

# NAG Library Routine Document

## G01DBF

**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

G01DBF calculates an approximation to the set of Normal Scores, i.e., the expected values of an ordered set of independent observations from a Normal distribution with mean 0.0 and standard deviation 1.0.

### 2 Specification

```
SUBROUTINE G01DBF (N, PP, IFAIL)
  INTEGER          N, IFAIL
  REAL (KIND=nag_wp) PP(N)
```

### 3 Description

G01DBF is an adaptation of the Applied Statistics Algorithm AS 177.3, see Royston (1982). If you are particularly concerned with the accuracy with which G01DBF computes the expected values of the order statistics (see Section 7), then G01DAF which is more accurate should be used instead at a cost of increased storage and computing time.

Let  $x_{(1)}, x_{(2)}, \dots, x_{(n)}$  be the order statistics from a random sample of size  $n$  from the standard Normal distribution. Defining

$$P_{r,n} = \Phi(-E(x_{(r)}))$$

and

$$Q_{r,n} = \frac{r - \epsilon}{n + \gamma}, \quad r = 1, 2, \dots, n,$$

where  $E(x_{(r)})$  is the expected value of  $x_{(r)}$ , the current routine approximates the Normal upper tail area corresponding to  $E(x_{(r)})$  as,

$$\tilde{P}_{r,n} = Q_{r,n} + \frac{\delta_1}{n} Q_{r,n}^\lambda + \frac{\delta_2}{n} Q_{r,n}^{2\lambda} - C_{r,n}.$$

for  $r = 1, 2, 3$ , and  $r \geq 4$ . Estimates of  $\epsilon$ ,  $\gamma$ ,  $\delta_1$ ,  $\delta_2$  and  $\lambda$  are obtained. A small correction  $C_{r,n}$  to  $\tilde{P}_{r,n}$  is necessary when  $r \leq 7$  and  $n \leq 20$ .

The approximation to  $E(X_{(r)})$  is thus given by

$$E(x_{(r)}) = -\Phi^{-1}(\tilde{P}_{r,n}), \quad r = 1, 2, \dots, n.$$

Values of the inverse Normal probability integral  $\Phi^{-1}$  are obtained from G01FAF.

### 4 References

Royston J P (1982) Algorithm AS 177: expected normal order statistics (exact and approximate) *Appl. Statist.* **31** 161–165

## 5 Arguments

- 1: N – INTEGER *Input*  
*On entry:*  $n$ , the size of the sample.  
*Constraint:*  $N \geq 1$ .
- 2: PP(N) – REAL (KIND=nag\_wp) array *Output*  
*On exit:* the Normal scores. PP( $i$ ) contains the value  $E(x_{(i)})$ , for  $i = 1, 2, \dots, n$ .
- 3: 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,  $N < 1$ .

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  $n \leq 2000$ , the maximum error is 0.0001, but G01DBF is usually accurate to 5 or 6 decimal places. For  $n$  up to 5000, comparison with the exact scores calculated by G01DAF shows that the maximum error is 0.001.

## 8 Parallelism and Performance

G01DBF is not threaded in any implementation.

## 9 Further Comments

The time taken by G01DBF is proportional to  $n$ .

## 10 Example

A program to calculate the expected values of the order statistics for a sample of size 10.

### 10.1 Program Text

```

Program g01dbfe

!      G01DBF Example Program Text

!      Mark 26 Release. NAG Copyright 2016.

!      .. Use Statements ..
Use nag_library, Only: g01dbf, nag_wp
!      .. Implicit None Statement ..
Implicit None
!      .. Parameters ..
Integer, Parameter          :: nout = 6
!      .. Local Scalars ..
Integer                    :: ifail, n
!      .. Local Arrays ..
Real (Kind=nag_wp), Allocatable :: pp(:)
!      .. Executable Statements ..
Write (nout,*) 'G01DBF Example Program Results'
Write (nout,*)

!      Set the problem size
n = 10

Allocate (pp(n))

!      Calculate the normal scores
ifail = 0
Call g01dbf(n,pp,ifail)

!      Display results
Write (nout,99999) 'Sample size = ', n
Write (nout,*) 'Normal scores'
Write (nout,99998) pp(1:n)

99999 Format (1X,A,I2)
99998 Format (10X,5F12.4)
End Program g01dbfe

```

### 10.2 Program Data

None.

### 10.3 Program Results

```

G01DBF Example Program Results

Sample size = 10
Normal scores
      -1.5388      -1.0014      -0.6561      -0.3757      -0.1227
      0.1227       0.3757       0.6561       1.0014       1.5388

```

This shows a Q-Q plot for a randomly generated set of data. The normal scores have been calculated using G01DBF and the sample quantiles obtained by sorting the observed data using M01CAF. A reference line at  $y = x$  is also shown.

**Example Program**  
Q-Q Plot for a Random Set of Data  
Using Approximate Normal Scores

