

# NAG Fortran 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, CINT, IFREQ, XMIN, XMAX, IFAIL)
INTEGER          N, K, ICLASS, IFREQ(K), IFAIL
real           X(N), CINT(K), XMIN, XMAX
```

### 3 Description

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

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

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

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  $y_1, y_2, \dots, y_{k-1}$ ;
- (b) if  $k = 2$ ,  $y_1 = \frac{1}{2}(a + b)$ .

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

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

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

$f_k =$  the number of  $x$  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 the user supplies 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 Parameters

- 1: N – INTEGER *Input*  
*On entry:* the number of observations,  $n$ .  
*Constraint:*  $N \geq 1$ .
- 2: K – INTEGER *Input*  
*On entry:* the number of classes desired in the frequency distribution,  $k$ . Whether or not class boundary values are user-supplied, K must include the two extreme classes which stretch to  $\pm\infty$ .  
*Constraint:*  $K \geq 2$ .
- 3: X(N) – *real* array *Input*  
*On entry:* the sample of observations of the variable for which the frequency distribution is required,  $x_i$ , for  $i = 1, 2, \dots, n$ . The values may be in any order.
- 4: ICLASS – INTEGER *Input*  
*On entry:* indicates whether class boundary values are to be calculated within the routine, or are supplied by the user.  
 If ICLASS  $\neq 1$ , then the class boundary values are to be calculated within the routine.  
 If ICLASS = 1, they are user-supplied.
- 5: CINT(K) – *real* array *Input/Output*  
*On entry:* if ICLASS = 0, then the elements of CINT need not be assigned values, as the routine calculates  $k - 1$  class boundary values.  
 If ICLASS = 1, the the first  $k - 1$  elements of CINT must contain the user-supplied class boundary values, in ascending order.  
 In both cases, the element CINT( $k$ ) need not be assigned, as it is not used in the routine.  
*On exit:* the first  $k - 1$  elements of CINT contain the class boundary values in ascending order.  
*Constraint:* if ICLASS = 1, CINT( $i$ ) < CINT( $i + 1$ ), for  $i = 1, 2, \dots, k - 2$ .
- 6: IFREQ(K) – INTEGER array *Output*  
*On exit:* the elements of IFREQ contain the frequencies in each class,  $f_i$  for  $i = 1, 2, \dots, k$ . In particular IFREQ(1) contains the frequency of the class up to CINT(1),  $f_1$ , and IFREQ( $k$ ) contains the frequency of the class greater than CINT( $k - 1$ ),  $f_k$ .
- 7: XMIN – *real* *Output*  
*On exit:* the smallest value in the sample,  $a$ .
- 8: XMAX – *real* *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. Users who are unfamiliar with this parameter should refer to Chapter P01 for details.  
*On exit:* IFAIL = 0 unless the routine detects an error (see Section 6).  
 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, for users 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.**

## 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.

## 7 Accuracy

The method used is believed to be stable.

## 8 Further Comments

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

## 9 Example

In the example program,  $NPROB$  determines the number of sets of data to be analysed. For each analysis the sample observations and optionally class boundary values, are read. After calling the routine the calculated frequency distribution and largest and smallest observations values are printed. In the example, there is one problem to be analysed, with 70 observations to be grouped into 5 routine-calculated classes.

### 9.1 Program Text

**Note:** the listing of the example program presented below uses *bold italicised* terms to denote precision-dependent details. Please read the Users' Note for your implementation to check the interpretation of these terms. As explained in the Essential Introduction to this manual, the results produced may not be identical for all implementations.

```
*      G01AEF Example Program Text
*      Mark 14 Revised.  NAG Copyright 1989.
*      .. Parameters ..
INTEGER          NMAX, K2MAX
PARAMETER       (NMAX=71, K2MAX=10)
INTEGER          NIN, NOUT
PARAMETER       (NIN=5, NOUT=6)
*      .. Local Scalars ..
real           XMAX, XMIN
INTEGER          I, ICLASS, IFAIL, J, N, NOC, NPROB
*      .. Local Arrays ..
real           A(NMAX), C(K2MAX)
INTEGER          JFREQ(K2MAX)
*      .. External Subroutines ..
EXTERNAL        G01AEF
*      .. Executable Statements ..
WRITE (NOUT,*) 'G01AEF Example Program Results'
*      Skip heading in data file
READ (NIN,*)
READ (NIN,*) NPROB
DO 20 I = 1, NPROB
    READ (NIN,*) N, ICLASS, NOC
    IF (N.GE.1 .AND. N.LE.NMAX .AND. NOC.GE.0 .AND. NOC.LE.K2MAX-2)
+      THEN
```

```

READ (NIN,*) (A(J),J=1,N)
WRITE (NOUT,*)
WRITE (NOUT,99997) 'Problem ', I
WRITE (NOUT,99997) 'Number of cases', N
WRITE (NOUT,99997) 'Number of classes', NOC
NOC = NOC + 1
IF (ICLASS.NE.1) THEN
  WRITE (NOUT,*) 'Routine-supplied class boundaries'
ELSE
  READ (NIN,*) (C(J),J=1,NOC)
  WRITE (NOUT,*) 'User-supplied class boundaries'
END IF
NOC = NOC + 1
IFAIL = 1
*
CALL G01AEF(N,NOC,A,ICLASS,C,JFREQ,XMIN,XMAX,IFAIL)
*
WRITE (NOUT,*)
IF (IFAIL.EQ.0) THEN
  WRITE (NOUT,*) 'Successful call of G01AEF'
  WRITE (NOUT,*)
  WRITE (NOUT,*) '*** Frequency distribution ***'
  WRITE (NOUT,*)
  WRITE (NOUT,*) '      Class              Frequency'
  WRITE (NOUT,*)
  WRITE (NOUT,99999) '  Up to      ', C(1), JFREQ(1)
  NOC = NOC - 1
  IF (NOC.GT.1) WRITE (NOUT,99998) (C(J-1),' to ',C(J),
+    JFREQ(J),J=2,NOC)
  WRITE (NOUT,99996) C(NOC), ' and over ', JFREQ(NOC+1)
  WRITE (NOUT,*)
  WRITE (NOUT,99995) 'Total frequency = ', N
  WRITE (NOUT,99994) 'Minimum = ', XMIN
  WRITE (NOUT,99994) 'Maximum = ', XMAX
ELSE
  WRITE (NOUT,99997)
+  'Unsuccessful call of G01AEF. IFAIL = ', IFAIL
  END IF
ELSE
  STOP
  END IF
20 CONTINUE
STOP
*
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

```

## 9.2 Program Data

G01AEF Example Program Data

```

1
 70      0      5
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

```

### 9.3 Program Results

G01AEF Example Program Results

Problem 1  
Number of cases 70  
Number of classes 5  
Routine-supplied class boundaries

Successful call of G01AEF

\*\*\* 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

---