NAG Library Routine Document

G01AEF

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

G01AEF constructs a frequency distribution of a variable, according to either user-supplied, or routine-calculated class boundary values.

2 Specification

```
SUBROUTINE G01AEF (N, K, X, ICLASS, CB, IFREQ, XMIN, XMAX, IFAIL)

INTEGER             N, K, ICLASS, IFREQ(K), IFAIL
REAL (KIND=nag_wp) X(N), CB(K), XMIN, XMAX
```

3 Description

The data consists of a sample of n observations of a continuous variable, denoted by x_i, for $i = 1, 2, \ldots, n$. Let $a = \min(x_1, \ldots, x_n)$ and $b = \max(x_1, \ldots, x_n)$.

G01AEF constructs a frequency distribution with $k(> 1)$ classes denoted by f_i, for $i = 1, 2, \ldots, k$.

The boundary values may be either user-supplied, or routine-calculated, and are denoted by y_j, for $j = 1, 2, \ldots, k - 1$.

If the boundary values of the classes are to be routine-calculated, then they are determined in one of the following ways:

(a) if $k > 2$, the range of x values is divided into $k - 2$ intervals of equal length, and two extreme intervals, defined by the class boundary values $y_1, y_2, \ldots, y_{k-1}$;

(b) if $k = 2$, $y_1 = \frac{1}{2}(a + b)$.

However formed, the values $y_1, \ldots, y_{k-1}$ are assumed to be in ascending order. The class frequencies are formed with

$$f_1 = \text{ the number of } x \text{ values in the interval } (-\infty, y_1)$$

$$f_i = \text{ the number of } x \text{ values in the interval } [y_{i-1}, y_i), \quad i = 2, \ldots, k - 1$$

$$f_k = \text{ the number of } x \text{ values in the interval } [y_{k-1}, \infty),$$

where [means inclusive, and) means exclusive. If the class boundary values are routine-calculated and $k > 2$, then $f_1 = f_k = 0$, and y_1 and y_{k-1} are chosen so that $y_1 < a$ and $y_{k-1} > b$.

If a frequency distribution is required for a discrete variable, then it is suggested that you supply the class boundary values; routine-calculated boundary values may be slightly imprecise (due to the adjustment of y_1 and y_{k-1} outlined above) and cause values very close to a class boundary to be assigned to the wrong class.

4 References

None.

5 Arguments

1: N – INTEGER *Input*

On entry: n, the number of observations.

Constraint: $\mathrm{N} \geq 1$.

2: K – INTEGER *Input*

On entry: k, the number of classes desired in the frequency distribution. Whether or not class boundary values are user-supplied, K must include the two extreme classes which stretch to $\pm\infty$.

Constraint: $\mathrm{K} \geq 2$.

3: X(N) – REAL (KIND=nag_wp) array *Input*

On entry: the sample of observations of the variable for which the frequency distribution is required, x_i, for $i = 1, 2, \ldots, n$. The values may be in any order.

4: ICLASS – INTEGER *Input*

On entry: indicates whether class boundary values are to be calculated within G01AEF, or are supplied by you.

If $\mathrm{ICLASS} = 0$, then the class boundary values are to be calculated within the routine.

If $\mathrm{ICLASS} = 1$, they are user-supplied.

Constraint: $\mathrm{ICLASS} = 0$ or 1.

5: CB(K) – REAL (KIND=nag_wp) array *Input/Output*

On entry: if $\mathrm{ICLASS} = 0$, then the elements of CB need not be assigned values, as G01AEF calculates $k - 1$ class boundary values.

If $\mathrm{ICLASS} = 1$, the first $k - 1$ elements of CB must contain the class boundary values you supplied, in ascending order.

In both cases, the element $\mathrm{CB}(k)$ need not be assigned, as it is not used in the routine.

On exit: the first $k - 1$ elements of CB contain the class boundary values in ascending order.

Constraint: if $\mathrm{ICLASS} = 1$, $\mathrm{CB}(i) < \mathrm{CB}(i+1)$, for $i = 1, 2, \ldots, k - 2$.

6: IFREQ(K) – INTEGER array *Output*

On exit: the elements of IFREQ contain the frequencies in each class, f_i, for $i = 1, 2, \ldots, k$. In particular $\mathrm{IFREQ}(1)$ contains the frequency of the class up to $\mathrm{CB}(1)$, f_1, and $\mathrm{IFREQ}(k)$ contains the frequency of the class greater than $\mathrm{CB}(k-1)$, f_k.

7: XMIN – REAL (KIND=nag_wp) *Output*

On exit: the smallest value in the sample, a.

8: XMAX – REAL (KIND=nag_wp) *Output*

On exit: the largest value in the sample, b.

9: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation 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 argument, 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, $K < 2$.

IFAIL $= 2$

On entry, $N < 1$.

IFAIL $= 3$

On entry, the user-supplied class boundary values are not in ascending order.

IFAIL $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

Your licence key may have expired or may not have been installed correctly.

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

The method used is believed to be stable.

8 Parallelism and Performance

G01AEF is not threaded in any implementation.

9 Further Comments

The time taken by G01AEF increases with K and N. It also depends on the distribution of the sample observations.

10 Example

This example summarises a number of datasets. For each dataset the sample observations and optionally class boundary values are read. G01AEF is then called and the frequency distribution and largest and smallest observations printed.

10.1 Program Text

```
    Program g01aefe

!     G01AEF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g01aef, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: xmax, xmin
      Integer                          :: iclass, ifail, j, k, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: cb(:), x(:)
      Integer, Allocatable             :: ifreq(:)
!     .. Executable Statements ..
      Write (nout,*) 'G01AEF Example Program Results'
      Write (nout,*)

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

!     Read in the problem size
      Read (nin,*) n, iclass, k

      Allocate (x(n),cb(k),ifreq(k))

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

      Write (nout,99997) 'Number of cases', n
      Write (nout,99997) 'Number of classes', k

!     Get the class boundaries
      If (iclass/=1) Then
        Write (nout,*) 'Routine-supplied class boundaries'
      Else
        Read (nin,*) cb(1:(k-1))
        Write (nout,*) 'User-supplied class boundaries'
      End If
      Write (nout,*)

!     Construct the frequency table
      ifail = 0
      Call g01aef(n,k,x,iclass,cb,ifreq,xmin,xmax,ifail)

!     Display results
      Write (nout,*) '*** Frequency  distribution ***'
      Write (nout,*)
      Write (nout,*) '       Class            Frequency'
      Write (nout,*)
      Write (nout,99999) '   Up to    ', cb(1), ifreq(1)
      k = k - 1
      If (k>1) Then
        Write (nout,99998)(cb(j-1),' to ',cb(j),ifreq(j),j=2,k)
      End If
      Write (nout,99996) cb(k), '    and over  ', ifreq(k+1)
      Write (nout,*)
      Write (nout,99995) 'Total frequency = ', n
      Write (nout,99994) 'Minimum = ', xmin
      Write (nout,99994) 'Maximum = ', xmax

99999 Format (1X,A,F8.2,I11)
99998 Format (1X,F8.2,A,F8.2,I11)
```

```
99997 Format (1X,A,I4)
99996 Format (1X,F8.2,A,I9)
99995 Format (1X,A,I6)
99994 Format (1X,A,F9.2)
    End Program g01aefe
```

10.2 Program Data

```
G01AEF Example Program Data
   70    0        7
22.3   21.6   22.6   22.4   22.4   22.4   22.1   21.9   23.1   23.4
23.4   22.6   22.5   22.5   22.1   22.6   22.3   22.4   21.8   22.3
22.1   23.6   20.8   22.2   23.1   21.1   21.7   21.4   21.6   22.5
21.2   22.6   22.2   22.2   21.4   21.7   23.2   23.1   22.3   22.3
21.1   21.4   21.5   21.8   22.8   21.4   20.7   21.6   23.2   23.6
22.7   21.7   23.0   21.9   22.6   22.1   22.2   23.4   21.5   23.0
22.8   21.4   23.2   21.8   21.2   22.0   22.4   22.8   23.2   23.6
```

10.3 Program Results

```
 G01AEF Example Program Results

 Number of cases  70
 Number of classes   7
 Routine-supplied class boundaries

 *** Frequency  distribution ***

        Class            Frequency

    Up to       20.70          0
    20.70 to    21.28          6
    21.28 to    21.86         16
    21.86 to    22.44         21
    22.44 to    23.02         14
    23.02 to    23.60         13
    23.60    and over          0

 Total frequency =     70
 Minimum =     20.70
 Maximum =     23.60
```