

NAG Library Routine Document

G01ARF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of *bold italicised* terms and other implementation-dependent details.

1 Purpose

G01ARF produces a stem and leaf display for a single sample of observations.

2 Specification

```
SUBROUTINE G01ARF (RANGE, PRT, N, Y, NSTEPX, NSTEPY, UNIT, PLOT, LDPLOT, &
                  LINES, SORTY, IWORK, IFAIL)

INTEGER          N, NSTEPX, NSTEPY, LDPLOT, LINES, IWORK(N), IFAIL
REAL (KIND=nag_wp) Y(N), UNIT, SORTY(N)
CHARACTER(1)    RANGE, PRT, PLOT(LDPLOT,NSTEPX)
```

3 Description

G01ARF produces a stem and leaf display for a single sample of n observations. The stem and leaf display shows data values separated into the form of a 'stem' and a 'leaf'. For example, a value of 473 could be represented as 47 3 where the stem is 47 and the leaf is 3. The data is scaled using a value known as the 'leaf digit unit'. In the above example the leaf digit unit would be 1.0.

The following example illustrates a stem and leaf display.

For the 10 observations:

1.8 2.3 2.1 1.9 2.1 2.4 2.0 2.0 1.9 2.1

the stem and leaf display is:

```
1  1  8
3  1  99
5  2  00
5  2  111
2  2
2  2  3
1  2  4
```

where the leaf digit unit is 0.1 so that 1 8 represents 1.8 (i.e., 18×0.1). The leaf digit unit distinguishes between the numbers 18.0, 1.8, 0.18, etc. which may otherwise all be represented by 1 8.

Included in the above display is an initial column specifying the cumulative count of values, up to and including that particular line, from either the top or bottom of the display, whichever is smaller. An exception to this is when the line on which the median lies is reached, in which case the actual count of values on that line is displayed, rather than a cumulative count, and this is highlighted by enclosing the count in parentheses. In this case the median is 2.05 and thus falls between the two lines at which the cumulative count has reached $n/2$ where n is the number of observations.

Some of the other features of the stem and leaf display are illustrated by the following two examples.

For the 30 observations:

```
−19.0  −3.0  −1.0  0.0  1.0  2.0  2.0  3.0  3.0  3.0
   4.0   4.0   4.0  4.0  4.0  5.0  5.0  5.0  5.0  6.0
   6.0   6.0   7.0  7.0  8.0 10.0 11.0 11.0 13.0 31.0
```

the stem and leaf display may be:

```

1  1.  9
1  1*
1  -0.
3  -0*  13
15 +0*  012233344444
15 +0.  555566667788
5  1*  011
2  1.  3
1  2
1  2.
1  3  1

```

In the above display all the data are plotted and the leaf digit unit is 1.0. Also in this display different leaves, that is different digits, may be plotted on a particular line. In this case we have 5 possible digits per line, that is 2 lines per stem, and these are represented as follows:

- * indicates that the line may contain the digits 0 to 4;
- . indicates that the line may contain the digits 5 to 9.

Alternatively the stem and leaf display may look like:

```

LO  -19

2  -0*  3
3  +0T  1
5  +0*  01
10 +0T  22333
( 9) +0F  444445555
11 +0*  66677
6  +0T  8
5  1*  011
2  1T  3

HI  31

```

Again the leaf digit unit is 1.0 but in this display just the data between the fences, which are the hinges $\pm 1\frac{1}{2} \times$ the inter-hinge range, are plotted. Any data points that fall outside the fences are presented separately in the display under the headings LO for those points below the lower fence and HI for those points above the upper fence.

Again in this display different leaves, that is different digits, may be plotted on a particular line. However in this case we have 2 possible digits per line, that is 5 lines per stem, and these are represented as follows

- * indicates that the line may contain the digits 0 or 1;
- T indicates that the line may contain the digits 2 or 3;
- F indicates that the line may contain the digits 4 or 5;
- S indicates that the line may contain the digits 6 or 7;
- . indicates that the line may contain the digits 8 or 9.

A display may also allow 10 different digits (0 to 9) per line, that is 1 line per stem, or just 1 digit per line, that is 10 lines per stem, as in the first of the three examples above.

Note that the median here is 4.5. This falls between two lines in the first display but is highlighted on the second display since it lies on a particular line.

Finally if there are positive and negative numbers on the display these are highlighted by a + or - sign where the distinction is required, that is near the zero-point.

If there are too many leaves to fit in the plot width allowed, G01ARF plots as many leaves as possible and places an asterisk to the right to indicate that some leaves are not displayed. If this occurs and you wish to be able to plot all the leaves then the width of the plot may be adjusted.

Options also allow the leaf unit and the height of the display to be specified by you or calculated by G01ARF. These parameters may be used to control the type of the display you wish to obtain. Fixing the unit and changing the height of the display may alter the number of lines used per stem, that is the number of different digits per line. G01ARF will choose a display for the fixed unit that attempts to make as much use of the available height as possible, thus increasing the height may allow for more lines per stem whereas decreasing the height may force the display to use fewer lines per stem. Similarly you may wish to fix the height and vary the leaf digit unit used on the display. See Section 8 for further details.

The display is returned in a character array with the option of printing the display.

4 References

Erickson B H and Nosanchuk T A (1985) *Understanding Data* Open University Press, Milton Keynes

Tukey J W (1977) *Exploratory Data Analysis* Addison–Wesley

Velleman P F and Hoaglin D C (1981) *Applications, Basics, and Computing of Exploratory Data Analysis* Duxbury Press, Boston, MA

5 Parameters

- 1: RANGE – CHARACTER(1) *Input*
On entry: indicates whether you wish to scale the plot to the extremes of the data or to the fences.
 RANGE = 'E'
 The display is a plot to the extremes, that is a plot of all the data.
 RANGE = 'F'
 The display is a plot of the data between the fences.
Constraint: RANGE must be one of 'E' or 'F'.

- 2: PRT – CHARACTER(1) *Input*
On entry: indicates whether the stem and leaf display is to be output to an external file.
 PRT = 'N'
 The display is not output to an external file.
 PRT = 'P'
 The display is output to the current advisory message unit as defined by X04ABF. Only the first 132 characters of each line are actually printed.
Constraint: PRT must be one of 'P' or 'N'.

- 3: N – INTEGER *Input*
On entry: n , the number of observations.
Constraint: $N \geq 2$.

- 4: Y(N) – REAL (KIND=nag_wp) array *Input*
On entry: the n observations.

- 5: NSTEPX – INTEGER *Input*
On entry: the number of character positions to be plotted horizontally.
Constraint: NSTEPX ≥ 35 .

- 6: NSTEPY – INTEGER *Input*
On entry: the maximum number of character positions to be plotted vertically.

If $NSTEPY \leq 0$ a suitable value will be used by G01ARF for the number of character positions to be plotted vertically. This will clearly be less than or equal to the value of LDPLOT.

Constraint: $NSTEPY \leq 0$ or $NSTEPY \geq 5$.

- 7: UNIT – REAL (KIND=nag_wp) *Input/Output*
On entry: indicates the leaf digit unit to be used.
 If $UNIT > 0.0$ and is not a power of ten, it will be converted to the nearest power of ten below the input value for unit.
 If $UNIT \leq 0.0$, the optimum unit will be used. This is based on the range of the data to be plotted and the number of lines available for the display.
On exit: contains the actual unit used in the stem and leaf display.
- 8: PLOT(LDPLOT,NSTEPX) – CHARACTER(1) array *Output*
On exit: the stem and leaf display.
- 9: LDPLOT – INTEGER *Input*
On entry: the first dimension of the array PLOT as declared in the (sub)program from which G01ARF is called.
Constraint: $LDPLOT \geq \max(5, NSTEPY)$.
- 10: LINES – INTEGER *Output*
On exit: the actual number of lines needed for the display.
- 11: SORTY(N) – REAL (KIND=nag_wp) array *Output*
On exit: the observations sorted into ascending order.
- 12: IWORK(N) – INTEGER array *Workspace*
- 13: IFAIL – INTEGER *Input/Output*
On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction for details.
 For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this parameter, the recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**
On exit: IFAIL = 0 unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL = 0 or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL = 1

On entry, $N < 2$,
 or $NSTEPX < 35$,
 or $0 < NSTEPY < 5$,
 or $LDPLOT < 5$,
 or $LDPLOT < NSTEPY$.

IFAIL = 2

On entry, PRT \neq 'P' or 'N',
or RANGE \neq 'E' or 'F'.

IFAIL = 3

The number of lines needed to produce the display exceeds the maximum number of lines allowed.
You may wish to increase NSTEPY.

IFAIL = 4

One of the observations is too large and causes a value to exceed the maximum integer allowed.

7 Accuracy

Accuracy is limited by the number of significant figures that may be represented on the display which will depend on the data, the number of lines available and the unit used.

8 Further Comments

G01ARF uses integer representations of the data. If very large data values are being used they should be scaled before using this routine. The largest integer can be found by calling X02BBF.

If an asterisk is plotted at the end of a line to indicate that some leaves are not displayed you should increase NSTEPX if they wish to be able to print the rest of the leaves on that line.

Note that if you request G01ARF to print the plot only the first 132 characters of each line are printed. The full plot is stored in the array PLOT so you do have the option of printing a plot which has more than 132 characters on a line.

When the leaf digit unit is set, the number of lines per stem is decided as follows:

Let r be the range of the data to be plotted:

r = largest observation – smallest observation: if all the data to both extremes are to be plotted (that is if RANGE = 'E'),

r = upper fence – lower fence: if only the data between the fences are to be plotted (that is if RANGE = 'F').

Let l be the number of lines available for the plot:

l = NSTEPY – 4 if NSTEPY > 0,

l = LDPLLOT – 4 if NSTEPY \leq 0.

The 4 lines are subtracted to allow space for the display headings. If only the data between the fences are to be plotted then l must be further reduced to allow space to present those values outside the fences. This will involve a minimum of another 4 lines.

Let $e = \frac{(r/\text{UNIT}) + 1}{l}$,

then the number of lines per stem is:

- 1 if $5 < e \leq 10$, that is digits per line is 10,
- 2 if $2 < e \leq 5$, that is digits per line is 5,
- 5 if $1 < e \leq 2$, that is digits per line is 2,
- 10 if $0 < e \leq 1$, that is digits per line is 1.

The time taken by the routine increases with n .

9 Example

A program to produce two stem and leaf displays for a sample of 30 observations. The first illustrates a plot produced automatically by G01ARF and the second shows how to print the display under your control.

9.1 Program Text

```

Program g01arf

!      G01ARF Example Program Text

!      Mark 24 Release. NAG Copyright 2012.

!      .. Use Statements ..
Use nag_library, Only: g01arf, nag_wp, x04abf
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: iset = 1, nin = 5, nout = 6
!      .. Local Scalars ..
Real (Kind=nag_wp)         :: unit, unit1, unit2
Integer                    :: i, ifail, ldplot, lines, n, nstepx, &
                          nstepy, outchn
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: sorty(:), y(:)
Integer, Allocatable          :: iwork(:)
Character (1), Allocatable    :: plot(:, :)
!      .. Intrinsic Procedures ..
Intrinsic                    :: max
!      .. Executable Statements ..
Write (nout,*) 'G01ARF Example Program Results'
Write (nout,*)
Flush (nout)

!      Skip heading in data file
Read (nin,*)

!      Read in the problem size
Read (nin,*) n, unit, nstepx, nstepy

!      Make sure there is more than the minimum number of lines available
!      for the plot
ldplot = max(100,nstepy)

Allocate (y(n),iwork(n),sorty(n),plot(ldplot,nstepx))

!      Read in data
Read (nin,*) y(1:n)

!      Set advisory channel
outchn = nout
Call x04abf(iset,outchn)

!      Produce and display the plot
unit1 = unit
ifail = 0
Call g01arf('Fences','Print',n,y,nstepx,nstepy,unit1,plot,ldplot,lines, &
           sorty,iwork,ifail)

Write (nout,*)
Flush (nout)

!      Produce the plot, without printing it
unit2 = unit
ifail = 0
Call g01arf('Extremes','Noprint',n,y,nstepx,nstepy,unit2,plot,ldplot, &
           lines,sorty,iwork,ifail)

```

```

!      Display the plot
      Do i = 1, lines
        Write (nout,99999) plot(i,1:nstepx)
      End Do

99999 Format (1X,132A)
      End Program g01arfe

```

9.2 Program Data

```

G01ARF Example Program Data
30 0.0 72 0
 31.0 1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 -9.0
  1.0 2.0 3.0 4.0 5.0 6.0 7.0 8.0 2.0  3.0
  4.0 5.0 6.0 7.0 3.0 4.0 5.0 6.0 4.0  5.0

```

9.3 Program Results

G01ARF Example Program Results

```

Stem-and-leaf display
Leaf digit unit = 1.0
1 2 represents 12.

```

```

      LO  -9

      3   0  11
      6   0  222
     10   0  3333
     15   0  44444
     15   0  55555
     10   0  6666
      6   0  777
      3   0  88

```

```

      HI  31

```

```

Stem-and-leaf display
Leaf digit unit = 1.0
1 2 represents 12.

```

```

      1   -0. 9
      1   -0*
     15  +0* 112223333444444
     15  +0. 55555666677788
      1   1*
      1   1.
      1   2*
      1   2.
      1   3* 1

```
