NAG Library Routine Document

G10ZAF

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

G10ZAF orders and weights data which is entered unsequentially, weighted or unweighted.

2 Specification

```
SUBROUTINE G10ZAF (WEIGHT, N, X, Y, WT, NORD, XORD, YORD, WTORD, RSS,      &
                   IWRK, IFAIL)
INTEGER            N, NORD, IWRK(N), IFAIL
REAL (KIND=nag_wp) X(N), Y(N), WT(*), XORD(N), YORD(N), WTORD(N), RSS
CHARACTER(1)       WEIGHT
```

3 Description

Given a set of observations (x_i, y_i), for $i = 1, 2, \ldots, n$, with corresponding weights w_i, G10ZAF rearranges the observations so that the x_i are in ascending order.

For any equal x_i in the ordered set, say $x_j = x_{j+1} = \cdots = x_{j+k}$, a single observation x_j is returned with a corresponding y' and w', calculated as

$$w' = \sum_{l=0}^{k} w_{i+l}$$

and

$$y' = \frac{\sum_{l=0}^{k} w_{i+l} y_{i+l}}{w'}.$$

Observations with zero weight are ignored. If no weights are supplied by you, then unit weights are assumed; that is $w_i = 1$, for $i = 1, 2, \ldots, n$.

In addition, the within group sum of squares is computed for the tied observations using West's algorithm (see West (1979)).

4 References

Draper N R and Smith H (1985) *Applied Regression Analysis* (2nd Edition) Wiley

West D H D (1979) Updating mean and variance estimates: An improved method *Comm. ACM* **22** 532–555

5 Arguments

1: WEIGHT – CHARACTER(1) *Input*

On entry: indicates whether user-defined weights are to be used.

If WEIGHT = 'W', user-defined weights are to be used and must be supplied in WT.

If WEIGHT = 'U', the data is treated as unweighted.

Constraint: WEIGHT = 'W' or 'U'.

2: N – INTEGER *Input*

On entry: n, the number of observations.

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

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

On entry: the values, x_i, for $i = 1, 2, \ldots, n$.

4: Y(N) – REAL (KIND=nag_wp) array *Input*

On entry: the values y_i, for $i = 1, 2, \ldots, n$.

5: WT($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array WT must be at least N if WEIGHT = 'W'.

On entry: if WEIGHT = 'W', WT must contain the n weights. Otherwise WT is not referenced and unit weights are assumed.

Constraints:

if WEIGHT = 'W', $\mathrm{WT}(i) > 0.0$, for $i = 1, 2, \ldots, n$;
if WEIGHT = 'W', $\sum_{i=1}^{n} \mathrm{WT}(i) > 0$.

6: NORD – INTEGER *Output*

On exit: the number of distinct observations.

7: XORD(N) – REAL (KIND=nag_wp) array *Output*

On exit: the first NORD elements contain the ordered and distinct x_i.

8: YORD(N) – REAL (KIND=nag_wp) array *Output*

On exit: the first NORD elements contain the values y' corresponding to the values in XORD.

9: WTORD(N) – REAL (KIND=nag_wp) array *Output*

On exit: the first NORD elements contain the values w' corresponding to the values of XORD and YORD.

10: RSS – REAL (KIND=nag_wp) *Output*

On exit: the within group sum of squares for tied observations.

11: IWRK(N) – INTEGER array *Workspace*

12: 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, WEIGHT $\neq$ 'W' or 'U',
or N < 1.

IFAIL $= 2$

On entry, WEIGHT $=$ 'W' and at least one element of WT is < 0.0, or all elements of WT are 0.0.

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

For a discussion on the accuracy of the algorithm for computing mean and variance see West (1979).

8 Parallelism and Performance

G10ZAF is not threaded in any implementation.

9 Further Comments

G10ZAF may be used to compute the pure error sum of squares in simple linear regression along with G02DAF; see Draper and Smith (1985).

10 Example

A set of unweighted observations are input and G10ZAF used to produce a set of strictly increasing weighted observations.

10.1 Program Text

```
    Program g10zafe

!     G10ZAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g10zaf, nag_wp
!     .. Implicit None Statement ..
```

```
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rss
      Integer                          :: i, ifail, lwt, n, nord
      Character (1)                    :: weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: wt(:), wtord(:), x(:), xord(:),      &
                                          y(:), yord(:)
      Integer, Allocatable             :: iwrk(:)
!     .. Executable Statements ..
      Write (nout,*) 'G10ZAF Example Program Results'
      Write (nout,*)

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

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

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      Allocate (x(n),y(n),iwrk(n),wt(lwt),xord(n),yord(n),wtord(n))

!     Read in data
      If (lwt>0) Then
        Read (nin,*)(x(i),y(i),wt(i),i=1,n)
      Else
        Read (nin,*)(x(i),y(i),i=1,n)
      End If

!     Reorder data
      ifail = 0
      Call g10zaf(weight,n,x,y,wt,nord,xord,yord,wtord,rss,iwrk,ifail)

!     Display results
      Write (nout,99999) 'Number of distinct observations = ', nord
      Write (nout,99998) 'Residual sum of squares         = ', rss
      Write (nout,*)
      Write (nout,*) '              X                 Y                 WT'
      Write (nout,99997)(xord(i),yord(i),wtord(i),i=1,nord)

99999 Format (1X,A,I6)
99998 Format (1X,A,F13.5)
99997 Format (5X,F13.5,5X,F13.5,5X,F13.5)
    End Program g10zafe
```

10.2 Program Data

```
G10ZAF Example Program Data
'U' 10    :: WEIGHT,N
1.0 4.0
3.0 4.0
5.0 1.0
5.0 2.0
3.0 5.0
4.0 3.0
9.0 4.0
6.0 9.0
9.0 7.0
9.0 4.0   :: End of X,Y
```

10.3 Program Results

```
 G10ZAF Example Program Results

 Number of distinct observations =      6
 Residual sum of squares         =       7.00000

           X                 Y                 WT
        1.00000           4.00000           1.00000
        3.00000           4.50000           2.00000
        4.00000           3.00000           1.00000
        5.00000           1.50000           2.00000
        6.00000           9.00000           1.00000
        9.00000           5.00000           3.00000
```