Citation for published version

DOI

Link to record in KAR
http://kar.kent.ac.uk/21161/

Document Version
UNSPECIFIED

Copyright & reuse
Content in the Kent Academic Repository is made available for research purposes. Unless otherwise stated all content is protected by copyright and in the absence of an open licence (eg Creative Commons), permissions for further reuse of content should be sought from the publisher, author or other copyright holder.

Versions of research
The version in the Kent Academic Repository may differ from the final published version. Users are advised to check http://kar.kent.ac.uk for the status of the paper. Users should always cite the published version of record.

Enquiries
For any further enquiries regarding the licence status of this document, please contact:
researchsupport@kent.ac.uk

If you believe this document infringes copyright then please contact the KAR admin team with the take-down information provided at http://kar.kent.ac.uk/contact.html
A Remark on Algorithm AS30 – Half-Normal Plotting
By Tim Hopkins and David Morse
Computing Laboratory, University of Kent,
Canterbury, Kent CT2 7NF, UK.

Keywords: Half-normal plot; Normality of residuals; Factorial experiments; Analysis of variance.

LANGUAGE
Fortran 77.

DESCRIPTION AND PURPOSE
Given a set of root mean squares obtained from the analysis of a factorial experiment, one method of testing the significance, and/or examining for evidence of certain types of error, is to draw a half-normal plot of these values (Daniel, 1959). The subroutine takes such a set of root mean squares and plots them on a half-normal probability scale via the line printer or other device. For this sort of application the accuracy of the average line printer is more than adequate.

This routine is a Fortran 77 version of that appearing in Griffiths and Hill (1985) (originally Sparks (1970)). It uses character variables rather than storing character information in integers which was the only way of implementing such an algorithm in standard Fortran 66. Where possible data statements have been replaced by parameter statements. The code has been tidied up considerably and all output confined to a single routine to facilitate user tailoring of the final plot. A number of minor errors in a declared array length and format statements have been removed.

The only minor change to the plotting algorithm is in the range of values assigned to each line of the plot. Specifically the ith line of the plot, \( 0 \leq i \leq IDEPTH \), contains values in the range \((y - 0.5 \times YSTEP, y + 0.5 \times YSTEP)\) where \( y = i \times IDEPTH \), \( YSTEP = obsmax/IDEPTH \), \( obsmax \) is the maximum value to be plotted and \( IDEPTH \) defines the number of lines to be used in the plot. The original algorithm used the range \([y - 0.0001 \times YSTEP, y + 0.999 \times YSTEP]\) which does not give such an even distribution of points.
STRUCTURE

SUBROUTINE HNPL T(OBS, N, IWIDTH, IDEPTH, OUTPUT, IFAULT)

Formal parameters

OBS Real array (N) input: the values to be plotted, sorted such that
OBS(1) ≤ OBS(2) ≤ ...

OBS(N). Values

must not be negative,
or greater than 999999.

There must be at least
two distinct points.

N Integer input: the number of values to
be plotted. Must be
between 2 and 1250
inclusive.

IWIDTH Integer input: the width of the plot. Must be between 40
and 200 inclusive. See
below for further
details.

ID EPTH Integer input: the number of lines of
the plot. Must not be
less than 15. Including
a horizontal axis and
scaling, the total depth
will be IDEPTH + 3.

OUTPUT External input: user provided function
controlling how the plot
is produced. See below
for further details.
IF A U L T  Integer output:  1 for illegal value of \( N \), \( I WIDTH \) or \( I DEPTH \);
   2 if \( O B S \) array not correctly sorted or there are not at least two distinct points;
   3 for illegal \( O B S \) value;
   4 for illegal argument to \( P P N D \) (this fault is probably impossible);
   0 otherwise.

Auxiliary routine

STRUCTURE

SUBROUTINE OUTPUT(IDEPTH, IWIDTH, LINE, IOUT, YSTEP, RESET)

Formal parameters

\( I DEPTH \)  Integer  input:  as routine \( H N P L O T \).
\( I WIDTH \)  Integer  input:  as routine \( H N P L O T \).
\( L I N E \)  Integer  input:  0 signifies the \( x \)-axis line is in \( I O U T \);
  \(-1 \ I O U T \) contains the \( x \)-axis label markers;
  \(-2 \ I O U T \) contains the \( x \)-axis labels;
  \([1, I DEPTH]\) the array \( I O U T \) contains the line of the plot at
  \( y = L I N E \ast Y S T E P \)
  where \( Y S T E P \) is defined above.
**IOUT** Character array (IWIDTH) input: contains the character DTCCHAR (see Adjustable constants section below) in elements where data values appear, blanks otherwise.

**YSTEP** Real input: the distance between successive lines on the plot in data units, may be used to label the y-axis.

**RESET** Logical input: if .FALSE. then no data values appear on the current line; may be used to reduce the amount of output performed by HNPLOT.

**Auxiliary algorithm**

*REAL FUNCTION PPND (P, IFAULT) – Algorithm AS 111 (Beasley and Springer, 1977).*

**Adjustable constants**
The following values are defined in PARAMETER statements:

In routine HNPLOT

**XMCHAR** Character constant: the character used as the label marker on the x-axis.

**DTCHAR** Character constant: the character used for plotting the data points.

In routine OUTPUT

**XAXCHR** Character constant: the character used to print the x-axis of the plot.
**YAXCHR**  Character constant: the character used to print the y-axis of the plot.

The values set are those used in Griffiths and Hill (1985).

**Width of plot**
Including a vertical axis and scale the total width will be $IWIDTH + 12$ if the output device is one defined by Fortran as a ‘printing’ device which takes the first character of the line as a carriage control character, or $IWIDTH + 13$ otherwise.

**RESTRICTIONS**
None.

**PRECISION**
Real precision is perfectly adequate on any computer, so the usual *Applied Statistics* devices to enable easy translation to double precision are not incorporated. If the observations are held in a double precision array, they should be copied to a single precision array to be used as the actual argument corresponding to *OBS*.

**REFERENCES**