NAG Library Routine Document

G08EDF

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

G08EDF performs a gaps test on a sequence of observations.

2 Specification

```
SUBROUTINE G08EDF (CL, N, X, M, MAXG, RLO, RUP, TOTLEN, NGAPS, NCOUNT,     &
                   EX, CHI, DF, PROB, IFAIL)

INTEGER            N, M, MAXG, NGAPS, NCOUNT(MAXG), IFAIL
REAL (KIND=nag_wp) X(N), RLO, RUP, TOTLEN, EX(MAXG), CHI, DF, PROB
CHARACTER(1)       CL
```

3 Description

Gaps tests are used to test for cyclical trend in a sequence of observations. G08EDF computes certain statistics for the gaps test.

G08EDF may be used in two different modes:

(i) a single call to G08EDF which computes all test statistics after counting the gaps;

(ii) multiple calls to G08EDF with the final test statistics only being computed in the last call.

The second mode is necessary if all the data does not fit into the memory. See argument CL in Section 5 for details on how to invoke each mode.

The term gap is used to describe the distance between two numbers in the sequence that lie in the interval (r_l, r_u). That is, a gap ends at x_j if $r_l \leq x_j \leq r_u$. The next gap then begins at x_{j+1}. The interval (r_l, r_u) should lie within the region of all possible numbers. For example if the test is carried out on a sequence of $(0, 1)$ random numbers then the interval (r_l, r_u) must be contained in the whole interval $(0, 1)$. Let t_{len} be the length of the interval which specifies all possible numbers.

G08EDF counts the number of gaps of different lengths. Let c_i denote the number of gaps of length i, for $i = 1, 2, \ldots, k-1$. The number of gaps of length k or greater is then denoted by c_k. An unfinished gap at the end of a sequence is not counted unless the sequence is part of an initial or intermediate call to G08EDF (i.e., unless there is another call to G08EDF to follow) in which case the unfinished gap is used. The following is a trivial example.

Suppose we called G08EDF twice (i.e., with CL = 'F' and then with CL = 'L') with the following two sequences and with RLO = 0.30 and RUP = 0.60:

(0.20 0.40 0.45 0.40 0.15 0.75 0.95 0.23) and

(0.27 0.40 0.25 0.10 0.34 0.39 0.61 0.12).

Then after the second call G08EDF would have counted the gaps of the following lengths:

2, 1, 1, 6, 3 and 1.

When the counting of gaps is complete G08EDF computes the expected values of the counts. An approximate χ^2 statistic with k degrees of freedom is computed where

$$X^2 = \frac{\sum_{i=1}^{k}(c_i - e_i)^2}{e_i},$$

where

$e_i = ngaps \times p \times (1-p)^{i-1}$, if $i < k$;

$e_i = ngaps \times (1-p)^{i-1}$, if $i = k$;

$ngaps =$ the number of gaps found and

$p = (r_u - r_l)/t_{\text{len}}$.

The use of the χ^2-distribution as an approximation to the exact distribution of the test statistic improves as the expected values increase.

You may specify the total number of gaps to be found. If the specified number of gaps is found before the end of a sequence G08EDF will exit before counting any further gaps.

4 References

Dagpunar J (1988) *Principles of Random Variate Generation* Oxford University Press

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

Morgan B J T (1984) *Elements of Simulation* Chapman and Hall

Ripley B D (1987) *Stochastic Simulation* Wiley

5 Arguments

1: CL – CHARACTER(1) *Input*

On entry: indicates the type of call to G08EDF.

CL = 'S'
This is the one and only call to G08EDF (single call mode). All data are to be input at once. All test statistics are computed after the counting of gaps is complete.

CL = 'F'
This is the first call to the routine. All initializations are carried out before the counting of gaps begins. The final test statistics are not computed since further calls will be made to G08EDF.

CL = 'I'
This is an intermediate call during which the counts of gaps are updated. The final test statistics are not computed since further calls will be made to G08EDF.

CL = 'L'
This is the last call to G08EDF. The test statistics are computed after the final counting of gaps is complete.

Constraint: CL = 'S', 'F', 'I' or 'L'.

2: N – INTEGER *Input*

On entry: n, the length of the current sequence of observations.

Constraint: N $\geq$ 1.

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

On entry: the sequence of observations.

4: M – INTEGER *Input*

On entry: the maximum number of gaps to be sought. If M ≤ 0 then there is no limit placed on the number of gaps that are found.

M should not be changed between calls to G08EDF.

Constraint: if CL = 'S', M $\leq$ N.

5: MAXG – INTEGER *Input*

On entry: k, the length of the longest gap for which tabulation is desired.

MAXG must not be changed between calls to G08EDF.

Constraints:

MAXG > 1;
if CL = 'S', MAXG $\leq$ N.

6: RLO – REAL (KIND=nag_wp) *Input*

On entry: the lower limit of the interval to be used to define the gaps, r_l.

RLO must not be changed between calls to G08EDF.

7: RUP – REAL (KIND=nag_wp) *Input*

On entry: the upper limit of the interval to be used to define the gaps, r_u.

RUP must not be changed between calls to G08EDF.

Constraint: RUP $>$ RLO.

8: TOTLEN – REAL (KIND=nag_wp) *Input*

On entry: the total length of the interval which contains all possible numbers that may arise in the sequence.

Constraint: TOTLEN > 0.0 and RUP $-$ RLO $<$ TOTLEN.

9: NGAPS – INTEGER *Input/Output*

On entry: if CL = 'S' or 'F', NGAPS need not be set.

If CL = 'I' or 'L', NGAPS must contain the value returned by the previous call to G08EDF.

On exit: the number of gaps actually found, $ngaps$.

10: NCOUNT(MAXG) – INTEGER array *Input/Output*

On entry: if CL = 'S' or 'F', NCOUNT need not be set.

If CL = 'I' or 'L', NCOUNT must contain the values returned by the previous call to G08EDF.

On exit: the counts of gaps of the different lengths, c_i, for $i = 1, 2, \ldots, k$.

11: EX(MAXG) – REAL (KIND=nag_wp) array *Output*

On exit: if CL = 'S' or 'L' (i.e., if it is a final exit) then EX contains the expected values of the counts.

Otherwise the elements of EX are not set.

12: CHI – REAL (KIND=nag_wp) *Output*

On exit: if CL = 'S' or 'L' (i.e., if it is a final exit) then CHI contains the χ^2 test statistic, X^2, for testing the null hypothesis of randomness.

Otherwise CHI is not set.

13: DF – REAL (KIND=nag_wp) *Output*

On exit: if CL = 'S' or 'L' (i.e., if it is a final exit) then DF contains the degrees of freedom for the χ^2 statistic.

Otherwise DF is not set.

14: PROB – REAL (KIND=nag_wp) *Output*

On exit: if CL = 'S' or 'L' (i.e., if it is a final exit) then PROB contains the upper tail probability associated with the χ^2 test statistic, i.e., the significance level.

Otherwise PROB is not set.

15: 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, because for this routine the values of the output arguments may be useful even if IFAIL $\neq 0$ on exit, the recommended value is -1. **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).

Note: G08EDF may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, CL $= \langle value \rangle$.
Constraint: CL = 'S', 'F', 'I' or 'L'.

IFAIL $= 2$

On entry, N $= \langle value \rangle$.
Constraint: N ≥ 1.

IFAIL $= 3$

On entry, M $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: if CL = 'S', M $\leq$ N.

IFAIL $= 4$

On entry, MAXG $= \langle value \rangle$.
Constraint: MAXG > 1.

On entry, MAXG $= \langle value \rangle$ and N $= \langle value \rangle$.
Constraint: if CL = 'S', MAXG $\leq$ N.

IFAIL $= 5$

On entry, RLO $= \langle value \rangle$, RUP $= \langle value \rangle$ and TOTLEN $= \langle value \rangle$.
Constraint: RUP $-$ RLO $<$ TOTLEN.

On entry, RLO $=\langle value\rangle$ and RUP $=\langle value\rangle$.
Constraint: RUP $>$ RLO.

On entry, TOTLEN $=\langle value\rangle$.
Constraint: TOTLEN > 0.0.

IFAIL $= 6$

No gaps were found. Try using a longer sequence, or increase the size of the interval RUP $-$ RLO.

IFAIL $= 7$

The expected frequency in class $i = \langle value\rangle$ is zero. The value of (RUP $-$ RLO)/TOTLEN may be too close to 0.0 or 1.0. or MAXG is too large relative to the number of gaps found.

IFAIL $= 8$

The number of gaps requested were not found, only $\langle value\rangle$ out of the requested $\langle value\rangle$ where found.
All statistics are returned and may still be of use.

IFAIL $= 9$

The expected frequency of at least one class is less than one.
This implies that the χ^2 may not be a very good approximation to the distribution of the test statistics.
All statistics are returned and may still be of use.

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 computations are believed to be stable. The computation of PROB given the values of CHI and DF will obtain a relative accuracy of five significant places for most cases.

8 Parallelism and Performance

G08EDF is not thread safe and should not be called from a multithreaded user program. Please see Section 3.12.1 in How to Use the NAG Library and its Documentation for more information on thread safety.

G08EDF is not threaded in any implementation.

9 Further Comments

The time taken by G08EDF increases with the number of observations n, and depends to some extent whether the call is an only, first, intermediate or last call.

10 Example

The following program performs the gaps test on 500 pseudorandom numbers. G08EDF is called 5 times with 100 observations on each call. All gaps of length 10 or more are counted together.

10.1 Program Text

```
    Program g08edfe

!     G08EDF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g08edf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: chi, df, prob, rlo, rup, totlen
      Integer                          :: i, ifail, m, maxg, n, ngaps, nsamp,  &
                                          pn
      Character (1)                    :: cl
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: ex(:), x(:)
      Integer, Allocatable             :: ncount(:)
!     .. Executable Statements ..
      Write (nout,*) 'G08EDF Example Program Results'
      Write (nout,*)

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

!     Read in number of samples and control parameters
      Read (nin,*) nsamp, m, maxg
      Read (nin,*) rlo, rup, totlen

      Allocate (ncount(maxg),ex(maxg),x(1))

      If (nsamp==1) Then
        cl = 'S'
      Else
        cl = 'F'
      End If

      pn = 0
      Do i = 1, nsamp
!       Skip run heading in data file
        Read (nin,*)

!       Read in sample size
        Read (nin,*) n

        If (n>pn) Then
!         Reallocate X if required
          Deallocate (x)
          Allocate (x(n))
          pn = n
        End If

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

!       Process the sample
        ifail = -1
        Call g08edf(cl,n,x,m,maxg,rlo,rup,totlen,ngaps,ncount,ex,chi,df,prob,  &
          ifail)
        If (ifail/=0 .And. ifail<8) Then
```

```
          Go To 100
        End If

!       Adjust CL for intermediate calls
        If (i<nsamp-1) Then
          cl = 'I'
        Else
          cl = 'L'
        End If

      End Do

!     Display results
      Write (nout,99999) 'Total number of gaps found = ', ngaps
      If (ifail==8) Then
        Write (nout,*)                                                  &
          ' ** Note : the number of gaps requested were not found.'
      End If
      Write (nout,*)
      Write (nout,*) 'Count'
      Write (nout,*)                                                    &
        '      0      1      2      3      4      5      6      7      8',     &
        '     >9'
      Write (nout,99998) ncount(1:maxg)
      Write (nout,*)
      Write (nout,*) 'Expect'
      Write (nout,*)                                                    &
        '      0      1      2      3      4      5      6      7      8',     &
        '     >9'
      Write (nout,99997) ex(1:maxg)
      Write (nout,*)
      Write (nout,99996) 'Chisq = ', chi
      Write (nout,99995) 'DF    = ', df
      Write (nout,99996) 'Prob  = ', prob
      If (ifail==9) Then
        Write (nout,*) ' ** Note : expected value <= 5.0'
        Write (nout,*)                                                  &
          '    the chi square approximation may not be very good.'
      End If

100   Continue

99999 Format (1X,A,I10)
99998 Format (1X,10I7)
99997 Format (1X,10F7.1)
99996 Format (1X,A,F10.4)
99995 Format (1X,A,F7.1)
    End Program g08edfe
```

10.2 Program Data

```
G08EDF Example Program Data
5  0  10           :: NSAMP,M,MAXG
0.4  0.6  1.0      :: RLO,RUP,TOTLEN
## Sample 1
100                :: N
0.11389 0.84996 0.84821 0.18431 0.14104 0.03144 0.68013 0.13297 0.27696 0.86743
0.32674 0.87990 0.85580 0.47830 0.75318 0.93643 0.19396 0.31091 0.34956 0.94923
0.18940 0.24715 0.62503 0.50406 0.05686 0.26481 0.68746 0.80387 0.48184 0.25034
0.20141 0.35062 0.58591 0.93407 0.93848 0.98496 0.66180 0.35957 0.71122 0.35875
0.96504 0.60832 0.36569 0.73499 0.25223 0.88296 0.06659 0.78113 0.40016 0.31768
0.47655 0.15008 0.20608 0.62633 0.62737 0.16400 0.44104 0.56993 0.13178 0.50499
0.44176 0.44385 0.75372 0.82178 0.60227 0.98944 0.33133 0.81067 0.40798 0.71608
0.69306 0.22144 0.47942 0.65697 0.50881 0.25223 0.82373 0.50148 0.65246 0.53275
0.92935 0.13455 0.19901 0.78844 0.14006 0.50600 0.41069 0.49703 0.47858 0.02210
0.91444 0.10784 0.54642 0.63091 0.14419 0.80457 0.51336 0.71451 0.12564 0.88051
## Sample 2
100                :: N
0.84976 0.63094 0.46109 0.80538 0.62387 0.90670 0.09969 0.67992 0.70503 0.09560
0.69991 0.37616 0.42030 0.23665 0.28771 0.24935 0.94950 0.12008 0.66217 0.20900
```

```
0.97026 0.98368 0.80206 0.43918 0.73232 0.03533 0.97995 0.06637 0.54726 0.48530
0.68865 0.94302 0.33718 0.61014 0.70127 0.36827 0.51335 0.24476 0.14203 0.02428
0.73691 0.22192 0.40374 0.85757 0.83335 0.73309 0.05563 0.17332 0.72253 0.43291
0.77476 0.35967 0.94242 0.61337 0.43513 0.80573 0.70630 0.83115 0.24622 0.45445
0.53595 0.31476 0.87968 0.75365 0.86291 0.34051 0.62232 0.16762 0.45506 0.15561
0.76615 0.77421 0.06035 0.72290 0.93712 0.83223 0.40044 0.96575 0.73176 0.27827
0.02174 0.75326 0.82876 0.64979 0.98038 0.61054 0.87742 0.95273 0.39091 0.42146
0.89020 0.08617 0.90953 0.00416 0.70915 0.21123 0.95342 0.19269 0.68252 0.27600
## Sample 3
100                :: N
0.40629 0.96486 0.66026 0.07134 0.35492 0.34348 0.87164 0.59746 0.43724 0.26730
0.11840 0.04604 0.49037 0.99669 0.32784 0.34772 0.93599 0.95806 0.80635 0.18897
0.60061 0.83359 0.63026 0.14084 0.05323 0.70247 0.28532 0.09572 0.36153 0.50378
0.42679 0.71801 0.51010 0.72090 0.97537 0.29919 0.30059 0.23610 0.25668 0.07510
0.92481 0.65715 0.69686 0.27840 0.20555 0.64015 0.05725 0.25120 0.32288 0.22320
0.16582 0.71466 0.34030 0.55575 0.51468 0.18013 0.74670 0.21455 0.52649 0.47487
0.85805 0.24616 0.11459 0.38690 0.83475 0.83629 0.83754 0.18998 0.46715 0.24162
0.19488 0.03281 0.39291 0.37834 0.97169 0.65229 0.88913 0.53777 0.05780 0.20468
0.33788 0.10130 0.72771 0.31306 0.74279 0.26546 0.37941 0.04878 0.03061 0.52394
0.74104 0.97192 0.04550 0.81382 0.44430 0.32402 0.06791 0.73602 0.22640 0.67260
## Sample 4
100                :: N
0.46016 0.95901 0.37581 0.45836 0.26220 0.30389 0.46845 0.52940 0.71121 0.89187
0.33346 0.81783 0.07194 0.01163 0.63324 0.69208 0.28685 0.02491 0.97931 0.53225
0.47009 0.12105 0.80291 0.21191 0.74158 0.78269 0.30493 0.06901 0.54152 0.88463
0.60358 0.81066 0.77771 0.74140 0.65465 0.32613 0.42757 0.36584 0.42506 0.39980
0.04686 0.79805 0.53593 0.15562 0.09924 0.68011 0.61072 0.88701 0.56239 0.64343
0.19223 0.07325 0.40971 0.85265 0.27507 0.88884 0.10551 0.62646 0.11055 0.91368
0.58845 0.68942 0.29994 0.30395 0.45696 0.88127 0.38773 0.12028 0.48981 0.28535
0.84174 0.46451 0.17140 0.90827 0.49424 0.29557 0.25788 0.76838 0.19073 0.26051
0.47442 0.03224 0.32034 0.97378 0.43992 0.13338 0.45850 0.02122 0.30482 0.49427
0.89839 0.01770 0.85679 0.90157 0.29537 0.15213 0.21464 0.37237 0.86199 0.60364
## Sample 5
100                :: N
0.66793 0.00711 0.17970 0.98702 0.50449 0.88105 0.08259 0.77263 0.06050 0.73389
0.86517 0.76088 0.40239 0.50178 0.13811 0.63441 0.91949 0.48518 0.96923 0.08820
0.14556 0.28177 0.99598 0.46908 0.83279 0.26252 0.64987 0.20426 0.41060 0.76120
0.78022 0.44662 0.04918 0.36644 0.62337 0.16849 0.63846 0.41247 0.54464 0.05721
0.79852 0.23048 0.76139 0.22493 0.45640 0.07671 0.96152 0.50771 0.02376 0.49537
0.07095 0.86385 0.71385 0.35192 0.68827 0.49737 0.44847 0.26744 0.46983 0.44270
0.78845 0.72560 0.38886 0.45552 0.45917 0.64241 0.44654 0.42665 0.01122 0.76716
0.01727 0.33687 0.02836 0.48409 0.02777 0.63643 0.59711 0.02880 0.63758 0.56746
0.41342 0.40939 0.61578 0.89186 0.70151 0.38707 0.94021 0.17271 0.27477 0.04308
0.91821 0.97517 0.57249 0.14325 0.46058 0.26434 0.85927 0.77526 0.64717 0.08314
```

10.3 Program Results

```
 G08EDF Example Program Results

 Total number of gaps found =         99

 Count
       0      1      2      3      4      5      6      7      8     >9
      22     11     10     13      6     12      4      6      2     13

 Expect
       0      1      2      3      4      5      6      7      8     >9
    19.8   15.8   12.7   10.1    8.1    6.5    5.2    4.2    3.3   13.3

 Chisq =     9.9540
 DF    =     9.0
 Prob  =     0.3542
```