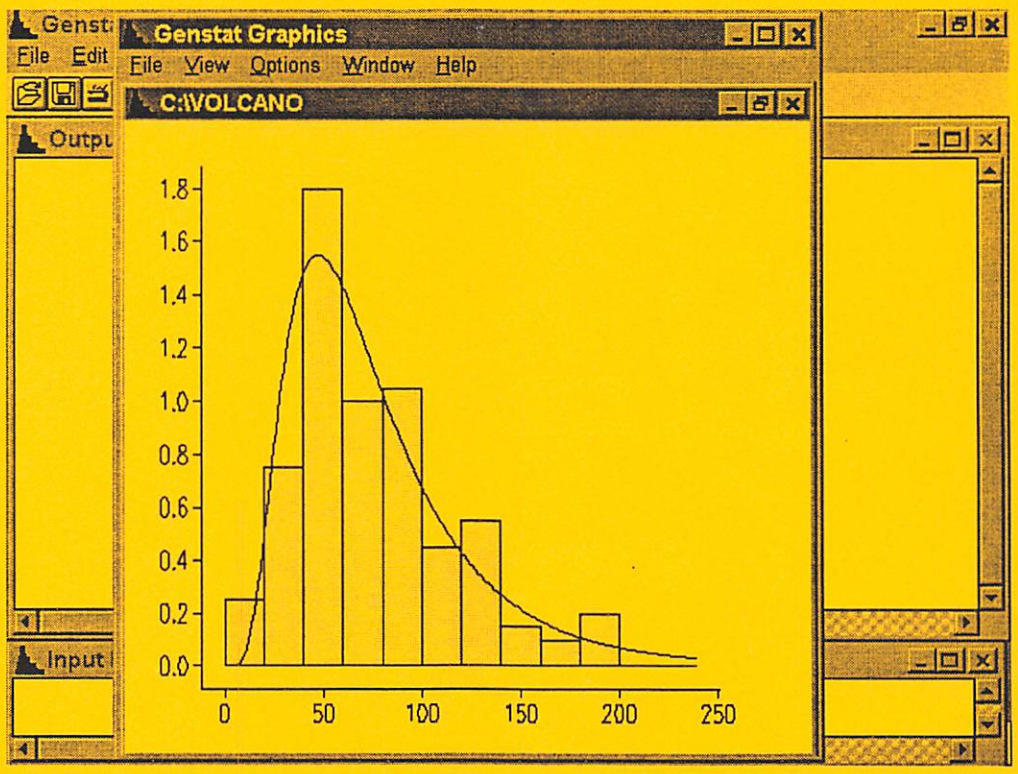
NAG®

GENSTAT

Newsletter

Issue 33



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

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

NAG Bulletin Board:

http://www.nag.co.uk/

Genstat Newsletter

Issue 33

NP3168

May 1997

Contents

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

Editorial

The editors would like to take the opportunity to apologise to Genstat users for the delay in producing Newsletter 33; the arrival of Genstat for Windows and the Second Edition of Genstat for Windows placed additional demands on the editors' time, resulting in a much later release date than desired. We are pleased to report however that the new Windows™ interface has been very well received and for those of you who have not yet been introduced to the new system, an on-line demonstration can be viewed on the internet at

http://www.nag.co.uk/stats/TT/g532demo/win2ed.html.

Such a delay is unlikely to be repeated and in fact the editors are pleased to announce that the production of Genstat Newsletter 34 is already well advanced. The future of the Newsletter beyond issue 34 is currently under discussion. It is likely that the Newsletter of present will continue in a similar vein as a once yearly technical journal, offering the user shared experiences, interesting applications and helpful user code that have become the hallmark of the current Genstat Newsletter. As is the norm, supporting material will continue to be placed on the NAG website. In order to cater for the full range of Genstat users, the technical journal will be supplemented by a more lively and more frequently occurring statistical bulletin from NAG. The purpose of this bulletin will be to help present Genstat news, facilities, user stories and applications in a less technical, and more readable fashion.

As for this issue, the articles include one of the papers and conference report from the Genstat 96 conference in Adelaide, an interesting application of Genstat within industry, and various articles centring on the modelling of different kinds of data. As usual, the code for any procedure listed in any of the articles, may be found on the NAG bulletin board.

Genstat 96, December 4–6, Adelaide

Bronwyn Harch
Biometrics Unit
CSIRO Mathematical and Information Sciences
PMB No. 2
GLEN OSMOND, SA 5064, Australia

Genstat 96, the Australasian meeting of Genstat users, was held at the Waite Agricultural Institute, Urrbrae, Australia (a campus of The University of Adelaide). The Genstat 96 conference organising committee included Chris Brien (The University of South Australia), Ray Correll (CSIRO), Trevor Hancock (The University of Adelaide), Rita Middleberg (The University of Adelaide), Angela Reid (CSIRO) and Jeff Wood (CSIRO).

Eighty Genstat users, including statisticians/biometricians and biological scientists, came mostly from the Australasian region (Australia and New Zealand), but the current developers of the statistics package travelled from IACR-Rothamsted, Harpenden, UK and other participants from Belgium, Brasil and the UK. The theme of the conference was the use of Genstat for spatial statistics and longitudinal data. Invited speakers included Roger Payne and Peter Lane from IACR-Rothamsted (UK), Granville Tunnicliffe-Wilson from Lancaster University (UK), Jeff Wood from CSIRO (Canberra), David Baird from AgResearch (New Zealand) and Brian Cullis from NSW Department of Agriculture (Wagga Wagga). Their talks generally focused on the use of Genstat for specific statistical applications, for example, the design of experiments, spatial analysis, generalized linear models, time series, generalized additive models, spreadsheets in Genstat and repeated measures. Most contributed papers gave some form of reference to Genstat, but mainly provided interesting insights into various biometrical projects that the speaker had been involved with.

There were three particular highlights of the conference:

- Whilst opening the conference, Graham Wilkinson gave insights into how he and John Nelder came to develop the first module of Genstat.
- The conference dinner at Warrawong Sanctuary in the Adelaide Hills was truly an event to remember. After a BBQ dinner, the founder of the wildlife sanctuary, the controversial Dr John Wamsley, took the group on a dusk walk to hear and see Australia's native wildlife. Besides learning about Australian native flora and fauna, the group were also given a very serious but amusing spiel on Dr Wamsley's attitude to the conservation of wildlife in Australia. This included not only his very negative attitude towards cats, but also his very highly developed theory about platypus sightings and their apparent significant correlation to the "vibes" visitors to the sanctuary emit. We didn't see any platypus!?!
- The demonstration of defining a random event by Jeff Wood was also a great "Wake up" call to himself and the audience. Being the Chair of the session, Jeff somehow managed to select a chair in full view of the audience which was poised to collapse during the session. Thankfully Jeff reports he is OK.

The other main highlights included the three separate workshops held prior to the conference. On Friday 29 November, Peter Lane started the proceedings with a Genstat for Windows Introductory Course. Up until earlier this year, Genstat had only been available as a command line language, but now Genstat is largely available in a "menu driven environment". New users to Genstat will certainly appreciate this new version of Genstat because it makes the learning curve less steep! Granville Tunnicliffe-Wilson delivered a workshop about Modelling Dependence between Time Series on Monday 2 December. Granville provided not only the information needed to understand time series and forecasting methods, but also illustrated the basic directives and usage of procedures within Genstat. Wednesday 3 December dawned with Roger Payne giving insight into the analysis of repeated measurements in Genstat. The methodology and accompanying Genstat facilities were introduced from a wide range of applications. The practical sessions for all three of the workshops were extremely valuable, enabling the participants to explore real examples or investigate their own data. Course manuals and conference abstracts are available for a small cost from Ray Correll.

Overall, Genstat 96 was a successful meeting of Australasian Genstat users providing a great environment to gain further statistical knowledge, exchange ideas, meet and maintain friendships and provide future Genstat conference organisers with a tough social program to top.

Using Genstat to fit continuous actuarial distributions

D A Preece
Institute of Mathematics and Statistics
Cornwallis Building
The University
CANTERBURY, Kent CT2 7NF, UK

G J S Ross
Statistics Department
IACR Rothamsted
HARPENDEN, Herts AL5 2JQ, UK

1. Introduction

In a previous paper in the Genstat Newsletter, Preece and Ross (1995) described the use of Genstat 5 Release 3 for fitting the negative binomial distribution to data. That paper discussed in detail the sort of output that Genstat produces when the directive DISTRIBUTION is used for fitting discrete non-negative univariate distributions. We now turn our attention to the use of DISTRIBUTION for fitting *continuous* non-negative univariate distributions, especially long-tailed distributions that feature in the actuarial statistics component of the examination syllabuses of the Institute of Actuaries. Once again we aim to show how easily DISTRIBUTION can be used, even by someone who knows little of Genstat.

The distributions that we consider in this paper are the exponential, the Weibull (of which the exponential is a special case), the gamma (again having the exponential as a special case), and the log–Normal, each of these being defined in such a way as to have the entire positive x-axis as its support (i.e., defined without a location parameter).

The only Genstat directives used in this paper are UNITS, READ, PRINT, CALCULATE, SCALAR, VARIATE, FACTOR, TABLE and DISTRIBUTION.

Our illustrative example is that used by Currie (1993, Section 1.3, Table 1), the data being the values of 96 individual insurance claims, as in the following output from the first part of a Genstat run for dealing with them:

```
 1  JOB 'ACTUARIAL'
 2
 3  OUTPUT [WIDTH=76] 1
 4
 5  "Genstat analyses of data of Currie (1993, Section 1.3, Table 1)
-6   ------------------------------------------------------------"
 7
 8  "Read and print the values of the 96 individual claims."
 9  UNITS [96]
10  READ [PRINT=summary,data] Claim

11    24     26     73     84    102    115
12   132    159    207    240    241    254
13   268    272    282    300    302    329
14   346    359    367    375    378    384
15
16   452    475    495    503    531    543
17   563    594    609    671    687    691
18   716    757    821    829    885    893
19   968   1053   1081   1083   1150   1205
20
21  1262   1270   1351   1385   1498   1546
22  1565   1635   1671   1706   1820   1829
23  1855   1873   1914   2030   2066   2240
24  2413   2421   2521   2586   2727   2797
25
26  2850   2989   3110   3166   3383   3443
```

```
 27  3512   3515   3531   4068   4527   5006
 28  5065   5481   6046   7003   7245   7477
 29  8738   9197  16370  17605  25318  58524  :

    Identifier   Minimum   Mean   Maximum    Values   Missing   Skew
    Claim        24        2990     58524        96         0
 30
 31  "Calculate and print the mean and variance of the 96 claim-values."
 32  CALCULATE MnClaim = MEAN(Claim)
 33  CALCULATE VClaim = VAR(Claim)
 34  PRINT MnClaim,VClaim; 12,12; 2,0

    MnClaim      VClaim
    2989.83    47006240
```

The above output shows that input of the data was immediately followed by calculation of the sample mean and variance; these give

mean = 2990
standard deviation = 6856

As Currie (1993, p. 11) pointed out, the fact that the standard deviation (S.D.) is so much larger than the mean "suggests that there are more very large claims than the exponential distribution would predict". Accordingly, our approach in this paper is to fit the exponential distribution first, and then to see how much better a fit is provided by each of the Weibull, the gamma, and the log–Normal. Each distribution is fitted first to the raw, i.e., *ungrouped*, data. Currie (1993, p. 12) also grouped his data for the purpose of making chi-squared tests; we use his grouping to show additionally how Genstat can be used for fitting distributions to *grouped* data. For an example such as Currie's, which has just 96 units, anybody who has the raw data should, of course, use them ungrouped for the fitting. However, if an example has very many more units (e.g., the 200,000 units in a certain example from the literature), an attempt to use the raw data would come up against serious problems of data-entry and statistical computing (e.g., a long run-time, and rounding and truncation errors); grouping will then be indicated.

2. Notation

The notations used by Genstat differ, sometimes confusingly, from those used by the Institute of Actuaries (1980). We therefore set them out now, with $x > 0$ throughout (with "Institute of Actuaries" herinafter referred to as IoA).

The exponential distribution

This has a single parameter, which is a scale parameter. In Genstat, this parameter is denoted as b, so the probability density function (PDF) is taken as

$$b \exp(-bx) \qquad \text{for } b > 0,$$

and the mean and the standard deviation of the distribution are both equal to $1/b$. IoA writes the parameter b as lower-case Greek lambda. Some authors, including Currie (1993, p. 9), use $1/b$ in place of b, so that the mean and the standard deviation of the distribution are then both equal to b .

The Weibull distribution

This has two parameters. The Genstat form for the PDF is

$$cb^c x^{c-1} \exp(-(bx)^c) \qquad \text{for } b > 0,\ c > 0$$

where b is a scale parameter and c is a shape parameter. IoA and Currie (1993, p. 20) have a different use for c, with

$$c \text{ (Genstat)} = \gamma \text{ (IoA)}$$
and
$$b^c \text{ (Genstat)} = c \text{ (IoA)}$$

Some authors use the Genstat parameterisation except that, again, they use $1/b$ in place of b.

If we take the Genstat c to be 1, we obtain the exponential distribution.

For the Weibull distribution to have its standard deviation greater than its mean, the Genstat c must be less than 1, so that the mode of the distribution must be an infinite mode at $x = 0$.

Some values of the Weibull's coefficients of skewness and kurtosis are as follows; here and elsewhere in this paper, the formula for calculating kurtosis from the fourth central moment includes the term -3, and thus gives zero, not +3, as the theoretical value for the Normal distribution.

c (Genstat)	0.50	0.75	0.90	1.00	2.00	3.00	4.00
skewness	6.62	3.12	2.34	2.00	0.63	0.17	-0.09
kurtosis	84.72	15.99	8.53	6.00	0.25	-0.27	-0.25

The gamma distribution

Genstat's form for the PDF is

$$b^k x^{k-1} \exp(-bx) / \Gamma(k), \qquad b>0,\ k>0$$

where b is a scale parameter and k is a shape parameter. The IoA form, also used by Currie (1993, p. 34), is the same save that b is replaced by lower-case Greek lambda, and k is replaced by lower-case Greek alpha. Again, some authors use $1/b$ in place of b .

If we take $k = 1$, we are again back to the exponential.

For the gamma distribution the mean, in Genstat notation, is k/b and the standard deviation is $(\sqrt{k})/b$; accordingly, for the standard deviation to be greater than the mean we must have $k < 1$, and so must have a distribution whose mode is an infinite mode at $x = 0$.

The skewness of the gamma distribution is $2/\sqrt{k}$, and the kurtosis is $6/k$, so neither coefficient can be negative. Some values of these coefficients are as follows:

k	0.10	0.20	0.25	0.50	1.00	2.00	4.00
skewness	6.32	4.47	4.00	2.83	2.00	1.41	1.00
kurtosis	60.00	30.00	24.00	12.00	6.00	3.00	1.50

The log–Normal distribution

No notational problems arise so long as we recall that the statement "X has a log–Normal distribution" means that the natural logarithm of X has a Normal (i.e., Gaussian) distribution, and that the two parameters of a log-Normal distribution are those of the corresponding Normal distribution.

If we take the mean and standard deviation of the corresponding Normal distribution to be m and s respectively, and we write

$$b = \exp(m), \qquad k = \exp(s^2),$$

then b is a scale parameter of the log–Normal distribution and k is a shape parameter. In this notation, the log–Normal distribution has

$$\text{mean} = b\sqrt{k}\,, \qquad \text{S.D.} = b\sqrt{k}\,\sqrt{(k-1)};$$

accordingly, for the standard deviation to be greater than the mean, we need $k > 2$, i.e. the variance of the corresponding Normal distribution has to be greater than ln(2).

The skewness of the log–Normal distribution is

$$(k+2)\ \sqrt{(k-1)}$$

and the kurtosis is

$$k^4 + 2k^3 + 3k^2 - 6,$$

so neither coefficient can be negative. Some values of these coefficients are as follows where, for ease of comparison with preceding tables of values, we now tabulate *decreasing* values of k:

k	2.50	2.25	2.00	1.75	1.50	1.25	1.10
skewness	5.51	4.75	4.00	3.25	2.47	1.62	0.98
kurtosis	83.06	57.60	38.00	23.29	12.56	5.04	1.76

3. Using DISTRIBUTION to fit continuous non-negative univariate distributions

If ungrouped data are to be used, these should be supplied to the directive DISTRIBUTION as a variate; the identifier of the variate is specified as the first parameter of DISTRIBUTION, as in the following, where Claim is the variate:

```
DISTRIBUTION [DISTRIBUTION=exponential] Claim
```

If only grouped data are available, they should similarly be supplied as a one-way table of counts (frequencies), e.g.,

```
DISTRIBUTION [DISTRIBUTION=exponential] ObsFreqs
```

For this latter alternative, the levels of the factor that classifies the table must be, in ascending order, the uppermost values for the classes created by the grouping, except that the last level, which is not used in calculations, can be any arbitrary large value. To illustrate this, we use Currie's grouping (Currie, 1993, p. 12) of the insurance claims in his example. His classes for the grouping were

$$0 - 260\,,\quad 260 - 545\,,\quad 545 - 860\,,\quad \ldots\,,\quad 5357 - 7430\,,\quad 7430 - \infty\,,$$

so the required factor-levels can be specified as

$$260\,,\quad 545\,,\quad 860\,,\quad \ldots\,,\quad 7430\,,\quad 100000\,.$$

(As we shall see, the class-limits here were chosen so that the expected class-frequencies will be equal when an exponential distribution is fitted.) If the table is named ObsFreqs (Observed Frequencies), we are then led to the following Genstat input and output for the grouping; the reason for also forming a variate named Limits, whose values are the successive factor-levels except the last, is given below.

```
35
36  "Enter class-boundaries for grouping the claims."
37  SCALAR Limit[1...13]; \
38    VALUE=0,260,545,860,1212,1612,2073,2618,3285,4145,5357,7430,100000
39
40  "Put class-boundaries, except the first and last, into a variate"
41  VARIATE [VALUES=Limit[2...12]] Limits
42
43  "Find observed frequency f for each of the 12 classes of claims."
44  CALCULATE f[1...12] = \
45                  SUM(Claim<Limit[2...13] .AND. Claim>=Limit[1...12])
46
47  "Prepare and print table of observed frequencies for the 12 classes."
48  FACTOR [LEVELS=!(Limit[2...13])] UpLimits; DECIMALS=0
49  TABLE [CLASSIFICATION=UpLimits; VALUES=f[1...12]] ObsFreqs; DECIMALS=0
50  PRINT [ACROSS=UpLimits] ObsFreqs; FIELDWIDTH=9
```

```
          ObsFreqs
UpLimits      260       545       860      1212      1612      2073      2618
               12        18        10         8         7        10         5

UpLimits     3285      4145      5357      7430    100000
               6         6         3         4         7
```

With `Claim` as the name of the variate containing the ungrouped data, and `ObsFreqs` as the name of the table of grouped data, we could now fit the exponential distribution to the ungrouped and grouped data by, respectively, the commands

```
DISTRIBUTION [DISTRIBUTION=exponential] Claim
```

and

```
DISTRIBUTION [DISTRIBUTION=exponential] ObsFreqs
```

However, the output for ungrouped data, like that for grouped data, includes a table of observed and fitted values (the latter being expected frequencies), and a grouping into classes is required for this. By default, this table is formed for ungrouped data by dividing the data into m groups of approximately equal observed frequency, where $m = \sqrt{n}$ with n as the number of data-values. Alternatively, the `LIMITS` option of the directive `DISTRIBUTION` may be used to supply the uppermost values for each class except the last. This facility being available, we decided to carry Currie's grouping over to the output table for the ungrouped data. This is why we prepared the variate named `Limits` in the above Genstat output, and then used

```
DISTRIBUTION [DISTRIBUTION=exponential; LIMITS=Limits] Claim
```

to fit the exponential distribution to the ungrouped data. Notwithstanding the equal *expected* class-frequencies for the exponential distribution fitted to Currie's data, this grouping produces *observed* class-frequencies that range from 3 to 18 and that are therefore perhaps more variable than might be thought desirable, especially as we shall be fitting distributions other than the exponential. However, we stick to Currie's grouping, to facilitate comparison with his own results, and we leave the reader to explore other groupings, as an exercise.

4. How Genstat fits continuous non-negative univariate distributions

When the directive `DISTRIBUTION` is used, the specified distribution is fitted to the data by maximum-likelihood methodology. If the data are entered ungrouped, the full log-likelihood is used in the optimization algorithm. But when the fit is to grouped data, the computed log-likelihood is only an approximation to the full log-likelihood and the solution obtained will depend partly on where the class-limits fall. For grouped data, the fitting process uses (a) the log-likelihood based on the observed frequency in each class, and (b) the probability that an observation falls within the class-limits, this probability being computed from the difference between the values of the cumulative distribution function (CDF) at the upper and lower limits. The CDF has minimum value 0 at the lower end of the range (i.e., usually at zero for positive variables) and maximum value 1 at the upper end (usually infinity). The minimised log-likelihood is used in calculating the residual deviance, and is therefore the minimum deviance achievable for any set of values of the parameters; this minimum value will always be less than the deviance given when the full data and log-likelihood are used.

For some distributions, stable "working parameters" have to be used in the optimization algorithm (Ross, 1990, pp. 165-168), and the "defining parameters" are then evaluated by a simple transformation. An iterative Gauss-Newton method of optimization is used. Options and parameters of the `DISTRIBUTION` directive are available for dealing with convergence problems that may arise for particular data-sets, but these facilities are not dealt with in this paper.

The output from `DISTRIBUTION` first gives sample statistics, including mean and variance, coefficients of skewness and kurtosis, and *approximate* sample quartiles. When the fitting is to ungrouped data, the moments and the coefficients of skewness and kurtosis are calculated from the raw data-values. When grouped data are to be fitted, the sample statistics are all estimated from the frequencies and mid-points for each class, with no corrections for grouping. As this estimation is independent of the distribution that is subsequently to be fitted, it incorporates no requirement that the variate values should be non-negative or otherwise restricted in magnitude, so it proceeds as if the first and last of the classes are for lower and upper tails of a distribution; the values used

as notional mid-points for these two classes are extrapolated from the mid-points of the adjoining classes. For example, the notional mid-point for the right-hand tail is taken to be as far *above* the previous class-limit as the previous mid-point was *below* it. This procedure, which is not always appropriate, tends to underestimate the moments if there are large values in the upper tail.

The formula for calculating the printed kurtosis coefficient includes the term -3 (as above), and is thus equivalent to the kurtosis formula that gives zero, not +3 , as the theoretical value for the Normal distribution. The printed values of sample skewness and kurtosis are of little use in determining which long-tailed distributions should be fitted. The population values of skewness and kurtosis given in the above tables are strongly influenced by the extreme values in the upper tail, which are very infrequent but very large, and which may not be included in finite samples. The sample distributions of skewness and kurtosis are biased downwards, and for sample-sizes usually encountered in practical work the theoretical values are seldom achieved. This is illustrated in the following table pertaining to skewness and kurtosis in random samples. The table was obtained by simulations generated from rectangular random numbers, and is for Weibull distributions where the Genstat parameter c takes the values c = 2, 1 and 0.5 respectively (where c = 1 produces the exponential distribution), with progressively longer tails; sample entries in the table are based on 1000 samples of sizes 10000, 1000 and 100 respectively.

	c = 2.0	c = 1.0	c=0.5
Theoretical skewness	0.62	2.00	6.62
Median skewness (and percentage of samples with skewness less than theoretical skewness):			
n = 10000	0.62 (52%)	1.97 (60%)	6.00 (74%)
n = 1000	0.61 (53%)	1.94 (61%)	5.30 (80%)
n = 100	0.58 (57%)	1.70 (73%)	3.64 (96%)
Theoretical kurtosis	0.25	6.00	84.72
Median kurtosis (and percentage of samples with kurtosis less than theoretical kurtosis):			
n = 10000	0.23 (60%)	5.70 (65%)	65.00 (75%)
n = 1000	0.20 (58%)	5.15 (66%)	40.20 (87%)
n = 100	-0.02 (66%)	3.20 (81%)	15.40 (99%)

Whether the data are supplied to Genstat ungrouped or grouped, the approximate sample quartiles are calculated using grouped data; when the fitting is to ungrouped data, the grouping used for calculating the quartiles is that provided, whether actually or by default, by the LIMITS option of the DISTRIBUTION directive. As with the printed coefficients of skewness and kurtosis, the printed approximate quartiles are intended for diagnostic purposes only; if exact quartiles are needed for a set of ungrouped data, they should be obtained, not via the DISTRIBUTION directive, but using Genstat calculating statements such as MEDIAN(x) or the directive SORT.

After the sample statistics, the output from DISTRIBUTION gives a summary of the fit, followed by parameter estimates printed with their standard errors and correlations. The deviance and the corresponding number of degrees of freedom are printed as part of the model summary and are based on the table of fitted values; they thus depend on the choice of class-limits. The computed residuals, labelled on the output as "weighted residuals", are "deviance residuals", whose sum of squares is the printed deviance.

5. Fitting the exponential distribution to Currie's data

For fitting the exponential distribution to ungrouped sample data, the maximum likelihood estimator of $1/b$ is the sample mean (Currie, 1993, p. 10). The parameter b is therefore estimated by the reciprocal of the sample

mean. Accordingly, if we use

```
DISTRIBUTION [DISTRIBUTION=exponential; LIMITS=Limits] Claim
```

to fit the exponential distribution to the ungrouped data of Currie's example, and

```
DISTRIBUTION [DISTRIBUTION=exponential] ObsFreqs
```

to fit it to the grouped data, we obtain output as follows:

```
 51
 52  "Fit exponential distribution to the UNGROUPED claim-values,
-53   but with
-54   observed and expected frequencies calculated for classes as above."
 55  DISTRIBUTION [DISTRIBUTION=exponential; LIMITS=Limits] Claim

55.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size         96
   Mean          2989.83      Variance 47006240.00
   Skewness         6.13      Kurtosis       43.89

   Quartiles:        25%       50%       75%
                   450.0    1212.0    2840.3

*** Summary of analysis ***

Observations: Claim
   Parameter estimates from individual data values.
  Distribution: Exponential
              f(x) = b.exp(-bx), x>0, b>0
   Deviance: 21.07 on 10 d.f.

*** Estimates of parameters ***

         estimate       s.e.
b          0.0003     0.0000

*** Fitted quartiles ***

             25%         50%         75%
         860.109    2072.364    4144.728

*** Fitted values (expected frequencies) and residuals ***

   x           Number     Number   Weighted
             Observed   Expected   Residual
< 260.0            12       8.00       1.32
< 545.0            18       8.00       3.03
< 860.0            10       8.00       0.68
< 1212.0            8       8.00       0.00
< 1612.0            7       8.02      -0.37
< 2073.0           10       8.00       0.68
< 2618.0            5       8.00      -1.14
< 3285.0            6       8.00      -0.74
< 4145.0            6       8.00      -0.74
< 5357.0            3       8.00      -2.03
< 7430.0            4       8.00      -1.57
> 7430.0            7       8.00      -0.36

 56
 57  "Fit exponential distribution to the GROUPED claim-values."
 58  DISTRIBUTION [DISTRIBUTION=exponential] ObsFreqs

58.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size         96
   Mean          2115.58      Variance 5583347.50
```

```
   Skewness          1.57       Kurtosis        1.50

   Quartiles:         25%         50%         75%
                    450.0      1212.0      2840.3

*** Summary of analysis ***

Observations: ObsFreqs
    Parameter estimates from tabulated data values.
     Distribution: Exponential
              f(x) = b.exp(-bx), x>0, b>0
    Deviance: 13.24 on 10 d.f.

*** Estimates of parameters ***

          estimate        s.e.
b           0.0005      0.0000

*** Fitted quartiles ***

              25%          50%          75%
          628.904     1515.294     3030.587

*** Fitted values (expected frequencies) and residuals ***

   x              Number      Number    Weighted
                Observed    Expected    Residual
< 260.0               12       10.76        0.37
< 545.0               18       10.42        2.13
< 860.0               10       10.04       -0.01
< 1212.0               8        9.63       -0.54
< 1612.0               7        9.22       -0.76
< 2073.0              10        8.73        0.42
< 2618.0               5        8.21       -1.21
< 3285.0               6        7.62       -0.61
< 4145.0               6        6.95       -0.37
< 5357.0               3        6.13       -1.41
< 7430.0               4        5.07       -0.49
> 7430.0               7        3.21        1.83
```

A quick glance at the output for either the ungrouped or grouped data shows that the estimate of *b* and its standard error (`s.e.`) are printed with insufficient significant figures, being curtailed at 4 decimal places. This suggests that we would have done well to scale the data before the fitting, say by dividing all claims by 1000. (As the original data were in "pounds sterling", the scaling would merely change the units to "thousands of pounds sterling".) However, the output otherwise provides all that we might wish for, so we here overlook the undesirable curtailment and continue working with the unscaled data.

The analysis of the ungrouped data is essentially the same as the analysis given by Currie (1993, pp. 10-12); when the ungrouped data are fitted by Genstat, the expected frequency for each of Currie's classes is 8.0, confirming the basis of his grouping.

Use of the ungrouped data clearly indicates that an exponential distribution provides a bad fit, as Currie (1993, p. 12) showed. (He used chi-squared, not deviance, to test for goodness-of-fit, and his chi-squared value 23.0, on 10 d.f., naturally differs little from the Genstat deviance 21.07.) As Currie pointed out, the claims up to about 500 pounds are under-fitted by the fitting to the ungrouped data, whereas claims from about 2000 to 7000 pounds are over-fitted.

The difference between the deviances for the Genstat fits to the ungrouped and grouped data is particularly striking. The deviance for the grouped data is 13.24, on 10 d.f., and is the minimum achievable when the data are assumed to have an exponential distribution. For the ungrouped data, the deviance is as large as 21.07 because parameter *b* is estimated by the reciprocal of the sample mean, which depends heavily on two extremely large values in the tail, which are not evident in the grouped data. Omission of the largest value reduces the mean from 2990 to 2405, which would alter the fit considerably. The difference between the two values of the deviance is an indication of the information lost by grouping when the model does not fit the data.

6. Fitting the Weibull distribution to Currie's data

When Genstat fits the Weibull distribution, with PDF

$$c\, b^c\, x^{c-1}\ \exp(\ -(bx)^c\ )\ ,\qquad b > 0\ ,\quad c > 0\ ,$$

the stable working parameters are c and the distribution's median, namely

$$(\ln 2)^{1/c} / b,$$

this latter being estimated initially by the sample median. Genstat expects c to take a value in the range 0.1 to 5.0, and takes $c = 1$ (the value that gives an exponential distribution) as an initial estimate.

When Genstat fits the Weibull distribution to, respectively, Currie's *ungrouped* and *grouped* data, we obtain output as follows:

```
 59
 60  "Fit Weibull distribution to the UNGROUPED claim-values,
-61   but with
-62   observed and expected frequencies calculated for classes as above."
 63  DISTRIBUTION [DISTRIBUTION=Weibull; LIMITS=Limits] Claim

63.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size         96
   Mean          2989.83       Variance 47006240.00
   Skewness         6.13       Kurtosis       43.89

   Quartiles:        25%        50%        75%
                   450.0     1212.0     2840.3

*** Summary of analysis ***

Observations: Claim
Parameter estimates from individual data values.
Distribution: Weibull
              f(x) = c.(b**c).(x**(c-1)).exp(-(bx)**c), x>0, b,c>0
Deviance: 11.52 on 9 d.f.

*** Estimates of parameters ***

          estimate        s.e.      Correlations
c           0.7132      0.0510      1.0000
b           0.0004      0.0001     -0.3336  1.0000

*** Fitted quartiles ***

              25%         50%         75%
          391.210    1342.501    3548.203

*** Fitted values (expected frequencies) and residuals ***

   x            Number     Number   Weighted
              Observed   Expected   Residual
< 260.0             12      18.57      -1.63
< 545.0             18      10.75       2.01
< 860.0             10       8.72       0.42
< 1212.0             8       7.57       0.16
< 1612.0             7       6.82       0.07
< 2073.0            10       6.26       1.37
< 2618.0             5       5.87      -0.37
< 3285.0             6       5.60       0.17
< 4145.0             6       5.45       0.23
< 5357.0             3       5.45      -1.15
< 7430.0             4       5.78      -0.78
> 7430.0             7       9.17      -0.75
```

```
 64
 65  "Fit Weibull distribution to the GROUPED claim-values."
 66  DISTRIBUTION [DISTRIBUTION=Weibull] ObsFreqs

66.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size        96
   Mean          2115.58      Variance 5583347.50
   Skewness        1.57      Kurtosis       1.50

   Quartiles:       25%       50%       75%
                  450.0    1212.0    2840.3


*** Summary of analysis ***

Observations: ObsFreqs
arameter estimates from tabulated data values.
Distribution: Weibull
              f(x) = c.(b**c).(x**(c-1)).exp(-(bx)**c), x>0, b,c>0
Deviance: 8.14 on 9 d.f.

*** Estimates of parameters ***

         estimate       s.e.     Correlations
c          0.8235     0.0743     1.0000
b          0.0005     0.0001    -0.2504  1.0000

*** Fitted quartiles ***

             25%         50%         75%
         446.072    1297.721    3011.186

*** Fitted values (expected frequencies) and residuals ***

   x           Number     Number   Weighted
             Observed   Expected   Residual
< 260.0            12      16.17      -1.09
< 545.0            18      11.45       1.78
< 860.0            10       9.80       0.06
< 1212.0            8       8.72      -0.25
< 1612.0            7       7.94      -0.34
< 2073.0           10       7.28       0.95
< 2618.0            5       6.73      -0.70
< 3285.0            6       6.26      -0.10
< 4145.0            6       5.84       0.07
< 5357.0            3       5.47      -1.15
< 7430.0            4       5.15      -0.53
> 7430.0            7       5.20       0.75
```

Once again, we have a scaling problem, and once again the fit to the ungrouped data is not as good as that to the grouped data. However, for the ungrouped data, the deviance of 11.52 (9 d.f.) for the Weibull is much better than that of 21.07 (10 d.f.) for the exponential. The difference between the deviances for the fits to the ungrouped data and the grouped data is less for the Weibull than for the exponential, as the precise values in the tail are now less important. The parameter estimates for ungrouped and grouped data are closer when the model fits well, there then being less information lost from grouping.

Currie (1993, pp. 22-24) fitted the Weibull distribution to his data by the method of percentiles, with the 25% and 75% sample quartiles taken as the quartiles of the fitted distribution. He obtained these quartiles, not by an approximate method, as in Genstat, but as, respectively, the "24.25th value" = 401.00 and the "72.75th value" = 2836.75. His expected values were as in the following table:

	Observed	Expected		
		Genstat (ungrouped data)	Genstat (grouped data)	Currie (by quartiles)
0 - 260	12	18.6	16.2	17.6
260 - 545	18	10.8	11.4	11.9
545 - 860	10	8.7	9.8	10.0
860 - 1212	8	7.6	8.7	8.8
1212 - 1612	7	6.8	7.9	7.9
1612 - 2073	10	6.3	7.3	7.1
2073 - 2618	5	5.9	6.7	6.5
2618 - 3285	6	5.6	6.3	6.0
3285 - 4145	6	5.4	5.8	5.5
4145 - 5357	3	5.4	5.5	5.1
5357 - 7430	4	5.8	5.2	4.8
7430 -	7	9.2	5.2	4.8

7. Fitting the gamma distribution to Currie's data

When Genstat fits the gamma distribution, with PDF

$$b^k x^{k-1} \exp(-bx) / \Gamma(k) , \qquad b > 0 , \quad k > 0 ,$$

the procedure differs little from that for the Weibull distribution. The stable parameters are $1/k$ and the distribution's median. As the mean and the variance of the distribution satisfy

$$k = (\text{mean})^2 / \text{variance},$$

an initial estimate of k is provided by

$$(\text{sample mean})^2 / (\text{sample variance}) ;$$

an initial estimate of b can then be obtained by equating the sample median to the approximation $(k+1)/b$ for the distribution's median. (For $k = 1$, the distribution's median is $(\ln 2) / b$.)

When Genstat fits the gamma distribution to, respectively, Currie's *ungrouped* and *grouped* data, we obtain output as follows:

```
 67
 68  "Fit gamma distribution to the UNGROUPED claim-values,
-69   but with
-70   observed and expected frequencies calculated for classes as above."
 71  DISTRIBUTION [DISTRIBUTION=gamma; LIMITS=Limits] Claim

71.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size        96
   Mean          2989.83      Variance 47006240.00
   Skewness         6.13      Kurtosis       43.89

   Quartiles:        25%       50%       75%
                   450.0    1212.0    2840.3

*** Summary of analysis ***

Observations: Claim
       Parameter estimates from individual data values.
     Distribution: Gamma
```

```
            f(x) = (b**k).(x**(k-1)).exp(-bx)/Gamma(k), x>0
      Deviance: 15.40 on 9 d.f.

*** Estimates of parameters ***

         estimate       s.e.     Correlations
k          0.6257     0.0762     1.0000
b          0.0002     0.0000     0.6862  1.0000

*** Fitted quartiles ***

             25%          50%          75%
         464.529     1619.854     4071.211

*** Fitted values (expected frequencies) and residuals ***

   x           Number     Number   Weighted
             Observed   Expected   Residual
< 260.0            12      16.96      -1.27
< 545.0            18       9.39       2.49
< 860.0            10       7.86       0.73
< 1212.0            8       7.07       0.34
< 1612.0            7       6.60       0.15
< 2073.0           10       6.29       1.36
< 2618.0            5       6.12      -0.47
< 3285.0            6       6.06      -0.02
< 4145.0            6       6.11      -0.05
< 5357.0            3       6.34      -1.48
< 7430.0            4       6.94      -1.21
> 7430.0            7      10.26      -1.08


 72
 73  "Fit gamma distribution to the GROUPED claim-values."
 74  DISTRIBUTION [DISTRIBUTION=gamma] ObsFreqs

74.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size        96
   Mean          2115.58      Variance 5583347.50
   Skewness         1.57      Kurtosis        1.50

   Quartiles:        25%        50%        75%
                   450.0     1212.0     2840.3

*** Summary of analysis ***

Observations: ObsFreqs
    Parameter estimates from tabulated data values.
     Distribution: Gamma
            f(x) = (b**k).(x**(k-1)).exp(-bx)/Gamma(k), x>0
    Deviance: 9.65 on 9 d.f.

*** Estimates of parameters ***

         estimate       s.e.     Correlations
k          0.7679     0.1086     1.0000
b          0.0003     0.0001     0.7772  1.0000

*** Fitted quartiles ***

             25%          50%          75%
         469.942     1364.846     3071.173

*** Fitted values (expected frequencies) and residuals ***

   x           Number     Number   Weighted
             Observed   Expected   Residual
< 260.0            12      15.71      -0.98
< 545.0            18      10.89       1.97
< 860.0            10       9.50       0.16
< 1212.0            8       8.64      -0.22
```

```
< 1612.0              7        8.02       -0.37
< 2073.0             10        7.49        0.87
< 2618.0              5        7.04       -0.81
< 3285.0              6        6.63       -0.25
< 4145.0              6        6.24       -0.10
< 5357.0              3        5.84       -1.30
< 7430.0              4        5.38       -0.62
> 7430.0              7        4.62        1.03
```

The fits of the gamma distribution to the ungrouped and grouped data differ little from the corresponding fits for the Weibull distribution, but produce slightly larger deviances. The expected frequencies obtained from the analysis of the ungrouped data are, of course, in very close agreement with those obtained by Currie (1993, pp. 34-37) when he used the method of maximum likelihood to fit the gamma distribution to the data.

8. Fitting the log–Normal distribution to Currie's data

Only if the sample skewness is positive will Genstat fit the log–Normal distribution. If the sample skewness is negative, an automatic switch is made to the Normal distribution.

When Genstat fits the log–Normal distribution to, respectively, Currie's *ungrouped* and *grouped* data, we obtain output as follows:

```
 75
 76  "Fit logNormal distribution to the UNGROUPED claim-values,
-77   but with
-78   observed and expected frequencies calculated for classes as above."
 79  DISTRIBUTION [DISTRIBUTION=logNormal; LIMITS=Limits] Claim

79.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size        96
   Mean          2989.83      Variance 47006240.00
   Skewness         6.13      Kurtosis       43.89

   Quartiles:        25%       50%       75%
                   450.0    1212.0    2840.3

*** Summary of analysis ***

Observations: Claim
  Parameter estimates from individual data values.
  Distribution: Lognormal
              Log(X) distributed as Normal(m,s**2), X>0
  Deviance: 4.57 on 9 d.f.

*** Estimates of parameters ***

         estimate       s.e.     Correlations
m          7.0215     0.1428     1.0000
s          1.3988     0.1010     0.0000  1.0000

*** Fitted quartiles ***

              25%         50%         75%
          436.162    1120.442    2878.262

*** Fitted values (expected frequencies) and residuals ***

   x           Number     Number   Weighted
             Observed   Expected   Residual
< 260.0            12      14.22      -0.61
< 545.0            18      14.88       0.78
< 860.0            10      11.69      -0.51
< 1212.0            8       9.35      -0.45
```

```
< 1612.0              7       7.70      -0.26
< 2073.0             10       6.47       1.28
< 2618.0              5       5.57      -0.25
< 3285.0              6       4.90       0.48
< 4145.0              6       4.43       0.71
< 5357.0              3       4.15      -0.59
< 7430.0              4       4.18      -0.09
> 7430.0              7       8.46      -0.52

 80
 81  "Fit logNormal distribution to the GROUPED claim-values."
 82  DISTRIBUTION [DISTRIBUTION=logNormal] ObsFreqs

82.....................................................................

***** Fit continuous distribution *****

*** Sample Statistics ***

   Sample Size         96
   Mean          2115.58      Variance 5583347.50
   Skewness        1.57       Kurtosis       1.50

   Quartiles:        25%        50%        75%
                   450.0     1212.0     2840.3

*** Summary of analysis ***

Observations: ObsFreqs
    Parameter estimates from tabulated data values.
     Distribution: Lognormal
              Log(X) distributed as Normal(m,s**2), X>0
    Deviance: 3.94 on 9 d.f.

*** Estimates of parameters ***

         estimate       s.e.      Correlations
m          7.0359     0.1362      1.0000
s          1.3051     0.1121     -0.0404  1.0000

*** Fitted quartiles ***

             25%         50%         75%
         471.384    1136.773    2741.403

*** Fitted values (expected frequencies) and residuals ***

   x           Number     Number    Weighted
             Observed   Expected    Residual
< 260.0            12      12.40       -0.11
< 545.0            18      15.12        0.72
< 860.0            10      12.36       -0.69
< 1212.0            8      10.01       -0.66
< 1612.0            7       8.25       -0.45
< 2073.0           10       6.90        1.11
< 2618.0            5       5.88       -0.37
< 3285.0            6       5.11        0.38
< 4145.0            6       4.54        0.65
< 5357.0            3       4.16       -0.60
< 7430.0            4       4.06       -0.03
> 7430.0            7       7.21       -0.08
```

The fits of the log–Normal distribution to the ungrouped and grouped data are both excellent. Once again, the expected frequencies obtained from the analysis of the ungrouped data are, of course, in very close agreement with those obtained by Currie (1993, pp. 39-40) when he fitted the log–Normal distribution by using log-transformed data.

Of the four distributions fitted in this paper, only the log–Normal came out with a non-zero mode. (This is because the fitted Weibull and gamma distributions have, respectively, $c < 1$ and $k < 1$.) Accordingly, only the log–Normal could pick up the increase in observed frequency from the first to second class of Currie's data. For these data, this is the reason for the supremacy of the fit of the log–Normal distribution.

Acknowledgements

The authors are grateful for the Actuarial Education Service's willingness for the data of Currie (1993, Section 1.3, Table 1) to be used in this paper. IACR receives grant-aided support from the Biotechnology and Biological Sciences Research Council of the United Kingdom.

References

Currie I D (1993) *Loss Distributions* London: Actuarial Education Service.
Institute of Actuaries and Faculty of Actuaries (1980) *Formulae and Tables for Actuarial Examinations* London.
Preece D A and Ross G J S (1995) Fitting the negative binomial distribution *Genstat Newsletter* **32** 20-30.
Ross G J S (1990) *Nonlinear estimation* New York: Springer-Verlag.

Cheese making with Genstat: a case study in design of industrial experiments

Eric D Schoen
TNO Dept. of Applied Statistics
PO Box 155
2600 AD, DELFT
The Netherlands

schoen@tpd.tno.nl

1. Introduction

Industrial food production often involves a stagewise reduction of the scale on which the amount of material is processed as a whole. In cheese making for example, the contents of a milk storage tank is used to fill several curds production tanks, each curds production giving rise to the production of many individual cheeses.

A typical cheese making experiment has factors associated with whole milk storage tanks, with curds production tanks and with the production and maintenance of individual cheeses respectively. So these experiments generally have a nested error structure. By their error structures and the division of their factors over various error strata, they are split–split–plot in nature.

As an additional feature of industrial cheese making experiments, the number of factors to be investigated may be quite large. Therefore, fractional factorials are often used in this context. The combination of these factorials with split–plot experimentation poses some interesting problems, which have not received much attention in the literature. More particularly, to design a fractional experiment with three error–strata, it may be necessary to design two experiments with two error strata each. The designs are to be linked subsequently, but how to do this is not straightforward.

The use of Genstat in split–plot experimentation is well established. Applications of Genstat in fractional factorial designs appear to be less common, let alone applications in fractional split–plot designs. This paper is a case study on the latter type of experimental design. It is on a cheese making experiment carried out in 1992 in a European cheese making factory. The paper is organized as follows. In Section 2, the requirements for the particular design are formulated. Section 3 presents the construction of the design and Section 4 gives the analysis of a coded compositional characteristic of the cheeses. The analysis is based on the halfnormal plotting of Yates effects as proposed by Daniel (1959). This technique has been implemented in a procedure submitted to the Genstat procedure library. The code of the procedure is presently available on request. General information about the procedure is given in the Appendix.

2. Requirements

The most important raw material of cheese is milk. A cheese making factory needs a constant stream of milk from the farms. Upon entering the factory, the milk is stored in huge milk storage tanks. Our experiment has two factors which work on whole milk supplies.

At an appropriate time, the contents of a milk storage tank is transported to several smaller tanks in which the curds is made. Curds is a white spongy mass somewhere between solid and fluid from which cheeses are made by pressing amounts of curd together in cheese presses. Five factors of the experiment work on whole curds productions. There were three factors working on individual cheeses. This makes a total of ten factors. For economical reasons, nine of these were to be investigated at two levels only. The tenth one, a cheese factor, was considered of such an importance that four levels were required. We will temporarily ignore this complication by acting as if we have to deal with a total of 11 two–level factors.

From the above description it is clear that the error structure of the experiment is a nested one: we have milk supplies, curds productions nested within milk supplies, and cheeses nested within curds productions. We will refer to the corresponding error–strata as *milk*, *curds*, and *cheese* stratum, respectively.

Due to its error structure the experiment strongly resembles a split–split–plot experiment with two whole plot factors, five split–plot factors and four split–split–plot factors. The only difference is in the absence of replication. This is quite common in industrial experiments. In the analysis section we will see that it is still possible to make estimates of the various errors.

One of the requirements common to all experiments in industrial production is to keep the size of the investigation limited. There are lower limits on the number of milk supplies, curds productions and cheeses, however. Firstly, it is no good to try to investigate two milk factors with only four milk supplies while at the same time trying to get an estimate of error between milk supplies. This implies a lower limit of eight milk supplies.

A second lower limit concerns the number of cheeses to be studied within one curds production. In view of the four–level cheese factor, four cheeses are needed to keep the main effect of this factor wholly within the *cheese* stratum.

The lower limit on the number of curds productions within one milk supply is two. This implies a total of 64 cheeses. The investigators of the factory indicated that a total of 128 cheeses was still acceptable. We will therefore consider designs with four curds productions as well as with two productions. Table 1 gives the total degrees of freedom and the expected mean square for errors for the various error strata for both potential numbers of curds productions. Designs with more cheeses were not considered in view of an excessive number of degrees of freedom in the lower stratum; designs with more than eight production tanks had too large a financial risk to be run.

For a 128–run experiment we have to use a one–sixteenth fraction of a two–level design with 11 factors, or a 2^{11-4} design. In this design each effect is aliased with fifteen other effects. Box *et al.* (1978) give a design with a resolution of V. The worst aliasing to occur is of first–order interactions with second–order ones. The design, however, is not compatible with the blocking structure of the cheese making experiment, because the resolution V design is a full factorial design in each subset of 7 factors, while we have to study the 7 milk and curds factors in a total of 32 curds productions. A similar constraint holds for the design with 64 runs.

In view of the above requirements and incompatibilities, we have to design the experiment in four stages: (1) design a 2^7 experiment with 8 milk supplies of 4 or 2 curds productions, (2) design an experiment to study 4 cheese–factors in 16 or 32 blocks of 4 cheeses, (3) link the designs, and (4) choose 2 two–level cheese factors to construct a four–level factor.

Table 1: Total degrees of freedom and expected error mean squares for each stratum in two blocking options of the cheese–making experiment[1]

error–stratum	2 curds productions		4 curds productions	
	df	EMS (error)	df	EMS (error)
milk	7	$\sigma^2 + 4\sigma_c^2 + 2\sigma_m^2$	7	$\sigma^2 + 4\sigma_c^2 + 4\sigma_m^2$
curds	8	$\sigma^2 + 4\sigma_c^2$	24	$\sigma^2 + 4\sigma_c^2$
cheese	48	σ^2	96	σ^2

[1] σ^2, σ_c^2, σ_m^2 : variance components between cheeses, between curds productions, and between milk supplies, respectively.

3. Design

Milk–and–curds design
The problem to investigate 2 milk factor and 5 curds factors in 8 milk supplies of 2 or 4 curds productions is fairly easy to solve. For the 4–curds case, we write down a full 2^5 design in factors A, B, C, D, and E, say, with levels -1 or +1. We calculate factor F as F=ABCD, and factor G as G=ABDE. The blocking generators A, B, and ACE yield a division of effects over the two error strata as given in Table 2.

We note that the main effects of the five factors to be varied between curds productions are all estimated in the curds stratum. It can be shown that this is not possible for the design with 2 curds productions per milk supply. We therefore no longer consider this option.

Cheese design
Four cheese factors are to be investigated in 32 blocks of 4 cheeses each. This amounts to four replicates of a design confounded in 4 blocks of 4 cheeses. Denoting the cheese factors with H, J K, and L, say, it is standard to confound the effects HJK, JKL, and HL with the blocks. It is required to link these effects with effects of the milk–and–curds design. Therefore, it is not sensible to use partial confounding.

Table 2: Effects between milk supplies and between curds productions in a design for 7 factors, and 8 milk supplies of 4 curds productions each

milk	curds
A, B	C, D, E, F, G
AB	AC ... AG
CE + FG	BC ... BG
ACE + AFG	CD, DE ... DG
CDG + DEF	CG + EF
BCE + BFG	CF + EG
	CDE + DFG
	ACG + AEF
	BCG + BEF

Linkage of designs
It is required to construct a 2^{11-4} fractional factorial. Such a design needs four generators and two of them were already obtained in the milk–and–curds design. The remaining two generators must be obtained by linking two blocking generators of the cheese design with effects in the milk–and–curds design. These effects should be preferably effects from the *curds* stratum, in view of their expected precision. From Table 2, we see that there are three second–order interactions not being aliased with lower–order effects. We have also to consider the products of these interactions, because these too must be aliased with cheese effects. ACG and BCG have AB as their product. The other two subsets of second–order interactions yield main effects for products. So {ACG, BCG, AB} is the best set of milk–and–curds effects to be linked with {HJK, JKL, HL}.

It still remains to decide which of the effects in the first of the above sets is to be linked with which of the effects in the second set. It is best here to make the resulting words as long as possible. More in particular, it is not optimal to link AB with HL, because this gives a confounding of potentially important interactions. In the final design, we linked ACG with HL and AB with JKL This choice yields the required two additional generators for the design, namely, ACGHL and ABJKL. Together with their product, BCGHJK, they define the introduction

of the cheese effects in the milk–and–curds design. The products of each of these words with the words in the defining relation of the milk–and–curds design can be shown to have a length of at least five. This implies that the cheese effects do not greatly disturb the estimation of milk or curds effects.

Four–level cheese factor

A four–level factor can easily be introduced in a two–level experiment by the method of grouping (Wu, 1989). This amounts to selecting two two–level factors for use as pseudofactors (Monod and Bailey, 1992). Their product is a part of the main effect. Therefore, Wu and Zhang (1993) advise to use a single letter for this product when studying the aliasing pattern of the effects. Denoting the pseudofactors and their product as p_1, p_2, and p_3, respectively the choice of H and J as pseudo–factors gives the division of cheese–effects over the error strata as given in Table 3.

Table 3: Division of the 15 effects between the cheese–factors over the error strata

milk	*cheese*
$AB + p_2KL$	p_1, p_2, p_3, K, L
	p_2L, p_3L
curds	p_1K, p_2K
$p_1L + ACG + AEF$	KL
$p_3K + BCG + BEF$	p_1KL, p_3KL

4. Analysis

A classical tool for analysing the effects in unreplicated two–level experiments is the halfnormal plotting of Yates effects (Daniel, 1959). Yates effects of one and the same stratum have a common variance. The inactive effects are i.i.d. random variables with expectation zero and common variance. When plotted halfnormally they are on a straight line through the origin. The active effects are off–line.

The author has written Genstat procedure AYPLOT to produce the required plots, one for each error–stratum. General information on the procedure is given in the Appendix. When applied to a coded compositional characteristic of the cheeses, the procedure gives the halfnormal plots shown in Figure 1.

In Figure 1 we see three active effects, one for each stratum. These are the main effects A (*milk* stratum), and C (*curds* stratum), and the four–level main effect component J (*cheese* stratum). The halfnormal plots when these effects are omitted can be produced using an option of the above procedure. These plots (not shown) do not show further effects being active.

An analysis of variance was carried out using the active effects plus the two additional main effect components of the four–level factor (see Table 4). The ANOVA clearly shows decreasing error mean squares for lower strata. The error between milk supplies, for example, is about eight times as large as the error between cheeses. So keeping track of the various error strata prevents one from declaring too many effects of the milk stratum "significant".

Using the expected mean squares from Table 1, we may calculate added components of variance between milk supplies and between curds productions, respectively. Both are statistically significant. We conclude that keeping track of the various error strata pays off in additional information on the random variation.

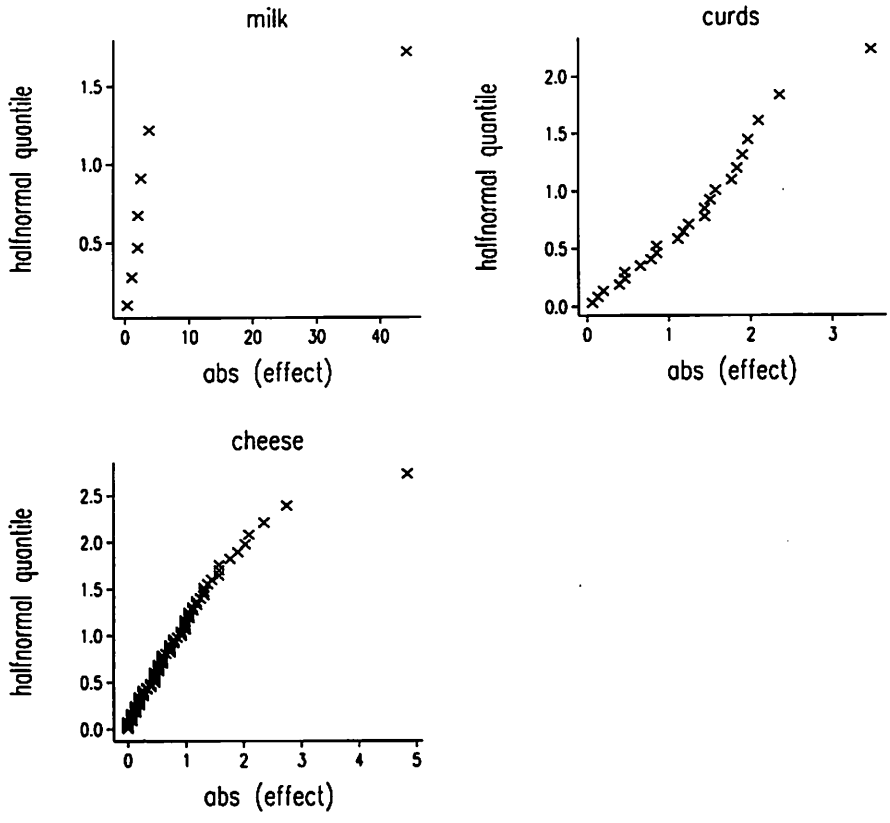


Figure 1: Halfnormal plots of the Yates effects in each error stratum

Table 4: Analysis of variance for coded compositional characteristic of the cheeses

source of variation	df	mean square	added component of variance
A	1	62419	
error between milk supplies	6	165	27.25
C	1	387	
error between curds	23	56	8.25
HJ	3	329	
error between cheeses	93	23	23

References

Box G E P, Hunter W G, and Hunter J S (1978) *Statistics for Experimenter* John Wiley, New York.

Daniel C (1959) Use of Half–Normal Plots in Interpreting Factorial Two–Level Experiments *Technometrics* **1** 311-341.

Monod H and Bailey R A (1992) Pseudofactors: normal use to improve design and facilitate analysis *Applied*

Statistics **41** 317–336.
Wu C F J (1989) Construction of 2^m4^n Designs via a Grouping Scheme *Annals of Statistics* **17** 1880-1885.
Wu C F J and Zhang R (1993) Minimum Aberration Designs with Two–Level and Four–Level Factors *Biometrika* **80** 203-209.

Appendix: Genstat procedure AYPLOT

Daniel (1959) shows how contrasts from two–level experiments in single or fractional replication can be evaluated through half–normal plotting. Box *et al.* (1978) emphasize normal plotting of the Yates effects. They suggest making separate plots for each error stratum.

Yates effects of one and the same error stratum have a common variance. When there is sparsity of effects and normality of error, most effects will come from a normal distribution with zero mean and unknown variance. Making (half)normal plots will separate the few active effects from the inactive ones.

AYPLOT calculates Yates effects from a two–level experiment. The effects to be extracted are specified through a TREATMENTSTRUCTURE setting given in advance. They are grouped according to the error strata as given by a previous BLOCKSTRUCTURE setting. Normal or halfnormal plots, according to option PLOT, are made in either line–printer or high–resolution quality (option GRAPHICS). Through option LAST the number of largest effects to be omitted from halfnormal plots can be specified (the option does not work with normal plots). Alternatively, option STRATUM can be set to those strata the Yates effects of which we wish to be plotted. The titles of the plots can be provided with option TITLE. There are three other options, FACTORIAL, PRINT, and CHANNEL, which are as for ADISPLAY. Note, however, that effects are printed as Yates effects, and that CHANNEL also bears on the line–printer graphics.

The data variates are specified with the Y parameter. Parameter EFFECT holds one variate of effects, sorted from small to large, for each error stratum. Effects are either the usual Yates effects (`PLOT=normal`) or their absolute values (`PLOT=halfnormal`).

Options: STRATUM, FACTORIAL, PRINT, PLOT, GRAPHICS, TITLE, CHANNEL, LAST
Parameters: Y, EFFECT

Options

STRATUM = *formula*	Error strata the Yates effects of which are to be plotted. If unset, plots for all strata are made
FACTORIAL = *scalar*	Limit for factorial expansion of TREATMENT formula; default 3
PRINT = *strings*	Which anova output to print, as in ANOVA; default `aovtable, effects`
PLOT = *string*	Whether to make halfnormal or normal plots (`halfnormal, normal`); default `halfnormal`
GRAPHICS = *string*	What type of graphics (`highresolution, lineprinter`); default `highresolution`
TITLE = *strings*	Separate titles for each of the plots
CHANNEL = *scalar*	What channel to use for anova and line-printer output; default 1, i.e., the current output channel
LAST = *scalars*	How many of the largest effects to withold from each of the halfnormal plots; default 0

Parameters

Y = *variates*	Data to be analysed
EFFECT = *pointers*	To save one variate with sorted Yates effects for each error stratum

Description of the method

AYPLOT accesses the current BLOCKSTRUCTURE and TREATMENTSTRUCTURE settings. If the STRATUM option is

unset, its task is to produce plots or effects of each of the strata. It therefore checks whether all strata are set explicitly. If this is not the case it augments the current BLOCKSTRUCTURE with a bottom stratum using procedure AFUNITS. If no BLOCKSTRUCTURE is set, it generates an explicit Units stratum and sets the BLOCKSTRUCTURE and STRATUM options to this stratum.

Yates effects for each stratum are saved with AKEEP. They are ordered and plotted against either normal or halfnormal quantiles. Normal quantiles are calculated as

q_i = NED((*i*-0.375)/(*n*+0.25)) i=1...n

Halfnormal quantiles are calculated as

q_i = NED(0.5+(*i*-0.375)/(*n*+0.25)×2) i=1...n

Code of procedure

```
PROCEDURE[RESTORE=blockstructure,treatmentstructure,asave]'AYPLOT'

"
Eric D. Schoen,
TNO Dept. of Applied Statistics,
PO Box 155,
2600 AD  Delft,
the Netherlands

email schoen@tpd.tno.nl

July 7, 1995.

Makes halfnormal or normal plots of Yates effects for all error strata
or a single one in a two-level experiment.Option LAST specifies the number of
largest values to be left out in halfnormal plots. The ordered effects can be saved
with parameter EFFECTS.
"

OPTION NAME=\
  'STRATUM',    "(I: formula, default all) error strata the Yates effects of
                 which     are to be plotted" \
  'FACTORIAL', "(I: scalar, default 3) limit for factorial expansion of
                 TREATMENT formula." \
  'PRINT',      "(I:string, default !T(aov,effect)) which anova output to
                 print, as in ANOVA" \
  'PLOT',       "(I: string {halfnormal, normal} default halfnormal) whether
                 to make halfnormal or normal plots" \
  'GRAPHICS',   "(I:\ string {highresolution, lineprinter} default highreso)
                 which type of graphical device to use" \
  'TITLE',      "(I: text) separate titles for each of the plots" \
  'CHANNEL',    "(I: scalar, default 1) what channel to use for anova and
                 line-printer output" \
  'LAST';       "(I: scalar, default 0) to remove LAST greatest effecst from
                 halfnormal plots; does nothing for normal plots" \
   MODE=f,v,4(t),2(v);\
   NVAL=1,1,!(1...9),2(1),*,1,*;\
   VALU=2(*),!T(AOVTABLE,INFORMAT,COVARIAT,EFFECTS,RESIDUAL,CONTRAST,MEANS,\
       %CV,MISSINGV),!T(HALFNORM,NORMAL),\
        !T(LINEPRIN,HIGHRESO),*,!(1...5),*;\
   DEFAULT=*,3,!T(aov,eff),'HALFNORM','HIGHRESO',*,1,0;\
   LIST=no,no,yes,no,no,yes,no,yes

PARAMETER NAME=\
  'Y',          "(I:variate) data" \
  'EFFECT';     "(O:pointer) holds variate(s) with ordered Yates effects for
                 specified strata"\
   MODE=p;\
  SET=yes,no;\
   DECLARED=yes,no;\
  TYPE=!T(VARIATE),!T(POINTER);\
  PRESENT=yes,no
```

```
"
assess details of STRATUM option and BLOCKSTRUCTURE setting
"

GET [SPECIAL=special]
GETATT [ATT=present] STRUCT=special['blockstructure']; P
CALC T = UNSET(STRATUM)
CALC index=2*P['present']+T+1

CASE index
   EXIT [CONTROL=procedure ; EXPLANATION=!T(\
      '*** any setting of the STRATUM option is incompatible with \
an unset BLOCKSTRUCTURE')]
OR
"
generation of a Units stratum if BLOCKSTRUCTURE directive and
STRATUM option are both unset; equating Units to BLOCKS and STRATUM
"
  DUPLICATE OLD=Y; NEW=dup
  REST dup
  CALC nval=NVAL(dup)
  FACT [LEV=nval; VAL=1...nval] Units
  FORMULA [VAL=Units] form
  ASSIGN form; STRATUM
  FORMULA [VAL=Units; MOD=yes] special['blockstructure']
OR
"
if BLOCKSTRUCTURE and STRATUM both are set no extra measures are
needed
"
  EXIT      [CONTROL=case]
OR
"
check whether all strata are set explicitly; if not:
generation of bottom stratum, modification of BLOCKS,
and setting of STRATUM
"
  AFUNITS   [BLOCKSTRUCTURE=#special['blockstructure']] Units
    IF NLEV(Units).GT.1
     FORMULA [VAL=(#special['blockstructure'])/Units] form
     ASSIGN form; STRATUM
     FORMULA [MOD=yes; VAL=#form] special['blockstructure']
   ENDIF
ENDCASE


"
making separate formulas for the error strata
"
FCLASS [NTERMS=nstrat] TERMS=#STRATUM
FCLASS #STRATUM; OUTTERMS=blterm[1...nstrat]

"
calculating # df for the various error strata
"
BLOCK #special['blockstructure']
TREAT
ANOVA [PRIN=*] Y
AKEEP #STRATUM; DF=df3[1...nstrat]

TREAT #special['treatmentstructure']
ANOVA [PRIN=*; FACT=FACTORIAL; TWOLEV=Yates] Y

"
printing anova results
"
CALC i=NOBS(PRINT)
IF i.GT.0
   ADISPLAY [PRIN=#PRINT; TWOLEV=Yates]
   ENDIF

"
calculating # effects in each stratum
"
AKEEP #STRATUM; DF=df4[1...nstrat]
CALC neff[1...nstrat]=df3[]-df4[]
```

```
"
creating temporary variate to hold saved effects
variate has conservative length equal to number of treatmentterms
"
FCLASS [NTERMS=ntterms; FACTORIAL=FACTORIAL] #special['treatmentstructure']
VARI [NVAL=ntterms] temp

ASSIGN [METHOD=preserve] Effect; EFFECT
CALC meth=1+MAX(PLOT.IN.!T(HALFNORM))+2*MAX(PLOT.IN.!T(NORMAL))

"
saving effects in temporary variate, transposing non missing
elements to variate of correct length, ordering of effects,
creating (half)normal quantiles, saving of ordered effects
"
FOR n=1...nstrat; str=blterm[]; nn=neff[]; lst=#LAST
   AKEEP [FACT=FACTORIAL; STRATUM=#str; SUPPRESS=yes; TWOLEV=Yates]\
                              #special['treatmentstructure'];\
                              EFFECTS=eff[1...ntterms]
   EQUA OLD=eff; NEW=temp
   REST temp; temp.NE.C('*'); SAVE=savedum
   VARI [NVAL=nn] effct
   CALC effct=temp$[savedum]
   REST temp
   IF meth.EQ.2
      CALC nn=nn-lst
      CALC effct=ABS(effct)
      CALC Quantile[n]=NED(0.5+(!(1...nn)-0.375)/(2*(nn+0.25)))
      SORT effct
      VARI [NVAL=nn] EFFECT[n]
      CALC EFFECT[n]$[1...nn]=effct$[1...nn]
   ELSIF meth.EQ.3
      CALC Quantile[n]=NED((!(1...nn)-0.375)/((nn+0.25)))
      SORT effct
      VARI [NVAL=nn] EFFECT[n]
      CALC EFFECT[n]=effct
   ENDIF

   DELETE [REDE=yes] effct
ENDFOR

CALC plt=1+MAX(GRAPHICS.IN.!T(HIGHRESO))+2*MAX(GRAPHICS.IN.!T(LINEPRIN))

CALC index=2*(plt-2)+(meth-1)
IF (plt.EQ.1).OR.(meth.EQ.1)
   EXIT [CONTROL=procedure ;\
   EXPLANATION='*** no proper setting of PLOT or GRAPHICS option ***']
   ENDIF

IF UNSET(TITLE)
   TEXT [VAL=#nstrat(' ')] title
   ASSIGN title; TITLE
   ENDIF

IF NVAL(TITLE).NE.nstrat
  EXIT [EXPL='*** number of separate titles does not equal number of\
 strata'; CONTROL=procedure]
  ENDIF

"
plotting of halfnormal/normal plots in line-printer/highresolution
quality
"
CASE index
      IF nstrat.EQ.1
         AXES 4; XTITLE='abs (effect)'; YTITLE='halfnormal quantile'
         DGRAPH [TITLE=#TITLE; KEYWINDOW=0;\
                 WINDOW=4] Quantile[1]; EFFECT[1]
      ELSE
         AXES 5...8; XTITLE='abs (effect)'; YTITLE='halfnormal quantile'
         FOR yvar=Quantile[]; xvar=EFFECT[]; win=5...8;\
             tit=#TITLE; scrn='clear',3('keep')
            DGRA [TITLE=tit; KEY=0; WINDOW=win; SCREEN=#scrn] yvar; xvar
         ENDFOR
      ENDIF
OR
```

```
      IF nstrat.EQ.1
         AXES 4; XTITLE='effect'; YTITLE='normal quantile'
         DGRAPH [TITLE=#TITLE; KEYWINDOW=0;\
                 WINDOW=4] Quantile[1]; EFFECT[1]
      ELSE
         AXES 5...8; XTITLE='effect'; YTITLE='normal quantile'
         FOR yvar=Quantile[]; xvar=EFFECT[]; win=5...8;\
             tit=#TITLE; scrn='clear',3('keep')
            DGRA [TITLE=tit; KEY=0; WINDOW=win; SCREEN=#scrn] yvar; xvar
         ENDFOR
      ENDIF
OR
      FOR yvar=Quantile[]; xvar=EFFECT[]
         GRAPH [CHAN=CHANNEL; TITLE=#TITLE; YTITLE='halfnormal quantile';\
            XTITLE='abs (effect)'] yvar; xvar; DESC=' '
      ENDFOR
OR
      FOR yvar=Quantile[]; xvar=EFFECT[]; tit=#TITLE
         GRAPH [CHAN=CHANNEL; YTITLE='normal quantile';\
            XTITLE='effect'; TITLE=tit] yvar; xvar; DESC=' '
      ENDFOR
ENDCASE
ENDPROCEDURE
```

Example Program

```
PRINT !T('Example of how to use procedure AYPLOT:',\
         'a half-fraction of a 2**5 design;',\
         'data from Box, Hunter, and Hunter (1978):',\
         '''Statistics for Experimenters'', p. 379',\
         '(normal plot on p. 380).');\
         JUSTIFICATION=left
UNITS [NVALUES=16]
FACTOR [LEVELS=!(-1,1)] feedrt,catalyst,agitrt,temp,conc
GENE temp,agitrt,catalyst,feedrt
CALC conc=feedrt*catalyst*agitrt*temp
READ %react
56 53 63 65 53 55 67 61 69 45 78 93 49 60 95 82:
TREATMENT feedrt*catalyst*agitrt*temp*conc
AYPLOT [TITLE='% reacted'; PLOT=normal; PRIN=eff] Y=%react
PRINT !T('to demonstrate handling of various error strata',\
         'interactions temp x agitrt and temp x catalyst',\
         'are used to define four blocks'); JUSTIFICATION=left
FACTOR [LEVELS=!(-1,1)] BD,CD
CALC BD,CD=catalyst,agitrt*temp
FACTOR [LEVELS=!(-3,-1,1,3)] Blocks
CALC Blocks=2*BD+CD

BLOCKS Blocks
TREATMENT feedrt*catalyst*agitrt*temp*conc
AYPLOT [TITLE='between blocks','within blocks';PRIN=eff] Y=%react; EFFECT=eff
PRIN eff[]
```

Fitting an ordinal regression model with random effects using composite link functions and REML

Steven Candy and Graham Wilkinson
Forestry Tasmania
Box 207B HOBART 7001
Tasmania, Australia

1. Introduction

As part of a study of genetic variation in *Eucalyptus obliqua*, open-pollinated seed collected from 12 mother trees in each of 4 populations (gully, midslope, plain, ridge) was planted in a randomized block experiment at each of the 4 sites from which seed was collected (= populations). At each site, the 4 blocks each contained 48 plots. Each plot consisted of six trees which were progeny of the same mother tree (called a family here). The location of the 12 family plots from each population were randomized within each block. Considering here just a single site (=midslope) the amount of leaf spot caused by the fungus *Mycosphaerella cryptica* on each tree was classified into 5 ordinal classes: 0–10%, 10–20%, 20–30%, 30–75%, >75%. These classes represent an ocular estimate of the percentage of leaf lamina for the whole tree covered by necrotic tissue which appears as small spots. The response variable is the number of trees on a plot in each of the above classes. Population (POPN) and block (BLOCK) are considered fixed effects while family (FAMILY) within population as a random effect. The main interest is in the effect of population on the prevalence of leaf spot. Also the amount of *between* compared to *within* family variability in prevalence of leaf spot is of interest.

These data were analysed in three ways. First and most simply, the class mid-points were used in a weighted linear mixed model analysis using the REML directive and weights equal to the 960 (=5×48×4) counts of trees across ordinal classes by plots within blocks by blocks. Second, the counts were considered a multinomial response with class probabilities determined from a logistic distribution for percentage leaf spot with the nominal class intervals dividing up this percentage scale (i.e., cut-points of 10, 20, 30, 75). The mean of the logistic was determined by the fixed and random effect models. To fit this model in Genstat, the method of composite link functions (Thompson and Baker (1981), Roger (1983)) is combined with the REML directive in a similar way to the implementation of Schall's algorithm (Schall 1991) for the generalized linear mixed model used in Genstat's GLMM procedure. Apart from the parameters for fixed and random effects, a single parameter, β, is estimated with the variance of the logistic distribution given by $\pi^2/(3\beta^2)$. The third method, described here in detail, is a generalisation of the second method whereby the nominal cut-points are used as initial values in the estimation of cut-point parameters. This method corresponds to fitting a random effects, proportional-odds ordinal regression model to the class tree counts. The general outline of the algorithm for this last model is given in Candy (1997) so here the Genstat procedures and some extra detail on practical considerations in model fitting, such as aliasing, are given. Unfortunately the Genstat procedures described are not completely general since they allow only a single random effect and are set up for 5 ordinal classes.

A fully general procedure called CLASS, by Keen, is given in the GLW-DLO Procedure Library (Goedhart and Tissen (1996)). The method used in CLASS (Keen and Engel 1997) is very similar to that used here. CLASS estimates the cut-points as nonlinear parameters using a Gauss–Newton method with the respective derivatives included as random effects covariates with very large fixed variances (e.g. 1000) in the REML step with the estimated random effects then used to update the cut-point parameter estimates. Here the derivatives are included as fixed effects in the REML step (Canady 1997). Also, we estimate the "units" variance component in REML whereas CLASS fixes this at 1. When we fixed this component at 1, the results were within rounding error, identical to those obtained in CLASS, with the exception of REML's deviance, which was different and had its degrees of freedom reduced by 3 in our method due to the estimation of the cut-points as fixed effects.

2. The fitting algorithm

The model for the class probabilty j is given by

$$Pr(class=j)=\frac{\exp(\beta_j+\eta))}{1+\exp(\beta_{j-1}+\eta))}-\frac{\exp(\beta_{j-1}+\eta))}{1+\exp(\beta_{j-1}+\eta))}\ ;\ j=2,...,4$$

where the cut-points are for the 5 classes are given by $\beta'=(\beta_1, \beta_2, \beta_3, \beta_4)$ (where $0<\beta_1<\beta_2<\beta_3<\beta_4$) and η comprises fixed and random effects. Calculation of the tail probabilities (j=1,5) for the first and last classes is discussed below.

The method of fitting composite link function generalized linear models involves constructing a "working" response variable, "working" linear predictor, and iterative weights in the same way as those for the standard GLM iteratively weighted least squares (IWLS) algorithm (McCullagh and Nelder 1989). The adaption of this algorithm for the case of composite link functions simply involves constructing "working" predictor variables for both fixed and random effects. Further simplification is achieved if the adaption given by Roger (1983) is used whereby the "working" versions of the predictor variables do not have to be constructed if they do not change across link functions, for example η here. For the leaf spot data, "working" values of the fixed terms in POPN*BLOCK and random term POPN.FAMILY are not required to calculate the working linear predictor using a method described by Candy (1985). A pair of dummy variables, corresponding to upper and lower class limits, is used to reference each cut-point. Since the upper limit dummy variable occurs in the first logit link and the lower in the second link, "working" predictor variables corresponding to each pair of dummy variables are required.

One cut-point parameter is aliased with the grand mean in η, so the pair of dummy variables for the first cut-point is replaced here by a vector of ones (i.e., explicity fitting the grand mean) and the remaining cut-points are redefined as

$\beta^{*T}=(\beta_2^*, \beta_3^*, \beta_4^*)$ where $\beta_2^*=\beta_2-\beta_1$; $\beta_3^*=\beta_3-\beta_1$; $\beta_4^*=\beta_4-\beta_1$.

The deviance (conditional on the estimated random effects) for the multinomial response can be calculated using the Poisson deviance as long as the estimated probabilities sum to unity. To do this the tails of the logistic distribution must be calculated. This is done using one of the dummy predictor variables for the cut-point parameters which is given a negative value corresponding to the first class so that a cumulative probability of zero (or numerically very close to zero) is obtained and a positive value for the last class ensures this probability is numerically very close to unity. Care must be taken to give these arbitrary negative and positive values large enough absolute values (taking into account the scaling by the corresponding β) to calculate the tails of the distribution but no so large as to cause numeric overflow.

In Genstat version 5.3 the option in the VCOMP directive not to automatically adjust covariates (here working predictor variables) for their mean should be used, ie., CADJUST=no.

3. Analysis of leaf spot data

The 960 counts (N) in the 5 percentage classes and plot totals (NT) were obtained (note that there were some missing values so NT may be less than 6) from the original data. The factors for BLOCK, FAMILY, and CLASS were generated using the order in N.

The values for factor POPN were obtained using the relationship between families and populations in the original data set (Genstat code not shown).

```
Job 'leaf spot at midslope site'
open 'midgroup.dat' ; 2
units[960]
```

```
" grouped data "

FACTOR[levels=48] FAMILY
FACTOR[levels=4] BLOCK
FACTOR[levels=4 ; LABELS=!t(gully,mid,plain,ridge)] POPN
FACTOR[levels=5] CLASS

READ[CHAN=2 ; SKIP=* ; END=*] N,NT,POPN ; frep=*,*,lev

GENERATE BLOCK,FAMILY,CLASS

" set up  for Procedure GLMORL to fit mixed ordinal (proportional odds) regression
model "

" set up initial values for GLMORL"

" (Note: some redundancies in data structures e.g. CL[1] and CU[1] are not used here
but conversion to the method which does not estimate cut-point parameters but uses the
nominal values is then made easier)
"
VARIATE WXS[1...5],CU[1...5],CL[1...5],REFFECT,WT,GM
CALC GM=!(960(1))
EXPRESSION LIN[1...5] ; \
   VALUE=!E(LP1=REFFECT+A[2]*CU[2]+A[3]*CU[3]+A[4]*CU[4]+CU[5]), \
         !E(LP2=REFFECT+A[2]*CL[2]+A[3]*CL[3]+A[4]*CL[4]+CU[5]), \
         !E(LP1=LP1*(LP1.LE.20)+21*(LP1.GT.20)), \
         !E(LP2=LP2*(LP2.LE.20)+20*(LP2.GT.20)), \
         !E(FITTEDVALUES=NT*(EXP(LP1)/(1+EXP(LP1))-EXP(LP2)/(1+EXP(LP2))) \
                        +(NT.EQ.0))

variate PAR ; values=!(1.0,2.0,3.0,7.5)
SCALAR A[1...4]
EQUATE OLD=PAR ; NEW=!P(A[1...4])
" set up CU, CL  "
" use CU[5] to store the fixed effect terms in the linear predictor"
CALC CU[5]=!(960(-3)) & REFFECT=!(960(0)) & WT=(NT.GT.0) & \
     CU[1]=!((1,0,0,0,0)192) & CU[2]=!((0,1,0,0,0)192) & \
     CU[3]=!((0,0,1,0,0)192) & CU[4]=!((0,0,0,1,3)192)  &  \
     CL[1]=!((0,1,0,0,0)192) & CL[2]=CU[3] & CL[3]=!((0,0,0,1,0)192) & \
     CL[4]=!((-3,0,0,0,1)192)
CALC #LIN[1] & #LIN[2] & #LIN[3] & #LIN[4] & #LIN[5]
CALC FD1=EXP(LP1)/(1+EXP(LP1))**2 & FD2=EXP(LP2)/(1+EXP(LP2))**2
CALC WXS[1]=(NT*(NT.GT.0)+(NT.EQ.0))*(FD1-FD2) & DERIVATIVE=1/WXS[1]
CALC WXS[2]=(FD1*CU[1]-FD2*CL[1])/(FD1-FD2)
CALC WXS[3]=(FD1*CU[2]-FD2*CL[2])/(FD1-FD2)
CALC WXS[4]=(FD1*CU[3]-FD2*CL[3])/(FD1-FD2)
CALC WXS[5]=(FD1*CU[4]-FD2*CL[4])/(FD1-FD2)
CALC LINEARPREDICTOR=REFFECT+A[2]*WXS[3]+A[3]*WXS[4] \
     +A[4]*WXS[5]+CU[5]

" run procedure GLMORL  for main effects"

GLMORL[ABSORB=FAMILY ; FIXED=BLOCK+POPN ; RANDOM=FAMILY] Y=N ; CU=CU ; \
  CL=CL ; WEIGHTS=WT ; NT=NT ; ILP=LINEARPREDICTOR ; \
  IFV=FITTEDVALUES;  WXS=WXS ; PAR=PAR ; A=A ; Z=Z

"STOT checks that the sum of the fitted values is correct "


        ITER         TDEV         PDEV         STOT
       1.000         1347            0        933.0
       2.000        836.0        37.92        933.0
       3.000        722.8        13.54        933.0
       4.000        708.9        1.916        933.0
       5.000        709.3    -0.05207        933.0

 "cut-point parameter estimates (PAR) and their standard error (SEPV)"
        PAR         SEPV
 -3.583985     0.268667
  2.255788     0.122405
  4.449117     0.148875
  6.952624     0.228215

  Summary statistics for FAM_R

             Number of values = 48
```

```
    Number of missing values = 0
                        Mean = 0.000
                      Median = 0.020
                     Minimum = -1.390
                     Maximum = 0.864
                       Range = 2.255
              Lower quartile = -0.404
              Upper quartile = 0.417
                    Variance = 0.264
          Standard deviation = 0.514
      Standard error of mean = 0.074

*** Estimated Components of Variance ***

                                    s.e.
FAMILY                  0.3949      0.1179
*units*                 0.8590      0.0404

*** Approximate stratum variances ***

                                            Effective d.f.
FAMILY                          3.03385          43.91
*units*                         0.858971        906.09

* Matrix of coefficients of components for each stratum *

FAMILY        5.502       1.000
*units*       0.000       1.000

*** Deviance: -2*Log-Likelihood ***

                  Deviance   d.f.
                     2607.   948

*** Table of effects for GM ***

                        1
                   -2.940

Table has only one entry: standard error      0.2644

*** Table of effects for WXS[3] ***

                        1
                    2.256

Table has only one entry: standard error      0.1224

*** Table of effects for WXS[4] ***

                        1
                    4.449

Table has only one entry: standard error      0.1489

*** Table of effects for WXS[5] ***

                        1
                    6.953

Table has only one entry: standard error      0.2282

*** Table of effects for BLOCK ***

   BLOCK          1           2           3           4
                0.5035      0.7899      0.5496      0.0000
```

```
Standard error of differences:     Average          0.1615
                                   Maximum          0.1652
                                   Minimum          0.1583

Average variance of differences:                   0.02607


*** Table of effects for POPN ***

         POPN       gully         mid       plain       ridge
                   0.0000      0.2479     -1.7216     -0.6442

Standard error of differences:     Average          0.3057
                                   Maximum          0.3095
                                   Minimum          0.3028

Average variance of differences:                   0.09345


" run GLMORL including interaction of BLOCK and POPN "

GLMORL[ABSORB=FAMILY ; FIXED=BLOCK*POPN ; RANDOM=FAMILY] Y=N ; CU=CU ; \
  CL=CL ; WEIGHTS=WT ; NT=NT ; ILP=LINEARPREDICTOR ; \
  IFV=FITTEDVALUES;  WXS=WXS ; PAR=PAR ; A=A ; Z=Z


" some output deleted "
        ITER        TDEV        PDEV        STOT
       1.000       831.0       60.60       933.0
 ...
       4.000       698.1    -0.03136       933.0

        PAR         SEPV
 -3.470686   0.321218
  2.268054   0.125045
  4.476767   0.152260
  7.001664   0.235980
```

4. Discussion

There appears to be no significant interaction between population and blocks. The populations appear to differ significantly with the greatest to least prevalence of leaf spot in the order `plain`, `ridge`, `gully`, `mid`(slope). Note that the signs of the parameter effects for POPN and BLOCK should be reversed to specify effects on the mean of the logistic distribution. There are also significant differences between blocks. The between family variance is significant with an inter-quartile range for the estimated random effects of 0.82 on the linear predictor scale compared to class intervals which are generally around 2. Using Method 2 where cut-points are not estimated but taken as the nominal values scaled by dividing by 10 (i.e., 1.0, 2.0, 3.0, 7.5), the conditional deviance was 1075.0, which is substantially greater than the corresponding value of 709.3 obtained by estimating the cut-points. The estimate of β using the nominal cut-point method was 1.593 (s.e. 0.263). If we scale up the nominal cut-points by 1.593, the class interval is 1.593 for classes 1, 2, and 3 but 7.2 for class 4. From the intervals obtained from the estimates of the cut-point parameters it appears that the nominal class intervals are underestimated for classes 2 and 3 while they are considerably overestimated for class 4. Note that we cannot make similar inferences for the first and last classes using the estimated cut-points. When the "units" variance was fixed at 1, the estimate of the FAMILY variance was decsreased to 0.364 and the multinomial deviance increased by 4.3 to 713.6. This deviance change and the estimated "units" variance of 0.86, with standard error 0.04, do not support the assumption of a "units" variance of 1.

References

Candy S G (1985) Using factors in composite link function models *GLIM Newsletter* **11** 24-28.

Candy S G (1997) Estimation in forest yield models using composite link functions with random effects *Biometrics* **53** 159-173.

Goedhart P W and Thissen J T N M (1996) *GLW-DLO Procedure Library Manual Release 3[2]* DLO-Agricultural Mathematics Group (GLW-DLO), Wageningen, The Netherlands.

Keen B and Engel B (1997) Analysis of a mixed model for ordinal data by iterative re-weighted REML *Statistica Neerlandica* (in press).

McCullagh P and Nelder J A (1989) *Generalized Linear Models (2nd ed)* Chapman and Hall, London.

Roger J H (1983) Composite link functions with linear log link and Poisson error *GLIM Newsletter* **7** 15-21.

Schall R (1991) Estimation in generalized linear models with random effects *Biometrika* **78** 719-727.

Thompson R and Baker R J (1981) Composite link functions in generalized linear models *Applied Statistics* **30** 125-131.

Appendix

Procedure GLMORL calls two other procedures GLMLINK and GLMDISTRIBUTION.

```
PROCEDURE 'GLMORL'
OPTION 'ABSORB','FIXED','RANDOM' ; MODE=p,f,f
PARAMETER 'Y','CU','CL','WEIGHTS','NT','ILP','IFV','WXS','PAR','A','Z' ; MODE=p
   GETATTRIBUTE[ATT=nlevels] ABSORB ; SAVE=NLEV
   PRINT NLEV[1]
   VARIATE[NVAL=#NLEV[1]] FAM_R ; VALUE=!(#NLEV[1](0))
   VARIATE FAM_V
   VARIATE[NVAL=4] SEPV
   CALC FAM_V = FAM_R $ [ABSORB]
   CALC Z=ILP+(Y-IFV)/WXS[1]
   GLMDISTRIBUTION Y=Y ; FITTEDVALUES=IFV ; VARIANCE=VAR ; DEVIANCE=DEV
   CALC TDEVI=SUM(DEV*WEIGHTS)
   CALC IW = WEIGHTS/(VAR/WXS[1]**2)
   MODEL[WEIGHT=IW] Z
   FIT[PR=*] WXS[3,4,5]+#FIXED
   CALC ITER=0 & GM=!(960(1))
   FOR[NTIMES=20]
      CALC ITER=ITER+1
      GLMLINK LINEARPREDICTOR=LP; FITTEDVALUES=FV ; DERIVATIVE=DER ; \
           CU=CU ; CL=CL ; NT=NT ; WXS=WXS ; PAR=PAR ; A=A ; REFFECT=FAM_V
      GLMDISTRIBUTION Y=Y ; FITTEDVALUES=FV ; VARIANCE=VAR ; \
           DEVIANCE=DEV
      CALC TDEV=SUM(DEV*WEIGHTS) & PDEV=100*(TDEVI-TDEV)/TDEVI & \
             STOT=SUM(FV*WEIGHTS)
      PRINT ITER,TDEV,PDEV,STOT
      EXIT ((PDEV.LT.-10).OR.((PDEV.GT.-0.1).AND.(PDEV.LT.0.1)).AND.(ITER.GT.1))
      CALC IW = WEIGHTS/(VAR*DER**2)
      CALC Z=(LP+(Y-FV)*DER)
      " USE REML "
      VCOMP[fixed=GM+WXS[3,4,5]+#FIXED ; absorb=ABSORB ;  \
        CONST=omit ; CADJUST=none] random=#RANDOM
      REML[pr=* ; weight=IW] Z ; SAVE=RSAVE

      " extract fixed and random effects "
      VKEEP[FITTEDVALUES=FV_R] terms=GM,WXS[3,4,5] ; \
                effects=PARR[1...4] ; SEDEFFECTS=SEP[1...4]
      VKEEP terms=#RANDOM ; effects=FAM_T

      EQUATE OLD=PARR,SEP,FAM_T ; NEW=PAR,SEPV,FAM_R
      CALC FAM_V = FAM_R $ [ABSORB]

      CALC TDEVI=TDEV

      EQUATE OLD=PAR ; NEW=!P(A[1...4])
      " use CU[5] to store the fixed effect terms in the linear predictor"
      CALC CU[5]=FV_R-FAM_V-(A[2]*WXS[3]+A[3]*WXS[4]+A[4]*WXS[5])
    ENDFOR
PRINT PAR,SEPV ; FIELD=10 ; DEC=6
DESCRIBE[SELECT=nval,nmv,mean,median,min,max,range,var,sd,sem,] FAM_R
VDISPLAY[PR=c,s,e,dev] RSAVE
ENDPROCEDURE


PROCEDURE 'GLMLINK'
PARAMETER 'LINEARPREDICTOR','FITTEDVALUES','DERIVATIVE','CU','CL', \
```

```
          'NT','WXS','PAR','A','REFFECT' ; MODE=p
  SCALAR A[1...4]
  EQUATE OLD=PAR ; NEW=!P(A[1...4])
  EXPRESSION LIN[1...5] ; \
   VALUE=!E(LP1=REFFECT+A[2]*CU[2]+A[3]*CU[3]+A[4]*CU[4]+CU[5]), \
         !E(LP2=REFFECT+A[2]*CL[2]+A[3]*CL[3]+A[4]*CL[4]+CU[5]), \
         !E(LP1=LP1*(LP1.LE.20)+21*(LP1.GT.20)), \
         !E(LP2=LP2*(LP2.LE.20)+20*(LP2.GT.20)), \
         !E(FITTEDVALUES=NT*(EXP(LP1)/(1+EXP(LP1))-EXP(LP2)/(1+EXP(LP2))) \
                        +(NT.EQ.0))
   CALC #LIN[1] & #LIN[2] & #LIN[3] & #LIN[4] & #LIN[5]
   CALC FD1=EXP(LP1)/(1+EXP(LP1))**2 & FD2=EXP(LP2)/(1+EXP(LP2))**2
   CALC WXS[1]=(NT*(NT.GT.0)+(NT.EQ.0))*(FD1-FD2) & DERIVATIVE=1/WXS[1]
   CALC WXS[2]=(FD1*CU[1]-FD2*CL[1])/(FD1-FD2)
   CALC WXS[3]=(FD1*CU[2]-FD2*CL[2])/(FD1-FD2)
   CALC WXS[4]=(FD1*CU[3]-FD2*CL[3])/(FD1-FD2)
   CALC WXS[5]=(FD1*CU[4]-FD2*CL[4])/(FD1-FD2)
   CALC LINEARPREDICTOR=REFFECT+A[2]*WXS[3]+A[3]*WXS[4]+A[4]*WXS[5]+CU[5]
ENDPROC


PROCEDURE 'GLMDISTRIBUTION'
PARAMETER 'Y','FITTEDVALUES','VARIANCE','DEVIANCE' ; MODE=p
   EXPRESSION VFN[1...2] ; VALUE= \
            !E(DEVIANCE=2*(Y*LOG(Y*(Y.GT.0)/(FITTEDVALUES*(FITTEDVALUES.GT.0) \
             +(FITTEDVALUES.LE.0))+(Y.EQ.0)) - (Y-FITTEDVALUES))), \
            !E(VARIANCE=FITTEDVALUES)
   CALC #VFN[1] & #VFN[2]
ENDPROC
```

A suite of Genstat procedures for the analysis of circular data

A J Rook
Institute of Grassland and Environmental Research
North Wyke
OKEHAMPTON
Devon EX20 2SB, UK

1. Introduction

Directional data, that is, data measured as angles, occur in a wide variety of situations. Examples of two–dimensional or circular data include wind direction and direction of animal movements. Time modulo some period may also be expressed as angles for example, the number of animals engaging in some activity in each hour of the day. Circular data cannot be analysed using standard linear methods as the algebraic structure of the circle is different from that of the line. Considerable progress has been made in deriving appropriate statistical methods for circular data, many of which are analogs of common linear methods. (see Mardia 1972; Fisher 1993). In this article, a suite of procedures is described which implements some of the simpler methods for circular data in Genstat.

2. Common input conventions

The procedures all share a common option: `UNITS=radians,degrees,hours,days`. This allows the angular data to be input using different units. By default radians are assumed. All calculations are carried out in radians after transformation. Results are back transformed and presented in terms of the original units. The settings `hours` and `days` allow for a 24–hour day and a 365–day year respectively.

The procedures also share a common input parameter: `DATA=variates`. This is used to enter the angular data as variates. Data are checked for compatibility with the setting of the `UNITS` option. No provision is made at present for the input of axial data, that is, data in which the angles represent undirected lines or axes and which are thus defined on a semi-circle. However, this can easily be dealt with using `CALCULATE` before calling the procedure and by appropriate interpretation of the output.

3. Procedure DROSE

This procedure draws a rose diagram using high quality graphics. The rose diagram is analogous to the histogram for linear data. An example, the vanishing direction of mallards from the Slimbridge wildfowl reserve (Mardia 1972) is shown in Figure 1. Each sector of the diagram represents a class interval of 20°.

The form of the procedure is

```
DROSE [UNITS=string; TITLE=text; WINDOW=scalar; SCREEN=string; \
  ZERODIR=scalar;  SENSE=string] DATA=variates; LIMITS=variates
```

Option `UNITS` has already been described. Options `TITLE`, `WINDOW` and `SCREEN` have the same form and function as for the `DHISTOGRAM` directive. Option `ZERODIR` supplies the angle of the zero direction from the positive x-axis in degrees, while option `SENSE`, which has settings clockwise and anticlockwise, determines the direction in which the angles are measured from the zero direction. In pure mathematics, angles are usually defined anticlockwise from the positive x-axis and this is the default setting. However, in many applications the angles will be measured clockwise from *North*, i.e., `ZERODIR=270`.

Parameter `DATA` has already been described. Parameter `LIMITS` defines the class limits and is directly analogous with the `LIMITS` option of `DHISTOGRAM`. The setting of `LIMITS` must be consistent with the `UNITS` option.

As implemented here, the radius of the sectors is proportional to the class frequency. However, it is generally thought to be easier to interpret the diagram if the area of the sector is proportional to the frequency; that is, the radius is equal to the square root of the frequency. It is intended to include an option for this equiareal rose diagram in future.

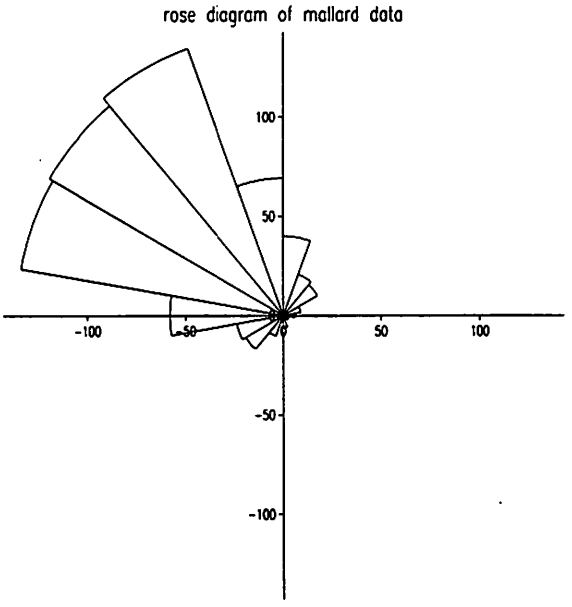


Figure 1: Rose diagram produced by DROSE procedure

4. Procedure CDESCRIBE

This procedure calculates simple summary statistics for circular data and is thus analogous to the library procedure DESCRIBE for linear data. The procedure has the form

```
CDESCRIBE [UNITS=string; SELECTION= string] DATA=variates
```

The parameter DATA and the option UNITS have already been described. Option SELECTION has settings cmean, cmedian, cvar, csd and crange. These are used to request the mean direction, median direction, circular variance, circular standard deviation and circular range, respectively.

By treating the data as vectors on a unit circle, the mean direction is defined as the direction of the resultant of these vectors. This is calculated using standard trigonometric formulae. The length of the resultant is used as a measure of the dispersion about the mean direction. If the data are closely clustered it will approach N, the number of data points, but if widely scattered it will approach 0. It should be noted however that a value of 0 does not necessarily imply maximal dispersion. The mean resultant length is obtained by dividing the resultant length by N. The circular variance is defined as (1- mean resultant length) and thus shares with the variance of linear data, the property of increasing with increasing dispersion. It also has a similar minimization property in that the variance is minimised about the mean direction. However, unlike the linear variance, the circular variance is a dimensionless quantity defined on the interval (0,1).

The circular variance is difficult to interpret. It can be transformed to the circular standard deviation which is defined on $(0,\infty)$. This is effectively a transformation to the linear scale and is analogous to the standard deviation on the line. It is expressed in the appropriate angular units defined by the UNITS option. The circular median is defined as a point P on the unit circle such that half the sample points lie on each side of the diameter PQ and the majority of the sample are nearer P than Q. The circular median is unique only for unimodal distributions. The procedure checks for uniqueness and prints a warning if non-uniqueness is detected. The

circular range is defined as the length of the smallest arc which encompasses all the sample observations. Again it is unique only for unimodal distributions and this is checked for by the procedure. It is intended to extend CDESCRIBE to include other descriptive statistics such as measures of skewness and kurtosis and the trigonometric moments.

5. Procedure CRUNS

This procedure carries out a simple non-parametric test for the equality of two distributions. It is based on cutting the circle at some arbitrary point and counting the number of runs on the resulting line. A small number of runs indicates separation of the samples while a large number indicates that the samples are mixed together. For total sample size greater than 40, a normal approximation is used to test the hypothesis. For values between 8 and 40, a look–up table (Mardia 1972) is used. The method used is printed in the output. If there are fewer than 8 samples, the test is not carried out and a warning is printed.

The form of the procedure is

```
CRUNS [UNITS=string; GROUPS=factor] DATA= variate
```

UNITS and DATA are as already described, while GROUPS is a factor defining the two samples. The data are ordered from the zero direction and the number of runs counted. The number of runs is invariant under rotation. Since the number of runs on a circle must be even, the (number of runs-1) is used if it is odd; that is, if the cut point is in the middle of a run.

6. Future Developments

It is hoped to develop this suite of procedures further by including a number of one– and two-sample parametric tests. This will also require the provision of a procedure to calculate the cumulative distribution functions for circular distribution, in particular the von Mises distribution. It is also hoped to produce some procedures for the analysis of spherical data.

References

Fisher N I (1993) *Statistical analysis of circular data* Cambridge University Press, Cambridge.
Mardia K V (1972) *Statistics of directional data* Academic Press, London.

The Procedure

```
PROCEDURE [RESTORE=dsave] 'DROSE'
"
A. J. Rook, Institute of Grassland and Environmental Research,
North Wyke, Okehampton, Devon EX20 2SB

Version 1.2  12/9/94

Procedure to draw a rose diagram for a circular frequency distribution.

References.
Mardia, K. V.  1972.  Statistics of Directional Data, Academic Press, London."

"
Declaration of options"
OPTION NAME= \
  'WINDOW',  "(I: scalar {1...8} \
             default 1) window number for high quality graph" \
  'SCREEN',  "(I: string {clear,keep} \
```

```
            default clear) whether to clear screen before plotting or not"\
  'UNITS', "(I: string {degrees,radians,hours,days} \
       default radians) unit of angular measurement to be used for input" \
  'ZERODIR', "(I: scalar {0...359} \
            default 0) angle in degrees of zero direction of data as \
            measured clockwise from positive y axis" \
  'SENSE',  "(I: string {clockwise,anticlockwise} \
          default clockwise)  whether to treat angles as measured in a \
          clockwise or anticlockwise direction from the zero direction" \
  'TITLE'; "(I: text default *) overall title for graph" \
  MODE=v,t,t,v,t,t; NVALUES=6(1); \
  VALUES= !(1...8),\
          !T(CLEAR,KEEP), \
          !T(DEGREES,RADIANS,HOURS,DAYS), \
          !(0...359), \
          !T(CLOCKWISE,ANTICLOCKWISE), \
          *; \
  DEFAULT= 1, 'CLEAR', 'RADIANS', 0, 'CLOCKWISE', *; \
  SET= 6(no); LIST=6(no)
"
Declaration of parameters"
PARAMETER NAME= \
  'DATA', "(I: variate) Contains directions (angles) expressed as degrees, \
          radians,hours of day (24 h clock),or days of year (sequential)" \
  'LIMITS'; "(I: variate  default *) Contains limits for groups of rose \
          diagram" \
  DEFAULT= *,*; SET= yes,no; DECLARED= yes,yes; \
  TYPE= !T(variate),!T(variate); PRESENT= yes,yes

"
Convert data and limits to radians if not entered as such.  Also check that
ranges of data and limits are compatible with setting of units option"
IF 'DEGREES'.EQS.UNITS
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Data entered as degrees include values outside 0-360 ***'] \
  SUM(DATA.LT.0.OR.DATA.GT.360).NE.0
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Limits entered as degrees include values outside 0-360 ***'] \
  SUM(LIMITS.LT.0.OR.LIMITS.GT.360).NE.0
  CALC data,limits=DATA,LIMITS*CONSTANTS('pi')/180
ELSIF 'HOURS'.EQS.UNITS
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Data entered as hours include values outside 0-24 ***'] \
  SUM(DATA.LT.0.OR.DATA.GT.24).NE.0
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Limits entered as hours include values outside 0-24 ***'] \
  SUM(LIMITS.LT.0.OR.LIMITS.GT.24).NE.0
CALC data,limits=DATA,LIMITS*15*CONSTANTS('pi')/180
ELSIF 'DAYS'.EQS.UNITS
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Data entered as days include values outside 0-365 ***'] \
  SUM(DATA.LT.0.OR.DATA.GT.365).NE.0
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Limits entered as days include values outside 0-365 ***'] \
  SUM(LIMITS.LT.0.OR.LIMITS.GT.360).NE.0
  CALC data,limits=DATA,LIMITS*360/365*CONSTANTS('pi')/180
ELSE
  CALC maxrad=2*CONSTANTS('pi')
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Data entered as radians includes values outside 0-2pi ***'] \
  SUM(DATA.LT.0.OR.DATA.GT.maxrad).NE.0
  EXIT [CONTROL=procedure; EXPLANATION= \
  '***  ERROR - Limits entered as radians include values outside 0-2pi ***'] \
  SUM(LIMITS.LT.0.OR.LIMITS.GT.maxrad).NE.0
  CALC data,limits=DATA,LIMITS
ENDIF
"
Calculate functions of number of limits for later use"
CALC nlim=NOBS(limits)
CALC nlimm1=nlim-1
CALC nlimp1=nlim+1
"
Use groups directive to form classes as defined by limits parameter"
GROUPS [LMETHOD=given] data; FACTOR=classes; LIMITS=limits; \
LEVELS=!(1...nlimp1)
"
```

```
Use tabulate directive to obtain number (frequency) in each class"
TABULATE [CLASS=classes] data; NOBS=tfreq1
COMBINE [OLD=tfreq1; NEW=tfreq] OLDDIM=classes; NEWDIM=classes; \
OLDPOS=!(1...nlimp1,1,nlimp1); NEWPOS=!(1...nlimp1,nlimp1,1)
"
Calculate x and y coordinates of arcs defining each class.  Include these
with 0's to provide continuous line for drawing rose."
CALC mnlimm1=-nlimm1
CALC npoints=(12*nlim)+1
VARIATE [NVAL=nlim; val=#nlim(0)] zero
VARIATE [NVAL=npoints] freq,cmin
CALC diff=CIRC(DIFF(limits);-1)
CALC diff$[nlim]=limits$[1]+2*CONSTANTS('pi')-limits$[nlim]
FOR i=1...9
  CALC inter[i]=limits+(diff/10)*i
ENDFOR
EQUATE [old=!((1,mnlimm1,-1,1,((mnlimm1,-1)2,1)10,mnlimm1,-1)#nlimm1,\
1,mnlimm1,-1,1,((mnlimm1,-1)2,1)10,mnlimm1,1)] \
!P(zero,tfreq); freq
EQUATE [OLD=!(((1,mnlimm1)11,mnlimm1,-2,1,(mnlimm1,-1)9,mnlimm1)#nlimm1,\
(1,mnlimm1)11,-1,1,mnlimm1,(mnlimm1,-1)9,1)] \
!P(zero,limits,inter[1...9]); cmin
"
Adjust angles so that zero direction (as given by zerodir option) is plotted
correctly"
CALC cmin=cmin+(ZERODIR*CONSTANTS('pi')/180)
"
Calculate x and y values for plotting according to whether angles are
measured clockwise or anticlockwise"
IF 'CLOCKWIS'.EQS.SENSE
  CALC y=freq*COS(cmin)
  CALC x=freq*SIN(cmin)
ELSIF 'ANTICLOC'.EQS.SENSE
  CALC y=freq*COS(cmin)
  CALC x=-freq*SIN(cmin)
ENDIF
"
Draw high quality graph"
CALC max=MAX(freq)
CALC mmax=-max
AXES WINDOW=WINDOW; YORIGIN=0; XORIGIN=0; \
STYLE=xy; PENAXES=2; YUP=max; YLO=mmax; XLO=mmax; XUP=max
PEN NUMB=1; LINE=1; METHOD=line; JOIN=given; SYMBOLS=0
PEN NUMB=2; LINE=1; COLOUR=3
DGRAPH [TITLE=TITLE; WINDOW=WINDOW; KEYWINDOW=*; SCREEN=#SCREEN] \
Y=y; X=x; PEN=1
ENDPROCEDURE
RETURN
```

Structure of a Genstat userfile

A D Todd
IACR Rothamsted
HARPENDEN
Hertfordshire AL5 2JQ, UK

1. Introduction

Genstat has facilities for storing data structures in binary files called backing-store files. Backing-store files fall into two categories: temporary files used by the program within a Genstat job, called *workfiles*, and permanent files created by users to transfer structures between Genstat jobs, called *userfiles*. Data structures stored in a userfile are arranged in sets called subfiles. Each subfile must have a unique name for easy access of data. The definition of some data structures will create links to other structures: for example, a factor with labels stored in a text vector. When Genstat creates a subfile, it will include both the structures requested by the user and all other structures linked to these (e.g., the labels of a factor, or the data structures pointed to by a pointer). Backing-store is the most efficient way of transferring data between Genstat jobs, since the complete definition of the structures is retained. Other directives can be used (READ and PRINT directives) but a user would have to go to the trouble of redefining data structures.

This article outlines the basic structure of a userfile in the simplest case, for those users who may wish to use programs other than Genstat to read or write userfiles. Note that Genstat data structures may be very complex, resulting in a complex structure for the userfile. However, here a simple example is considered, namely a userfile containing one subfile holding a few straightforward data structures.

2. Basic description of a userfile

A userfile is a self-contained file, named by the user when the file is opened using the OPEN directive, so that any run of Genstat is capable of accessing structures stored in it. To make this possible, each userfile and every subfile within it contains a catalogue. The catalogues are based on the catalogues for Genstat data structures and are only simple for the basic data structures. For this reason, we restrict discussion to userfiles containing subfiles whose names are not suffixed, and to the storing of three data structures types: scalar, variate and factor. In addition, we only discuss these structures when their definition does not depend on any other structures (e.g., no labels for the factor), and assume that the names of these data structures do not use suffixes.

Every userfile starts with some binary unformatted records (the *userfile catalogue*) that give a catalogue of the subfiles. The userfile catalogue contains information such as names of subfiles, type of subfile (ordinary or procedure) and number of records in the userfile catalogue and in each subfile. After the userfile catalogue there are records for every subfile in the userfile.

Each subfile starts with some records giving a catalogue of structures stored. This catalogue contains information such as the names of the structures stored, type of structure (scalar, text, etc), whether the structure has values, number of records used to store each structure, plus information about pointers and how structures depend on each other. Then each structure is stored, normally in two records: one for its attributes (number of values, etc) and the other for its values.

3. Format of records on backing-store

Each record is a binary record. A record starts with an integer (4 bytes; all word sizes given are for VAX computer range) giving the number of items in the record, followed by the items. There are five modes of item: Long Real (8 bytes); Real (4 bytes); Integer (4 bytes); Character (1 byte characters) or Word (8 bytes characters). So a record containing 6 double reals has 52 bytes: 4 for the initial integer containing the value 6 and 48 bytes

for the 6 long reals. There is also a maximum record size (including the initial integer) of MAXBPR bytes (usually 4096). The exact value can be found in common G5KICH and can be displayed for a version of Genstat using the command

```
DUMP [COMMON=ICH]
```

For mode character, the number of characters in a record can not exceed NCHBFF (usually 200). When backing-store is required to store more data than a record can hold, several records are written and all but the last record will be of maximum size.

The Fortran code for reading and writing data of mode integer and character of length LK (mode real, long real and word are similar to mode integer) is illustrated below.

```
CHARACTER*(200) STR
INTEGER IDATA(1000)
READ (DSN) LK,(IDATA(I),I=1,LK)
READ (DSN) LK,STR(1:LK)
WRITE (DSN) LK,(IDATA(I),I=1,LK)
WRITE (DSN) LK,STR(1:LK)
```

DSN is the Fortran channel number for a binary file (not to be confused with the Genstat channel number as given in the OPEN directive).

For Genstat development purposes, there is a special directive DBUG (not to be confused with DEBUG) that prints monitoring information. Output from this directive may be large, and unreadable to the uninitiated. However, if you wish to see the records read or written by backing-store you can do so by using the command

```
DBUG [BSIO=2]
```

before backing-store commands. For the STORE directive, the output will be most easily understood when storing to a new userfile. For the CATALOGUE directive, only records in the catalogues accessed will be displayed.

4. Example showing contents of a userfile

In this section, the contents of the backing-store file generated by the following program (run on VAX, release 3.2) are described, record by record.

```
VARIATE v; VALUES=!(1...9); DECIMALS=0
SCALAR s; VALUE=3; DECIMALS=2
FACTOR [LEVEL=3] f; VALUES=!(3(1...3))
STORE [SUBFILE=subfile; CHANNEL=1] STOREDIDENTIFIER=v,s,f
```

In the following, note that

1) data in word mode always has 8 characters and leading spaces are not displayed
2) * (missing value) takes a value that varies according to data mode: for integer, * = -2147483647; for real, * = -1.0E37 (on VAX). For other machines you can see the values by displaying common G5KUSY using the DUMP directive, the values are stored as IMV (integer) and RMV (real).

Within userfiles, all catalogues start with a record containing 20 integers. Below, record numbers are relative to the contents of the whole userfile and item numbers refer to items within records.

4.1 Userfile catalogue

Record 1: Initial record in userfile catalogue of mode integer, length 20.

item	value	meaning
1	1	number of subfiles in userfile
2	1	number of subfiles in userfile
3	1	number of subfiles in userfile
4	0	always 0 for userfile catalogue

5	0	length of userfile password (not described)
6	12	number of records in the userfile catalogue
7	532	Genstat version number; value for current release given by MARKNO in common G5KJRT (use DUMP directive)
8	0	number of pointer type structures if suffixed subfile name used (not described)
9	0	always 0 for userfile catalogue
10	8	backing-store type (value for release 3.2). This value only changes when the definition of structures changes between releases. If different to the current value in program, Genstat will sort out the problems if it can, or fail the program (rarely)
11	0	number of lines in password
12	2	number of records of mode word in this catalogue
13	0	number of procedures in userfile (used for procedure libraries)
14-19	0	currently not used, always 0
20	702	value for backing-store file, always the same (701 if RECORD/RESUME file)

There then follows the remaining 11 records of the userfile catalogue, all of length 1 (since only one subfile is stored here):

record	mode	value	meaning
2	integer	*	number of procedures in each subfile (* means none)
3	integer	1	type of subfile (2 = procedure subfile)
4	integer	15	number of records in subfile
5	word	subfile	name of subfile
6	word	subfile	name of subfile
7	integer	-1	structure number of subfile, always negative
8	integer	1	position of subfile name in record 5
9	integer	*	feature not described in this document
10	integer	10	maximum length of record in subfile of mode real, long real or integer in long real units (minimum value 10 in Release 3.2)
11	integer	0	maximum length of record in subfile of mode character
12	integer	3	maximum length of record in subfile of mode word (number of named structures stored in subfile)

End of userfile catalogue.

4.2 Subfile catalogue

Record number 13 is the initial record of the subfile catalogue, of mode integer, length 20.

item	value	meaning
1	3	number of structures stored in subfile (including unnamed if present)
2	3	number of named identifiers (not including procedures)
3	3	number of named identifiers (would include procedures if stored)
4	0	number of procedures in subfile
5	0	always 0 (only used by userfile catalogue)
6	9	number of records in subfile catalogue
7	532	Genstat version number, description same as userfile catalogue
8	0	number of pointer type structures
9	0	this item is used to link structures together, a feature not described in this document
10	8	backing-store type (value for release 3.2)
11	0	not used by subfile catalogue
12	2	number of records of mode word in this catalogue
13	0	not used by subfile catalogue
14-19	0	currently not used, always 0
20	702	value always the same

There then follow the remaining 8 records of the subfile catalogue, all of length 3, ie. one value per structure in each record.

record	mode	values	meaning
14	integer	4,1,2	type of stored structures (1 = scalar; 2 = factor; 4 = variate)
15	integer	2,2,2	number of records used to store each structure (note: attributes and values are stored as separate records)
16	word	v,s,f	names of stored structures
17	word	v,s,f	in this case same as record 16
18	integer	-1,-2,-3	structure numbers of the stored structures, always negative
19	integer	1,2,3	position of named structures in record 18 in record 16
20	integer	*,*,*	always * (no pointers stored)
21	integer	0,0,0	number of structures that the structure being stored directly depend on (does not include pointer values).

End of subfile catalogue

4.3 Attributes and values of data structures stored within the subfile

Record 22, mode integer length 11, holds the attributes of the first structure stored, variate v:

item	value	meaning
1	10846	this value indicates if values are present. * means no values, any positive integer indicates values present
2	9	number of storage units
3	9	number of values (usually the same as item 2 above, main exception is type TEXT)
4	1	mode of structure (1=long real)
5	0	number of missing values
6	*	unit labels vector number (not discussed here); * = not present
7	*	associated heading vector number (not discussed here); * = not present
8	0	default decimal places to print ; * = unset
9	*	structure number for minimum and maximum of a structure; * = not present
10	*	currently always set to *
11	*	structure number for restriction set; * = unset

Record 23 (mode long real, length 9) holds the values for variate v: values 1.0, 2.0, 3.0, 4.0, 5.0, 6.0, 7.0, 8.0, 9.0.

Record 24 (mode integer, length 10), holds the attributes of the scalar s and takes values: 10912, 1, 1, 1, 0, *, *, 2, *, *. The definition of these attributes is the same as for the variate attributes 1-10 given above.

Record 25 (mode long real, length 1), holds the value of scalar s, value = 3.0.

Record 26 (mode integer, length 13), holds the attributes of factor f, and takes the values 11020, 9, 9, 3, 0, *, *, *, *, *, *, *, 3, *. Attributes 1 to 8, 10 and 11 are similar to variates, except the structure is of mode integer (attribute 4). The remaining attributes (items) are defined as follows:

item	meaning
9	Structure number of variate holding actual values of the factor level; * = not present
12	The default number of character to print level names; * = unset
13	Number of levels
14	The number of the level names text vector (if any) which holds the list of level names; * = unset

Record 27 (mode integer, length 9), holds the values of factor f: 1, 1, 1, 2, 2, 2, 3, 3, 3 .
End of attributes and values in subfile and end of userfile.