NAG Library Routine Document

G01ADF

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

G01ADF calculates the mean, standard deviation and coefficients of skewness and kurtosis for data grouped in a frequency distribution.

2 Specification

```
SUBROUTINE G01ADF (K, X, IFREQ, XMEAN, S2, S3, S4, N, IFAIL)
INTEGER            K, IFREQ(K), N, IFAIL
REAL (KIND=nag_wp) X(K), XMEAN, S2, S3, S4
```

3 Description

The input data consist of a univariate frequency distribution, denoted by f_i, for $i = 1, 2, \ldots, k-1$, and the boundary values of the classes x_i, for $i = 1, 2, \ldots, k$. Thus the frequency associated with the interval (x_i, x_{i+1}) is f_i, and G01ADF assumes that all the values in this interval are concentrated at the point

$$y_i = (x_{i+1} + x_i)/2, \quad i = 1, 2, \ldots, k-1.$$

The following quantities are calculated:

(a) total frequency,

$$n = \sum_{i=1}^{k-1} f_i.$$

(b) mean,

$$\bar{y} = \frac{\sum_{i=1}^{k-1} f_i y_i}{n}.$$

(c) standard deviation,

$$s_2 = \sqrt{\frac{\sum_{i=1}^{k-1} f_i (y_i - \bar{y})^2}{(n-1)}}, \quad n \geq 2.$$

(d) coefficient of skewness,

$$s_3 = \frac{\sum_{i=1}^{k-1} f_i (y_i - \bar{y})^3}{(n-1) \times s_2^3}, \quad n \geq 2.$$

(e) coefficient of kurtosis,

$$s_4 = \frac{\sum_{i=1}^{k-1} f_i (y_i - \bar{y})^4}{(n-1) \times s_2^4} - 3, \quad n \geq 2.$$

The routine has been developed primarily for groupings of a continuous variable. If, however, the routine is to be used on the frequency distribution of a discrete variable, taking the values $y_1, \ldots, y_{k-1}$, then the boundary values for the classes may be defined as follows:

(i) for $k > 2$,

$$\begin{aligned} x_1 &= (3y_1 - y_2)/2 \\ x_j &= \big(y_{j-1} + y_j\big)/2, \qquad j = 2, \ldots, k-1 \\ x_k &= (3y_{k-1} - y_{k-2})/2 \end{aligned}$$

(ii) for $k = 2$,

$$x_1 = y_1 - a \quad \text{and} \quad x_2 = y_1 + a \quad \text{for any } a > 0.$$

4 References

None.

5 Parameters

1: K – INTEGER *Input*

On entry: k, the number of class boundaries, which is one more than the number of classes of the frequency distribution.

Constraint: K > 1.

2: X(K) – REAL (KIND=nag_wp) array *Input*

On entry: the elements of X must contain the boundary values of the classes in ascending order, so that class i is bounded by the values in X(i) and X$(i+1)$, for $i = 1, 2, \ldots, k-1$.

Constraint: X$(i) <$ X$(i+1)$, for $i = 1, 2, \ldots, k-1$.

3: IFREQ(K) – INTEGER array *Input*

On entry: the ith element of IFREQ must contain the frequency associated with the ith class, for $i = 1, 2, \ldots, k-1$. IFREQ(k) is not used by the routine.

Constraints:

IFREQ$(i) \geq 0$, for $i = 1, 2, \ldots, k-1$;
$\sum_{i=1}^{k-1}$IFREQ$(i) > 0$.

4: XMEAN – REAL (KIND=nag_wp) *Output*

On exit: the mean value, $\bar{y}$.

5: S2 – REAL (KIND=nag_wp) *Output*

On exit: the standard deviation, s_2.

6: S3 – REAL (KIND=nag_wp) *Output*

On exit: the coefficient of skewness, s_3.

7: S4 – REAL (KIND=nag_wp) *Output*

On exit: the coefficient of kurtosis, s_4.

8: N – INTEGER *Output*

On exit: the total frequency, n.

9: 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, K ≤ 1.

IFAIL $= 2$

On entry, the boundary values of the classes in X are not in ascending order.

IFAIL $= 3$

On entry, $\sum_{i=1}^{k-1}$IFREQ$(i) = 0$ or IFREQ$(i) < 0$ for some i, for $i = 1, 2, \ldots, k-1$.

IFAIL $= 4$

The total frequency, n, is less than 2, hence the quantities s_2, s_3 and s_4 cannot be calculated.

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

The method used is believed to be stable.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The time taken by G01ADF increases linearly with k.

10 Example

In the example program, NPROB determines the number of sets of data to be analysed. For each analysis, the boundary values of the classes and the frequencies are read. After G01ADF has been successfully called, the input data and calculated quantities are printed. In the example, there is one set of data, with 14 classes.

10.1 Program Text

```
    Program g01adfe

!     G01ADF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g01adf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: s2, s3, s4, xmean
      Integer                          :: i, ifail, k, kmin1, n
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: x(:)
      Integer, Allocatable             :: ifreq(:)
!     .. Executable Statements ..
      Write (nout,*) 'G01ADF Example Program Results'
      Write (nout,*)

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

!     Read in the problem size
      Read (nin,*) kmin1
      k = kmin1 + 1

      Allocate (ifreq(k),x(k))

!     Read in data
      Read (nin,*)(x(i),ifreq(i),i=1,kmin1), x(k)

!     Calculate summary statistics
      ifail = 0
      Call g01adf(k,x,ifreq,xmean,s2,s3,s4,n,ifail)

!     Display results
      Write (nout,99999) 'Number of classes ', kmin1
      Write (nout,*)
      Write (nout,*) '          Class          Frequency'
      Write (nout,*)
      Write (nout,99998)(x(i),x(i+1),ifreq(i),i=1,kmin1)
      Write (nout,*)
      Write (nout,99997) ' Mean ', xmean
      Write (nout,99996) ' Std devn', s2
      Write (nout,99996) ' Skewness', s3
      Write (nout,99996) ' Kurtosis', s4
      Write (nout,99995) ' Number of cases', n


99999 Format (1X,A,I4)
```

```
99998 Format (1X,2F10.2,I12)
99997 Format (1X,A,F16.4)
99996 Format (1X,A,F13.4)
99995 Format (1X,A,I8)
    End Program g01adfe
```

10.2 Program Data

```
G01ADF Example Program Data
14
     9.3       3      12      19      14      52      16      96
      18     121      20     115      22      86      24      70
      26      49      28      31      30      16      32       6
      34       8      36       7   39.7
```

10.3 Program Results

```
 G01ADF Example Program Results

 Number of classes   14

           Class          Frequency

       9.30     12.00           3
      12.00     14.00          19
      14.00     16.00          52
      16.00     18.00          96
      18.00     20.00         121
      20.00     22.00         115
      22.00     24.00          86
      24.00     26.00          70
      26.00     28.00          49
      28.00     30.00          31
      30.00     32.00          16
      32.00     34.00           6
      34.00     36.00           8
      36.00     39.70           7

  Mean          21.4932
  Std devn       4.9325
  Skewness       0.7072
  Kurtosis       0.5738
  Number of cases     679
```