NAG Library Routine Document

G01BJF

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

G01BJF returns the lower tail, upper tail and point probabilities associated with a binomial distribution.

2 Specification

```
SUBROUTINE G01BJF (N, P, K, PLEK, PGTK, PEQK, IFAIL)
INTEGER            N, K, IFAIL
REAL (KIND=nag_wp) P, PLEK, PGTK, PEQK
```

3 Description

Let X denote a random variable having a binomial distribution with parameters n and p ($n \geq 0$ and $0 < p < 1$). Then

$$\text{Prob}\{X = k\} = \binom{n}{k} p^k (1-p)^{n-k}, \quad k = 0, 1, \ldots, n.$$

The mean of the distribution is np and the variance is $np(1-p)$.

G01BJF computes for given n, p and k the probabilities:

$$\begin{aligned} \text{PLEK} &= \text{Prob}\{X \leq k\} \\ \text{PGTK} &= \text{Prob}\{X > k\} \\ \text{PEQK} &= \text{Prob}\{X = k\}. \end{aligned}$$

The method is similar to the method for the Poisson distribution described in KnÏsel (1986).

4 References

KnÏsel L (1986) Computation of the chi-square and Poisson distribution *SIAM J. Sci. Statist. Comput.* **7** 1022–1036

5 Arguments

1: N – INTEGER *Input*

On entry: the parameter n of the binomial distribution.

Constraint: N $\geq$ 0.

2: P – REAL (KIND=nag_wp) *Input*

On entry: the parameter p of the binomial distribution.

Constraint: 0.0 $<$ P $<$ 1.0.

3: K – INTEGER *Input*

On entry: the integer k which defines the required probabilities.

Constraint: 0 $\leq$ K $\leq$ N.

4: PLEK – REAL (KIND=nag_wp) *Output*

On exit: the lower tail probability, Prob$\{X \leq k\}$.

5: PGTK – REAL (KIND=nag_wp) *Output*

On exit: the upper tail probability, Prob$\{X > k\}$.

6: PEQK – REAL (KIND=nag_wp) *Output*

On exit: the point probability, Prob$\{X = k\}$.

7: 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 < 0.

IFAIL $= 2$

On entry, P ≤ 0.0,
or P ≥ 1.0.

IFAIL $= 3$

On entry, K < 0,
or K $>$ N.

IFAIL $= 4$

On entry, N is too large to be represented exactly as a real number.

IFAIL $= 5$

On entry, the variance ($= np(1-p)$) exceeds 10^6.

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

Results are correct to a relative accuracy of at least 10^{-6} on machines with a precision of 9 or more decimal digits, and to a relative accuracy of at least 10^{-3} on machines of lower precision (provided that the results do not underflow to zero).

8 Parallelism and Performance

G01BJF is not threaded in any implementation.

9 Further Comments

The time taken by G01BJF depends on the variance ($= np(1-p)$) and on k. For given variance, the time is greatest when $k \approx np$ (= the mean), and is then approximately proportional to the square-root of the variance.

10 Example

This example reads values of n and p from a data file until end-of-file is reached, and prints the corresponding probabilities.

10.1 Program Text

```
    Program g01bjfe

!     G01BJF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g01bjf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: p, peqk, pgtk, plek
      Integer                          :: ifail, k, n
!     .. Executable Statements ..
      Write (nout,*) 'G01BJF Example Program Results'
      Write (nout,*)

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

!     Display titles
      Write (nout,*) '   N     P      K     PLEK      PGTK      PEQK'
      Write (nout,*)

!     Loop over all data
d_lp: Do
        Read (nin,*,Iostat=ifail) n, p, k
        If (ifail/=0) Then
!         All data processed
          Exit d_lp
        End If

!       Calculate probability
        ifail = 0
```

```
        Call g01bjf(n,p,k,plek,pgtk,peqk,ifail)

!       Display results
        Write (nout,99999) n, p, k, plek, pgtk, peqk
      End Do d_lp

99999 Format (1X,I4,F8.3,I5,3F10.5)
    End Program g01bjfe
```

10.2 Program Data

```
G01BJF Example Program Data
   4  0.50    2   : N, P, K
  19  0.44   13
 100  0.75   67
2000  0.33  700
```

10.3 Program Results

```
 G01BJF Example Program Results

    N     P       K     PLEK      PGTK      PEQK

    4   0.500     2   0.68750   0.31250   0.37500
   19   0.440    13   0.99138   0.00862   0.01939
  100   0.750    67   0.04460   0.95540   0.01700
 2000   0.330   700   0.97251   0.02749   0.00312
```