Figure 1 presents the results of the four combinations (with or without log transform; with and without C_{min}). The residual sums of squares from the analysis of log-transformed data for with-, and without-, C_{min} were 0.01959 (on 14 d.f.) and 0.2536 (on 15 d.f.) respectively, giving an approximate *F*-statistic for comparing the constrained and unconstrained model of $(0.2536-0.01959)/(0.01959/14) = 167.24$ on 1 and 14 d.f.

There is thus clear evidence that C_{min} is significantly different from zero. The final parameter estimates (with

standard errors on 14 d.f. in parentheses) are:

C₀ (cpm per 0.1 ml aliquot) = 11621 (238);
T = 0.01970 (0.00076);

C_{min} (cpm per 0.1 ml aliquot) = 149 (10);
IK_m = 0.000143 (0.000017)

For this experiment the root weight (*W*) was 0.56 g, the volume of solution (*V*) was 50 ml, and the specific activity 3729.3 counts/ min/ nmol. This gives *C*_{min}, *K*_m and *C*₀ to be 0.40, 18.75 and 31.16 μMol respectively (for example *C*₀ = 11621 / 3729.3 nmol/ 0.1 ml = 3.116*10⁴ / 10³ μMol). The parameter *I*_m can be calculated from the estimated values of *K*_m and T to be 1.98 μmol/ g/ hr (0.01970 * 0.05 * 18.75 / 0.56 μmol/ g/ min).

References

- Brain P and Saker L R (1993) Using GENSTAT for inverse non-linear regression. Abstract for 8th International Genstat Conference, Canterbury, July 1993.
- Claassen N and Barber S A (1974) A method for characterizing the relation between nutrient concentration and flux into roots of intact plants. *Plant Physiol.* 54 564-568.
- Drew M C, Saker L R, Barber S A and Jenkins W (1984) Changes in the kinetics of phosphate and potassium absorption in nutrient-deficient barley roots measured by a solution-depletion technique. *Planta* 160 490-499.
- McLachlan K D, Kuang Yan-hua and Muller W J (1987) An assessment of the depletion technique for comparative measurement of phosphorus uptake in plants. *Aust. J. Agric. Res.* 38 263-277.
- Ridout M S (1993) A note on fitting a growth-curve model. *Genstat Newsletter* 29 6-8
- Rudemo M, Ruppert D and Streibig J C (1989) Random-effect models in non-linear regression with applications to bioassay. *Biometrics* 45 349-362.

RUNGEN – a user-friendly Genstat interface

*D Kilpatrick
Biometrics Division
Department of Agriculture for N Ireland
Newforge Lane, BELFAST BT9 5PX
N Ireland*

*L Easson
Agricultural Research Institute
Department of Agriculture for N Ireland
HILLSBOROUGH, Co Down, BT26 6DP
N Ireland*

1. Introduction

Agricultural and food research scientists in the Department of Agriculture for N Ireland carry out a wide range of experiments, some of which, while maybe not very sophisticated in terms of statistical design, tend to generate vast amounts of data. In this they are supported by technical staff, with minimal statistical expertise, who usually accumulate these data onto PC spreadsheets either manually or from direct capture data loggers. Consultant statisticians in Biometrics Division provide a statistical analysis service for these experiments but, in common with other similar divisions, can quickly be overwhelmed by the demands for their services. There is thus a need for a system, which can be easily used by support staff, to provide intermediate summaries and statistical analysis of experimental data. One of the main requirements for ease of use is the ability to readily accept large amounts of ASCII data in a typical spreadsheet rows and columns layout without the need for additional formatting. While Genstat undoubtedly has the necessary summary and analysis facilities, its command driven language proves an obstacle for most technical support staff. The Menu procedure developed by Peter Lane is a recent attempt to address this problem. An alternative method has been developed over a number of years within Biometrics Division. This article describes a Fortran program RUNGEN which provides a user-friendly interface to writing Genstat programs and which has special facilities to ease the input of spreadsheet data. There are two versions of the interface program. One is specifically intended for VAX/VMS usage as it makes use of its supplied screen management routines, while the other is for general use.

2. Features

RUNGEN operates via a series of menus from which the user specifies the form of the data and the type of summary/analysis required. On accessing RUNGEN the user is presented with the following main menu:

Main menu: Input Calculate Display Analysis Modify Exit

The user selects the required action by typing the unique capital letter associated with each option. Additionally on the VMS version, the arrow keys can be used to highlight the required action.

1. Input provides the following features:

- inputs data either from a file or via the keyboard;
- if from file, RUNGEN reads the data file and prompts the user to specify the type of information in the file;
- automatic incorporation of column headings, typically used in spreadsheets, into the Genstat program either as extra text for variates or as names for factors;
- automatic interpretation of data structures as either text, variate or factor with provision for the user to change if required.

2. *Calculate* provides prompts for the calculation of additional variate and factor structures.
3. *Display* and *Analysis* provide access to a range of pre-defined techniques which RUNGEN reads from a start-up file. The displays and analyses specified can be for either standard Genstat directives as in:

```
TABULATE[CLASSIFICATION=#F;MARGIN=#h(Yes/no);  
PRINT=#h(count,total,Mean,Nobs,total,mInimum,maXimum)]DATA=#V
```

or local procedures as in:

```
REGRESSION[CONSTANT=#h(Estimate/omit);  
METHOD=#h(Linear/asymptotic/sigmoidal/multiple/sTepwise)]X=#V;Y=#V
```

For both forms the following conventions apply:

- the # symbol generates a prompt for input;
- the letter immediately following # indicates the type of input, i.e. *F* for factors, *V* for variates, *T* for text, *S* for structures, *H* for options;
- a capital letter indicates that the input is compulsory, a lower-case letter that it is optional;
- valid options are enclosed by round brackets and are separated by either forward slashes if only one option is allowed or by commas if several are allowed;
- options with an initial upper-case letter indicate the default, e.g. *Mean* and *Nobs* for *Tabulate Print* options;
- options are toggled on/off by typing the appropriate initial or capital letter, e.g. *C* for *count* and *I* for *minimum Tabulate Print* options.

When either *Display* or *Analysis* is selected, RUNGEN presents a menu showing the pre-defined techniques from which the user selects the required one. RUNGEN then automatically generates prompts for the compulsory input followed by a prompt to allow the user to modify both the option settings and the compulsory inputs. For example, for the *Tabulate* directive selected from the *Display* menu:

Type of Display? Print Tabulate Graph Histogram Barchart Quit

Enter Tabulate Classification-factors:

Enter Tabulate Data-variates:

Modify Tabulate? Classification-factors Data-variates Print Margin Quit Zap

Selection of *Quit* completes the specification for the display/analysis while *Zap* deletes it.

4. *Modify* allows the user to modify and/or delete previously defined display and analysis operations. On specifying the type of technique to be modified, the user is presented with the corresponding modify menu as for *Modify Tabulate* in the previous paragraph.
5. *Exit* exits the interface program resulting in the creation of two files. The first is a file of Genstat statements to carry out the specified input, display and analysis operations, while the second is a control file of user input which can be replayed at a later session and the analyses specified modified and/or extended.

3. Example

To illustrate the operation of RUNGEN consider a $2 \times 2 + 1$ factorial experiment with 2 replicates and the data laid out in a file XYZ.DAT as shown in Figure 1. This layout is typical of spreadsheet data with titles at the top describing the experiment and supplying names for the data columns.

Figure 1. Example data file XYZ.DAT

Experiment XYZ				
Stage I measurements				
Rep	Diet Type	Meal	Milk Yield (1/d)	Fat %
1	A	2	25.8	3.13
1	B	4	29.3	2.97
1	C	0	21.2	3.18
1	A	4	30.0	2.85
1	B	2	24.5	3.44
2	C	0	27.6	2.71
2	B	2	28.1	3.10
2	A	2	29.9	3.28
2	A	4	23.9	4.01
2	B	4	23.6	3.54
:				

The dialogue between user and program to specify the type of information in this file is illustrated in Figure 2.

Figure 2. Example dialogue to read data file

(Program output is shown in italics, user input in bold and menu selections underlined)

```

Enter number of experimental units: 10
Main menu: Input Calculate Display Analyse Modify Exit
Input from: File Keyboard
Enter name of file: XYZ.DAT
File contents? Data-only Titles-plus-data
Read options: Continue Missing-value-indicator End-of-data-marker
    
```

program reads and displays each line in data file and prompts user to identify the type of information

```

Experiment XYZ
Type of information: General-title Sub-title Names Data Ignore Quit

Stage I measurements
Type of information: General-title Sub-title Names Data Ignore Undo Quit

      Diet           Milk           Fat
Type of information: General-title Sub-title Names Data Ignore Undo Quit
Rep  Type  Meal  Yield (1/d)  %
Type of information: General-title Sub-title Names Data Ignore Undo Quit
  1   A     2    25.8      3.13
Type of information: General-title Sub-title Names Data Ignore Undo Quit
Read options: Continue Options Undo Quit
    
```

program reads to end-of-data marker, displays current information and provides user with opportunity to make modifications

```

General Title: Experiment XYZ
Sub-title: Stage I measurements

Number      Name           Type           Levels          First           Last
1           Rep            Factor         2 Formal        1              2
2           Diettype      Factor         3 Text          A              B
3           Meal          Factor         3 Numeric       2              4
4           Milk Yield (ltd) Variate
5           Fat %         Variate

Modify: Continue General-title Sub-title structure-Names Types
    
```

program returns to main menu since end of data file reached

The main feature is that RUNGEN reads the data file line by line and prompts the user to specify the type of information contained on each line. After the user identifies the first data line, RUNGEN reads to the end of data marker and automatically determines both the number and the types of structures. Any factor names are automatically edited to be unique and obey the rules for such names. The user then has the opportunity to modify this if necessary. RUNGEN reads as many sets of data as are on the file and returns the user to the main menu when the end of file is encountered.

From the main menu, the user can select options to input additional data, calculate new structures, and display and analyse the data. As previously described the displays and analyses available are defined in a file read by RUNGEN at start-up. Dialogue corresponding to the Tabulate display directive is shown in Figure 3.

Figure 3. Example dialogue for Tabulate directive

```

Main menu: Input Calculate Display Analyse Modify Exit
Type of display? Print Histogram Graph Tabulate Barchart Quit
    
```

program prompts the user for the compulsory input displaying lists of factors and variates, identified by number and name, as appropriate

```

Num           Structure name      Num           Structure name
1             Rep                2             Diettype
3             Meal
Enter Tabulate Classification-factors: 2,3
Num           Structure name      Num           Structure name
4             Milk yield (ltd) 5             Fat %
Enter Tabulate Data-variates: 4,5
    
```

program shows current settings and prompts the user to make modifications

```

Classification-factors:      Diettype,Meal
Data-variates:              V[4,5]
Print:                       Nobs, Mean
Margin:                       Yes
Modify Tabulate?  Classification-factors Data-variates Print Margin Quit Zap

Edit Tabulate Print-options: count total Nobs Mean mIn maX var Quit
    
```

(the user selects/deselects options as required)

An additional feature worth mentioning is the automatic definition of a control factor to deal with the factorial plus added control design in this example. This is illustrated in Figure 4.

Figure 4. Example dialogue for Analysis of Variance

```

Main menu: Input Calculate Display Analyse Modify Exit
Type of analysis? Analysis-of-variance Regression Variate-regression
Type of design? Completely-randomised Randomised-block Latin-square Split-plot

Num      Structure name      Num      Structure name
1        Rep                 2        Diettype
3        Meal

Enter treatments factors: 2,3
Factorial plus control: No Yes

Diettype: 1=A   2=B   3=C
Enter number of control level for Diettype: 3

Meal:        1=0   2=2   3=4
Enter number of control level for Meal: 1

Num      Structure name      Num      Structure name
4        Milk yield (l/d)    5        Fat %

Enter Analysis-of-variance Y-variates: 4,5
Modify Analysis-of-variance: Y-variates SE Covariates Quit Zap

```

4. A user's viewpoint

A centre at which RUNGEN has been used for some time is the Agricultural Research Institute of Northern Ireland, Hillsborough. In agricultural, as in all research, the rapid statistical analysis of experimental data is of great importance in maintaining progress and research workers found that, in spite of good computer communications between centres, inevitable delays were taking place while numerous data sets were sent to the Biometrics service for analysis. Attempting to come to grips with this problem with the writing of Genstat programs by individual scientists or support staff at the Institute would have been an unwanted diversion from their main research activities and so the development of RUNGEN as a method of preparing statistical analyses without coming into direct contact with Genstat commands has been a very valuable service. The ease of use of RUNGEN has also reduced the need to perform intermediate calculation of means and other values as the full analysis of any data set can be achieved in minutes. The range of facilities in RUNGEN including the analysis of split-plot designs, the use of co-variates, factorials plus controls, correlations and regressions is adequate to meet most standard experimental designs at the Institute and other needs can be met by straightforward modification of the resulting Genstat program file. RUNGEN is made particularly user-friendly by the system of prompts, hints and menu choices so that relatively little training is required, provided users are familiar with the computer system and with the preparation of data. There have been few problems with acceptance of RUNGEN by scientific staff, experienced technicians and research assistants with sufficient computer experience and it has become the standard procedure for the analysis of about 60% of the research data from the Institute.

5. Discussion

RUNGEN has been used extensively by the non-specialist staff for whom it was intended and it has undoubtedly proved beneficial. The most commonly employed technique is analysis of variance of designed experiments and here it is emphasised that a statistician should always be consulted if there is any doubt over the design. Typically this is sorted out in the initial stages of data analysis, leaving the non-specialist staff to include extra

data for analysis as they become available. RUNGEN has also been of benefit to experienced Genstat users particularly regarding its ability to quickly input large numbers of structures with corresponding column headings. The file of Genstat statements can then be edited to carry out more complex operations. Alternatively the file containing pre-defined display and analysis operations can be easily extended to include other operations as required.

The major remaining stumbling block for users seems to be the detection and correction of coding errors when inputting data from files. To overcome this it is intended that a later version will include the detection of

- incomplete lines of data;
- possible typing errors e.g. "o" and "i" for the numbers zero and one;
- mis-coding of factor levels leading to unequal replication for a balanced design.

The fact that RUNGEN, unlike the Menu procedure, does not allow interactive Genstat use is more of a limitation to the experienced user than the non-specialist staff for whom RUNGEN was originally intended. Future developments may include the use of the Genstat *Own* directive to overcome this limitation.

An interface between Genstat and the Brief editor on PC

P W Goedhart
DLO Agricultural Mathematics Group
P.O. Box 100
6700 AC WAGENINGEN
The Netherlands

1. Introduction

Brief (Borland International, 1992) is a professional program editor for DOS PCs providing editing of multiple files in multiple windows, extensive search and translate capabilities, the ability to undo most commands, template editing, multiple keystroke macros, a completely reconfigurable keyboard, a flexible macro language, compilation of programs from within the editor and mouse support.

Brief's macro language gives the ability to extend and change the editing environment. Its syntax resembles the C language. I have written several Brief macros which allows you to:

- use an extended EDT-style keyboard, where EDT is the VAX/VMS editor EDIT/EDT;
- execute a DOS command from within Brief and automatically return to the editor when the DOS command is completed. This interface to DOS can be fully customized to your own needs. You can, for example, run a Genstat program from within Brief and automatically view/edit the output file after the Genstat run has been completed. You can also compile and link Fortran or C programs, for example, from within Brief with different options or you can send files to a printer;
- view the Genstat reference summary for every directive and procedure in a separate window and to automatically insert directives, options and parameters in the Genstat program file. You can also view the directive/procedure index and the directive/procedure modules.

The Genstat help macros and the interface to DOS are most easily explained by an example session, which is given in the next section. The macros and a full description are available from the author.

2. Example session

Suppose you are using Brief to edit a file TOMATO.GEN, which contains a Genstat program to calculate the weight of a tomato on different days. The PC screen then looks as follows; the cursor position is denoted by ■:

■:

```

tomato.gen
VARIATE [VALUES= 21, 25, 30, 35, 39] day
VARIATE [VALUES= 2, 5, 15, 22, 28] diameter
CALCULATE pi = CONSTANTS('pi')
CALCULATE weight = 4/3 * pi * (diameter/2)**3 / 1000
PRINT day, weight■
STOP
.
.
.
.

```

BRIEF v3.1 - Copyright (c) 1991 Borland Internat Line: 5 Col: 22 # 12 40

Suppose that you want to print the day and weight variates serially across the page with no decimal places for the day variate. Help on the PRINT directive can be obtained by pressing key F11. Help on the PRINT directive is retrieved because the cursor is located on a line with a PRINT directive. The PC screen then displays two windows: a top window with the Genstat program file and a bottom window with reference help on PRINT.

```

===== tomato.gen =====
VARIATE [VALUES= 21, 25, 30, 35, 39] day
VARIATE [VALUES= 2, 5, 15, 22, 28] diameter
CALCULATE pi = CONSTANTS('pi')
CALCULATE weight = 4/3 * pi * (diameter/2)**3 / 1000
PRINT day, weight
STOP

===== PRINT =====
PRINT
► Prints data in tabular format in an output file, unformatted file, or
► text.
Options
CHANNEL = identifier          Channel number of file, or identifier of a
                              text to store output; default current
                              output file
SERIAL = string              Whether structures are to be printed in
                              serial order, i.e. all values of the first
                              structure, then all of the second, and so
                              on (yes, no); default no, i.e. values in
                              parallel
IPRINT = string              What identifier (if any) to print for the
                              structure (identifier, extra,
                              associatedidentifier), for a table
    
```

Insert Directive PRINT Line: 5 Col: 22 # 12 40

The index of the PRINT directive is highlighted by means of the ► character. Key PageDown moves the highlight down to the next option or parameters in the help window, while PageUp moves the highlight up. So pressing PageDown twice moves the highlight to the SERIAL option. Key Insert then inserts the highlighted option in the Genstat file, as is shown on the PC screen below. Note that the cursor has moved to the position after the inserted option, so that the option setting (yes) can be typed conveniently.

```

===== tomato.gen =====
VARIATE [VALUES= 21, 25, 30, 35, 39] day
VARIATE [VALUES= 2, 5, 15, 22, 28] diameter
CALCULATE pi = CONSTANTS('pi')
CALCULATE weight = 4/3 * pi * (diameter/2)**3 / 1000
PRINT [SERIAL=►] day, weight
STOP

===== PRINT =====
Options
CHANNEL = identifier          Channel number of file, or identifier of a
                              text to store output; default current
                              output file
► SERIAL = string            Whether structures are to be printed in
►                             serial order, i.e. all values of the first
►                             structure, then all of the second, and so
►                             on (yes, no); default no, i.e. values in
►                             parallel
IPRINT = string              What identifier (if any) to print for the
                              structure (identifier, extra,
                              associatedidentifier), for a table
                              associatedidentifier prints the identifier
                              of the variate from which the table was
                              formed (e.g. by TABULATE), IPRINT=*
    
```

Insert Option SERIAL Line: 5 Col: 19 # 12 40

The ORIENTATION option is inserted in the same way, i.e. PageDown is pressed until the ORIENTATION option is highlighted and the Insert key is then used to insert the ORIENTATION option. The setting across is subsequently typed. In order to insert the DECIMALS parameter, PageDown can be pressed until the DECIMALS parameter is highlighted. Alternatively, Ctrl-P can be pressed to move the highlight to the first parameter. The key sequence PageDown, PageDown, Insert then inserts the DECIMALS parameter and the PC screen looks as follows:

```

tomato.gen
VARIATE [VALUES= 21, 25, 30, 35, 39] day
VARIATE [VALUES= 2, 5, 15, 22, 28] diameter
CALCULATE pi = CONSTANTS('pi')
CALCULATE weight = 4/3 * pi * (diameter/2)**3 / 1000
PRINT [SERIAL=yes ; ORIENTATION=across] day, weight ; DECIMALS=
STOP

```

```

PRINT
omitted, a default is determined (for
numbers, this is usually 12; for text, the
width is one more character than the
longest line)
▶ DECIMALS = scalars      Number of decimal places for numbers; if
▶                          omitted, a default is determined which
▶                          prints the mean absolute value to 4
▶                          significant figures
CHARACTERS = scalars     Number of characters to print in strings
SKIP = scalars or variates Number of spaces to leave before each value
of a structure (* means newline before
structure)
FREPRESENTATION = strings How to represent factor values (labels,
levels, ordinals); default is to use labels
if available, otherwise levels

```

Insert Parameter DECIMALS

Line: 5 Col: 68 # 12 41

The DECIMALS parameter is then set to 0,*. When you want to run this Genstat program and edit the output file afterwards, you normally have to exit the editor, run the DOS command

```
GENSTAT TOMATO.GEN,TOMATO.LIS
```

and edit the resulting output file TOMATO.LIS after Genstat has stopped. The interface to DOS allows you to combine these actions in key F12. Assuming the interface has been set up correctly, pressing F12 runs the Genstat program and edits the output file, giving the following PC screen:

```

tomato.lis
■Genstat 5 Release 3.1 (IBM-PC 80386/DOS) 10 June 1994
Copyright 1993, Lawes Agricultural Trust (Rothamsted Experimental Station)

1 VARIATE [VALUES= 21, 25, 30, 35, 39] day
2 VARIATE [VALUES= 2, 5, 15, 22, 28] diameter
3 CALCULATE pi = CONSTANTS('pi')
4 CALCULATE weight = 4/3 * pi * (diameter/2)**3 / 1000
5 PRINT [SERIAL=yes ; ORIENTATION=across] day, weight ; DECIMALS=0,*

    day          21          25          30          35          39
    weight      0.004      0.065      1.767      5.575      11.494

6 STOP

***** End of job. Maximum of 2350 data units used at line 5 (6533944 left)

```

BRIEF v3.1 - Copyright (c) 1991 Borland Internat Line: 1 Col: 1 # 12 41

The contents of the Genstat program, i.e. the file TOMATO.GEN, can be viewed by using Brief's facilities to switch between files (Alt-n would be sufficient for this session). External datafiles can also be edited simultaneously and modified if necessary. Moreover, Brief has the ability to edit files in multiple windows, so the Genstat program file can be edited in one window and the Genstat output file in another window. Running Genstat from inside Brief, by means of key F12, automatically updates the output file window with the new Genstat output file.

There is also help available on the directive/procedure modules and on the directive/procedure index. For example, the key sequence Alt-F11, I, Enter displays the AOV directive module with the first line highlighted:

```

z.gen
-----
Modules
-----
AOV directives
The following directives analyse balanced experiments:

BLOCKSTRUCTURE          Defines the design
TREATMENTSTRUCTURE, COVARIATE  Specifies effects
ANOVA                   Carries out the analysis
ADISPLAY, AKEEP        Displays or saves results

and unbalanced experiments can be analysed using:

REML                    Fits a variance-components model
VCOMPONENTS            Specifies a variance-components model
VDISPLAY, VKEEP        Displays or saves results

COMMUNICATION directives
The following directives control input and output of data:

File handling          OPEN, CLOSE, ENQUIRE
Switching between files INPUT, OUTPUT, RETURN
Reading data          READ, DREAD, SPREADSHEET
!!, PgUp, PgDn, End, Home  Enter to select  Esc to exit
    
```

Search for index of directive: Line: 31 Col: 1 # 12:42

The bottom of this screen indicates that you can search for the index of directives (and procedures). Typing AN switches to the index help screen in which the ANOVA directive is highlighted:

```

z.gen
-----
Index
-----
AKEEP (d)
Copies information from an ANOVA analysis into Genstat data structures.
AKEY (p)
generates values for treatment factors using the design key method
ALIAS (p)
finds out information about aliased model terms in analysis of variance
ANOVA (d)
Analyses y-variables by analysis of variance according to the model defined
by earlier BLOCKSTRUCTURE, COVARIATE, and TREATMENTSTRUCTURE statements.
ANTORDER (p)
assesses order of ante-dependence for repeated measures data
ANTTEST (p)
calculates overall tests based on a specified order of ante-dependence
AONEWAY (p)
provides one-way analysis of variance for inexperienced users
APLOT (p)
plots residuals from an ANOVA analysis
ASSIGN (d)
Sets elements of pointers and dummies.
ASWEEP (p)
!!, PgUp, PgDn, End, Home  Enter to select  Esc to exit
    
```

Search for index of directive: AN Line: 34 Col: 1 # 12 42

The letter (d) indicates that ANOVA is a directive, not a procedure. Pressing Enter would now exit the index help and display reference help on the ANOVA directive in a separate window, in the same way as reference help on the PRINT directive was displayed earlier in this section. The directive/procedure index is also directly available by means of Shift-F11. Cursor keys can be used to scroll in the index and module help.

Reference

Borland International (1992) *Brief for DOS and OS/2, Version 3.1* 1800 Green Hills Road, P O Box 660001, Scotts Valley, CA 95067-0001, USA.

A Genstat procedure to calculate a kappa coefficient of agreement for nominally scaled data

A. J. Rook

AFRC Institute of Grassland and Environmental Research

North Wyke, Okehampton

Devon EX20 2SB

UK

Consider an experiment in which each of k judges assigns each of N objects or subjects (e.g. patients) to one of m categories (e.g. response to drug). The results of the experiment can be formed into an $N \times m$ table in which the elements n_{ij} , represent the number of judges assigning object i to category j . The judges may show complete agreement, partial agreement or no agreement (other than due to chance).

The degree of agreement between judges can be measured using the statistic

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

where $P(E)$ is the expected proportion of times that the judges agree by chance and $P(A)$ is the actual proportion of times they agree. K is thus the ratio of the proportion of times the judges agree to the maximum proportion of times they could agree, both corrected for chance agreement. $K = 1$ when there is perfect agreement and 0 if assignment is at random.

The expected proportion of times the judges agree is calculated as

$$P(E) = \sum_{j=1}^m \left(\frac{\sum_{i=1}^N n_{ij}}{Nk} \right)^2$$

and the actual proportion of times they agree as

$$P(A) = \left[\frac{1}{Nk(k-1)} \sum_{i=1}^N \sum_{j=1}^m n_{ij}^2 \right] - \frac{1}{k-1}$$

The sampling distribution of K is asymptotically normal with mean 0 and variance

$$\text{var}(K) \approx \frac{2}{Nk(k-1)} \frac{P(E) - (2k-3)[P(E)]^2 + 2(k-2)\sum p_j^3}{[1-P(E)]^2}$$

Therefore, for large N , the statistic

$$z = \frac{K}{\sqrt{\text{var}(K)}}$$

can be used to test the hypothesis $H_0: K=0$ against $H_1: K>0$. For further details of the method see Siegel and Castellan (1988).

A Procedure (KAPPA) has been written to carry out this analysis in Genstat. A single parameter NASSIGNED is set to the $N \times m$ table described above. An example of the output is shown below. For $N < 20$ a warning that the hypothesis testing may not valid is printed.

*** Measures of agreement for nominally scaled data ***

Proportion of times judges agree

Actual	Expected	Kappa coefficient of agreement	Variance
0.580	0.288	0.410	0.00271

*** Test of significance of Kappa ***

Z	P
7.887	0.000

Reference

Siegel S and Castellan N J (1988) *Nonparametric Statistics for the Behavioral Sciences*. 2nd Edn. McGraw-Hill, Singapore.

Appendix: Genstat Procedure

PROCEDURE 'KAPPA'

A. J. Rook,
AFRC Institute of Grassland and Environmental Research,
North Wyke, Okehampton
Devon EX20 2SB

Version 1.2 4/12/92

Procedure to calculate a kappa coefficient of agreement for nominally scaled data. Input is a N x m table with N objects to be classified and m categories. Each entry n(ij) in the table is the number of judges assigning the ith object to the jth category.

It is assumed that all judges assign all items. Therefore all row totals must be equal to the numbers of judges. Missing values in the table are not allowed.

Reference: Siegel, S. and Castellan, N. J. (1988) *Nonparametric Statistics for the Behavioral Sciences*. 2nd Edn. McGraw-Hill, Singapore.

```
PARAMETER NAME= \
  'NASSIGNED'; "(I: table) table of N objects (rows) x m categories (columns)" \
  "with each entry being the number of judges assigning the ith \"
  "object to the jth category" \
  SET=yes; DECLARED=yes; TYPE='table'; PRESENT=yes
```

Obtain the number of rows and columns in the table. Check that the table contains no missing values. If it does print error message and abort procedure

```
GETATTRIBUTE [ATTRIBUTE=nmv,classification] NASSIGNED; SAVE=p
& [ATTRIBUTE=nlevels] p['classification'][1,2]; save=row,col
EXIT [CONTROL=procedure; EXPLANATION= \
  '*** ERROR - Input table to procedure KAPPA contains missing values ***'] \
  p['nmv'].GT.0
```

Add margins to table and save in new table internal to procedure

```
MARGIN NASSIGNED; NEWTABLE=nassign; METHOD=t
```

Check that all row totals are equal

```
SCALAR rmarg[1...#row['nlevels']],cols,pe,sumnass2,pa
CALC cols=-#col['nlevels']
EQUATE [OLDFORMAT=1((#cols,1)#row['nlevels'])] nassign; rmarg
FOR i=2...#row['nlevels']
  CALC j=i-1
  EXIT [CONTROL=procedure; EXPLANATION=\
  '*** ERROR - Row totals of input table for procedure KAPPA are not equal ***']\
  rmarg[i].NE.rmarg[j]
ENDFOR
```

Calculate expected proportion of times that the k judges agree by chance


```

"
TABLE [CLASS=p['classification'][2]] cmarg
CALC cmarg=TSUMS(NASSIGNED)
PERCENT [HUNDRED=yes; METHOD=totals] OLD=cmarg; NEW=%pj
MARGIN OLDTABLE=%pj; METHOD=deletion
CALC pe=SUM((%pj/100)**2)
"
Calculate actual proportion of time that k judges agree (k=rmarg[1])
"
  & sumnass2=SUM(NASSIGNED*NASSIGNED)
  & pa=((1/#row['nlevels']*rmarg[1]*(rmarg[1]-1))*sumnass2)-(1/(rmarg[1]-1))
"
Calculate kappa coefficient of agreement
"
  & K=(pa-pe)/(1-pe)
"
Calculate approximate variance of kappa coefficient
"
  & varK=(2/(#row['nlevels']*rmarg[1]*(rmarg[1]-1)) * \
    ((pe-((2*rmarg[1]-3)*(pe**2))+2*(rmarg[1]-2)*(sum((%pj/100)**3)))) / \
    ((1-pe)**2))
"
Calculate z statistic
"
  & z=K/sqrt(varK)
"
Obtain P value for z statistic
"
  & P=1-NORMAL(z)
"
Print results
"
PRINT '*** Measures of agreement for nominally scaled data ***'
  & [SQUASH=yes] 'Proportion of times judges agree'
  & '   Actual      Expected',\
  & '   Kappa coefficient of agreement   Variance'
  & [IPRINT=*] pa,pe,K,varK; FIELD=11,16,26,24; DECI=3(3),5
  & [SQUASH=no] '*** Test of significance of Kappa ***'
  & [SQUASH=yes] '   Z           P'
  & z,P; FIELD=11,18; DECI=2(3)
"
If number of objects judged is small print warning that test is not valid
"
IF #row['nlevels'].LE.20
  PRINT '*** WARNING ***'
  & [SQUASH=yes] 'LESS THAN 20 OBJECTS RATED - TEST NOT VALID'
ENDIF
ENDPROCEDURE
RETURN

```

Analysis of unbalanced multi-stratum trials using ANOVA and REML

Rosie Poultney
 Rothamsted Experimental Station
 Harpenden
 Herts AL5 2JQ

1. Introduction

Trials carried out in less developed countries tend to have more missing values than are usual in agricultural research in UK. There are many reasons for these missing values, some of which will be very familiar, others may not be. In addition, there are situations where modifications to trials have resulted in unbalanced designs.

The main causes of lack of balance in work I have received can be categorised into three groups; initial design problems, difficulties with treatments and difficulties with experimental material. Researchers are often in situations where it is difficult to consult a statistician and this can reflect in inappropriate initial designs or modifications of designs. Even with a well-designed trial, lack of balance can occur when the treatments are incorrectly applied or when the researcher cannot obtain sufficient quantities of the treatments. Finally there are problems associated with the experimental materials. This can range from insufficient seed to high mortality in crops, from crop pilfering to fires and floods. I recently heard from a colleague who, on returning to Malaysia, found that a herd of elephants had rampaged through one of her plots.

Lack of balance usually presents no major problems of analysis, except where the trial contains more than one error stratum. Before REML was implemented into Genstat, I would analyse these trials using a mixture of ANOVA and Regression techniques which required a great deal of explanation to clients without providing a satisfactory analysis.

2. What did I expect from REML ?

I wanted all of the facilities that I was already using in ANOVA and Regression, namely: 1) easy specification of the model; 2) clear output including means and standard errors, tests for the significance of treatments and estimates of variation in the different strata; and 3) model validation via residuals and fitted values. Since REML is equivalent to ANOVA for balanced datasets, I was also concerned to know where REML was analogous to ANOVA and where direct comparisons between the techniques could not be made for unbalanced data. Table 1 summarises similarities and differences between ANOVA and REML with regard to these criteria, assuming a designed experiment where the treatments are fixed effects and the blocks are random effects. Theoretically, REML could provide all of the things which I want from an analysis, but how simple was it in practice ?

Table 1: Comparison of Genstat REML and ANOVA

	REML	ANOVA
1: model specification	vcomponents [fixed=a*b] random=rep/block	block rep/block treatment a*b
2: output - means - s.e.d.'s - tests for treatment terms - stratum variances	predicted means s.e.d.'s (unaffected by order of fit) Wald Statistics tested against χ^2 (fitted sequentially) estimated variance components plus approximate stratum variances	means s.e.d.'s variance ratios residual mean squares (convert to variance components)
3: model validation	residuals and fitted values	residuals and fitted values

3. An Example

Cloves are a spice which in Europe are used most often in cooking. They are the dried unopened flower bud of the clove tree (*Syzygium aromaticum*). Cloves are indigenous to Indonesia and were introduced to Zanzibar in the 19th Century. A major constraint on the stability and expansion of the clove industry in Zanzibar has been the hostile environment (two dry seasons) which makes establishment of new trees difficult. Two of the many methods which have been used to aid establishment are cover crops and fertilisers.

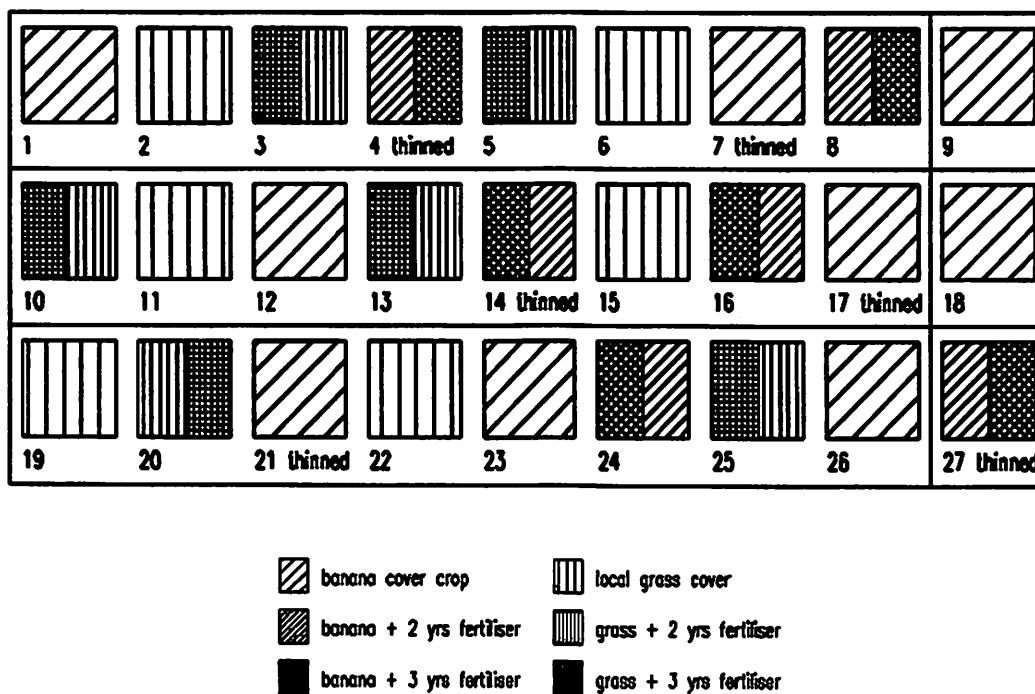


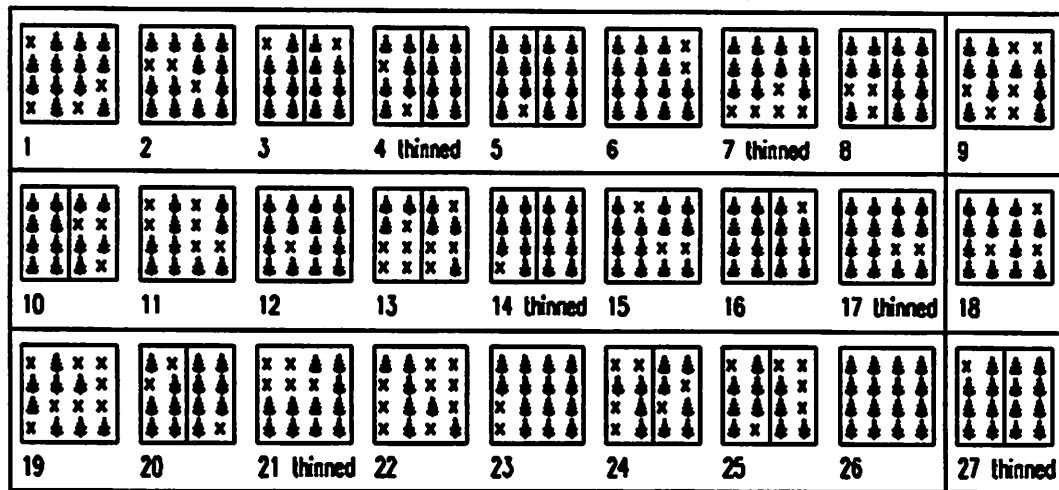
Figure 1. Field plan detailing treatment structure

The experiment described here began as a trial investigating the importance of cover crops. The trial was laid out as a randomised block design with three replicates, each of eight plots. Each plot comprised 36 clove trees with the inner 16 being monitored as part of the trial. Within each replicate, four plots had bananas grown as a shade crop whilst the remaining four had the natural vegetation. In addition, an extra banana plot was monitored in each replicate. The trial is described in detail as experiment 4 in Martin and Poultney (1992).

In 1988, the trial was modified to include fertiliser on two banana and two grass plots in each rep. Unfortunately there was insufficient fertiliser available during the first year and so it was only applied to half of each plot. In subsequent years fertiliser was applied to the whole plot. At this stage, one of the 'extra' plots was included in the trial. Also in 1988, banana plants on two of the banana plots in each replicate were thinned to examine the effect of increased light on the seedlings.

The experimental layout, as it stood in 1990, is shown in Figure 1. Early mortality in the trial was about 22% and is shown in Figure 2. The tree symbols represent live trees (although they bear no resemblance to clove trees, which are cylindrical in shape and can grow up to about 10 metres in height) and the crosses represent dead trees. Mortality was independent of treatment.

The variate of interest was the natural logarithm of canopy surface area of individual trees in 1990. Canopy surface area is related to yield and is a good indicator of future yields in trees of this age.





 live tree, September 1990
 dead tree, September 1990

Figure 2: Field plan showing tree mortality

4. Results

4.1 Specification of Analyses

The Genstat code for the specification of each analysis is given below. Use of a general fert factor to define fertiliser levels nil, 2 years or 3 years in ANOVA led to warnings of partial aliasing, since the two contrasts nil versus treated and 2 years vs 3 years are estimated in different strata. This problem is avoided by partitioning the factor into two nested two-level factors fcontrol and fert23 as below, or via a pseudo-factor. Thus in this example, specification of the model appeared easier using REML than ANOVA, since the ANOVA specification required more understanding of the structure of the design. Also, in order to keep the design balanced, one randomly selected plot which had treatment combination 'intercropped with bananas, no fertiliser, not thinned' had to be dropped from each replicate.

REML analysis specification:

```
vcomponents [fixed=(crop/thinned)*fert; absorb=mainplot]\
  random=rep/mainplot/subplot
reml [print=model,components,means,waldtests,stratumvariance] canopy90;\
  residuals=resid; fitted=fitted
```

ANOVA analysis specification:

```
fact [level=2] fcontrol,fert23
calc fcontrol = newlevels(fert; 1(1,2,2))          " nil vs fertiliser applied "
  & fert23 = newlevels(fert; 1(1,1,2))          " 2 years vs 3 years fertiliser "
rest canopy90; plot.ni.1(1,18,26)
block rep/mainplot/subplot
treat (crop/thinned) * (fcontrol/fert23)
anova [fact=4] canopy90; resid; fitted
```

4.2 Output

In order to make the results comparable, the REML analysis used the same restricted set of data as the ANOVA although REML could have been used to analyse the full data set. The summary ANOVA table is given below (Table 2) to illustrate the structure of the design. As stated above, there are a very large number of missing values in the data. However, these missing values are all in the lowest stratum where no treatment effects are estimated, so missing values will be estimated by sub-plot means and the treatment estimates will be equivalent

to those from an unweighted ANOVA of sub-plot means. Note that as trees are missing at random, analysis of weighted sub-plot means would lead to an unbalanced dataset so a weighted ANOVA could not be used.

Table 2: Analysis of variance table for the example

Source	df	Mean Square	Expected Mean Square
rep stratum			
Residual	2	28.33	$\sigma^2 + 8\sigma_{sp}^2 + 16\sigma_{mp}^2 + 128\sigma_r^2$
rep.mainplot stratum			
crop	1	85.01	
fcontrol	1	20.33	
crop.thinned	1	0.32	
crop.fcontrol	1	8.81	
crop.thinned.fcontrol	1	5.91	
Residual	16	4.16	$\sigma^2 + 8\sigma_{sp}^2 + 16\sigma_{mp}^2$
rep.mainplot.subplot stratum			
fcontrol.fert23	1	0.072	
crop.fcontrol.fert23	1	0.029	
crop.thinned.fcontrol.fert23	1	0.218	
Residual	21	1.895	$\sigma^2 + 8\sigma_{sp}^2$
rep.mainplot.subplot.units stratum	244 (92)	0.377	σ^2
Total	291 (92)		

The means from the REML and ANOVA analyses were very similar (Figure 3). Since REML does not weight predicted means by the number of units in a sub-plot, we would expect the REML means to be similar to the unweighted ANOVA means. If the fert factor was not partitioned, then partial aliasing would mean ANOVA was unable to estimate the treatment means correctly.

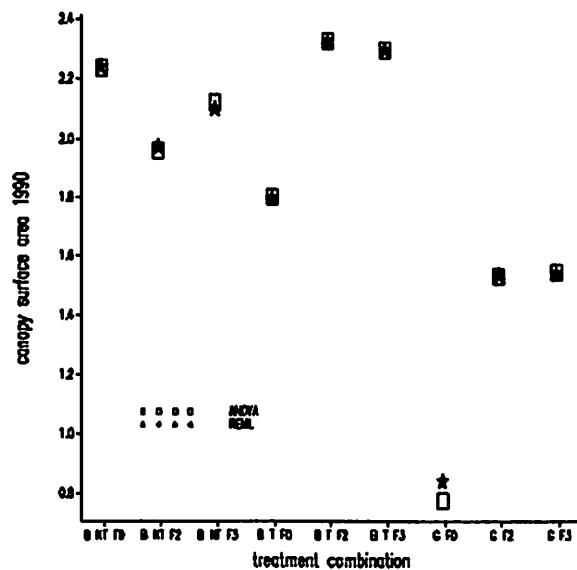


Figure 3: Comparison of treatment means from ANOVA and REML

Comparison of the standard errors of differences (SEDs) between pairs of means showed that the SEDs produced by ANOVA were higher than those produced by REML. This difference seemed surprising given the similarity in estimates of treatment means. To investigate this further, variance components were estimated from the expected values of residual mean squares in ANOVA (see Table 2) for comparison with REML estimates: both sets of estimates are shown in Table 3.

Table 3: variance components

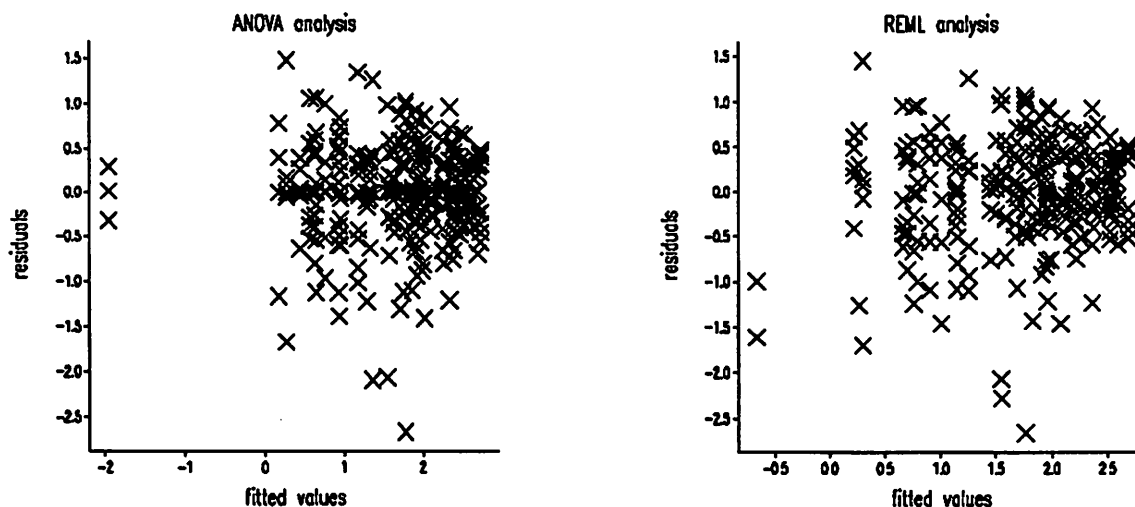
Stratum	REML	ANOVA
rep	0.169	0.189
rep.mainplot	0.137	0.141
rep.mainplot.subplot	0.091	0.190
units	0.385	0.377

Although the estimates of the residual (units) variance are similar from the two techniques, the ANOVA estimates are consistently higher than the REML estimates, particularly in the subplots stratum. The difference is due to the estimation of missing values in the units stratum by ANOVA: once these values have been estimated, they are treated as genuine data values when forming subplot means. This means that the variation due to these estimated values is added into the subplot variation along with variation from true data values, leading to over-estimation of the subplots stratum variance. Similarly, the estimated values are used to construct the treatment sum of squares hence this will also be an overestimate, as stated in the Genstat Manual.

F-tests from the summary ANOVA table indicated that model terms `crop` and `fcontrol` (the difference between treated and nil fertiliser plots) were significant ($p < 0.001$, $p = 0.042$). There was no evidence to suggest that any other significant treatment effects were present. In the REML analysis, the probability levels for terms `crop` and `fert` were $p < 0.001$ and $p = 0.082$ respectively. However, when `fert` was split up into two separate factors, the probability level for terms `fcontrol` and `fert23` were $p = 0.025$ and $p = 0.999$ respectively. Other terms were again not significant. In general, Wald statistics tend to be less conservative than F-tests from ANOVA.

4.3 Model validation

The residuals and fitted values from the two techniques are shown in Figures 4 and 5. By default, both ANOVA and REML give residuals from the residual stratum only.



The two sets of residuals are very similar apart from the outlying points on the left of the graph. These points are from a sub-plot where only two out of the eight trees survived. For this subplot, the REML estimate seems more plausible than the ANOVA estimate. The procedures APLOT and VPLOT can also be used to examine the residuals for signs of departures from a normal distribution.

5. Conclusions

Despite the large number of missing values, overall the two approaches gave similar results because the data were still reasonably balanced: the missing values were all in the lowest stratum whereas treatments were applied in higher strata. The major difference was that estimates of variance components and SEDs for means were larger from the ANOVA than from the REML analysis due to estimation of missing values in lower strata inflating the ANOVA sums of squares. The Genstat 5 Reference Manual warns about this - along with the suggestion that you should not use ANOVA when there are many missing values. At first, the REML analysis appeared simpler to specify, since the factor defining fertiliser levels had to be partitioned in order to make the dataset balanced for ANOVA. However, the benefits of using this slightly more complex structure became clear as ANOVA could produce separate tests for the individual components of `fert`, whereas REML gave an overall but less informative test.

On balance, the REML analysis using the partitioned `fert` factor is preferred since it gives an exact analysis. In particular, where information about plot variability is required, the REML estimates ignore units with missing values to give the correct estimate. This type of slightly unbalanced multi-stratum example where the data can be analysed by REML or (with trickery) by ANOVA is useful to demonstrate that the techniques give similar answers, but that the exact REML estimates are more appropriate.

Acknowledgement

The work on which this paper is based was carried out while the author was funded by the UK Overseas Development Administration. Thanks are due to Sue Welham for her help in interpreting the output from REML.

Reference

Martin, P.J. and Poultney, R. (1992). Survival and Growth of Clove Seedlings in Zanzibar. 1. Effect of mulching and Shade Crops. *Tropical Agriculture*. Vol. 69. No 4. pp 365-373.

Postscript

All of the figures were produced using Genstat graphics.

