NAG Library Routine Document

G01SAF

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

G01SAF returns a number of one or two tail probabilities for the Normal distribution.

2 Specification

```
SUBROUTINE G01SAF (LTAIL, TAIL, LX, X, LXMU, XMU, LXSTD, XSTD, P, IVALID,      &
                   IFAIL)
INTEGER            LTAIL, LX, LXMU, LXSTD, IVALID(*), IFAIL
REAL (KIND=nag_wp) X(LX), XMU(LXMU), XSTD(LXSTD), P(*)
CHARACTER(1)       TAIL(LTAIL)
```

3 Description

The lower tail probability for the Normal distribution, $P(X_i \leq x_i)$ is defined by:

$$P(X_i \leq x_i) = \int_{-\infty}^{x_i} Z_i(X_i)\, dX_i,$$

where

$$Z_i(X_i) = \frac{1}{\sqrt{2\pi{\sigma_i}^2}} e^{-(X_i-\mu_i)^2/\left(2{\sigma_i}^2\right)}, -\infty < X_i < \infty.$$

The relationship

$$P(X_i \leq x_i) = \tfrac{1}{2}\mathrm{erfc}\left(\frac{-(x_i-\mu_i)}{\sqrt{2}\sigma_i}\right)$$

is used, where erfc is the complementary error function, and is computed using S15ADF.

When the two tail confidence probability is required the relationship

$$P(X_i \leq |x_i|) - P(X_i \leq -|x_i|) = \mathrm{erf}\left(\frac{|x_i-\mu_i|}{\sqrt{2}\sigma_i}\right),$$

is used, where erf is the error function, and is computed using S15AEF.

The input arrays to this routine are designed to allow maximum flexibility in the supply of vector parameters by re-using elements of any arrays that are shorter than the total number of evaluations required. See Section 2.6 in the G01 Chapter Introduction for further information.

4 References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* (3rd Edition) Dover Publications

Hastings N A J and Peacock J B (1975) *Statistical Distributions* Butterworth

5 Parameters

1: LTAIL – INTEGER *Input*

On entry: the length of the array TAIL.

Constraint: LTAIL > 0.

2: TAIL(LTAIL) – CHARACTER(1) array *Input*

On entry: indicates which tail the returned probabilities should represent. Letting Z denote a variate from a standard Normal distribution, and $z_i = \frac{x_i - \mu_i}{\sigma_i}$, then for $j = ((i-1) \bmod \text{LTAIL}) + 1$, for $i = 1, 2, \ldots, \max(\text{LX}, \text{LTAIL}, \text{LXMU}, \text{LXSTD})$:

TAIL(j) = 'L'
The lower tail probability is returned, i.e., $p_i = P(Z \leq z_i)$.

TAIL(j) = 'U'
The upper tail probability is returned, i.e., $p_i = P(Z \geq z_i)$.

TAIL(j) = 'C'
The two tail (confidence interval) probability is returned, i.e., $p_i = P(Z \leq |z_i|) - P(Z \leq -|z_i|)$.

TAIL(j) = 'S'
The two tail (significance level) probability is returned, i.e., $p_i = P(Z \geq |z_i|) + P(Z \leq -|z_i|)$.

Constraint: TAIL(j) = 'L', 'U', 'C' or 'S', for $j = 1, 2, \ldots, \text{LTAIL}$.

3: LX – INTEGER *Input*

On entry: the length of the array X.

Constraint: LX > 0.

4: X(LX) – REAL (KIND=nag_wp) array *Input*

On entry: x_i, the Normal variate values with $x_i = \text{X}(j)$, $j = ((i-1) \bmod \text{LX}) + 1$.

5: LXMU – INTEGER *Input*

On entry: the length of the array XMU.

Constraint: LXMU > 0.

6: XMU(LXMU) – REAL (KIND=nag_wp) array *Input*

On entry: μ_i, the means with $\mu_i = \text{XMU}(j)$, $j = ((i-1) \bmod \text{LXMU}) + 1$.

7: LXSTD – INTEGER *Input*

On entry: the length of the array XSTD.

Constraint: LXSTD > 0.

8: XSTD(LXSTD) – REAL (KIND=nag_wp) array *Input*

On entry: σ_i, the standard deviations with $\sigma_i = \text{XSTD}(j)$, $j = ((i-1) \bmod \text{LXSTD}) + 1$.

Constraint: XSTD$(j) > 0.0$, for $j = 1, 2, \ldots, \text{LXSTD}$.

9: P$(*)$ – REAL (KIND=nag_wp) array *Output*

Note: the dimension of the array P must be at least $\max(\text{LX}, \text{LTAIL}, \text{LXMU}, \text{LXSTD})$.

On exit: p_i, the probabilities for the Normal distribution.

10: IVALID($*$) – INTEGER array *Output*

Note: the dimension of the array IVALID must be at least max(LX, LTAIL, LXMU, LXSTD).

On exit: IVALID(i) indicates any errors with the input arguments, with

IVALID(i) $= 0$
No error.

IVALID(i) $= 1$

On entry, invalid value supplied in TAIL when calculating p_i.

IVALID(i) $= 2$

On entry, $\sigma_i \leq 0.0$.

11: 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, at least one value of TAIL or XSTD was invalid.
Check IVALID for more information.

IFAIL $= 2$

On entry, LTAIL $= \langle value \rangle$.
Constraint: LTAIL > 0.

IFAIL $= 3$

On entry, LX $= \langle value \rangle$.
Constraint: LX > 0.

IFAIL $= 4$

On entry, LXMU $= \langle value \rangle$.
Constraint: LXMU > 0.

IFAIL $= 5$

On entry, LXSTD $= \langle value \rangle$.
Constraint: LXSTD > 0.

IFAIL $= -999$

Dynamic memory allocation failed.

7 Accuracy

Accuracy is limited by ***machine precision***. For detailed error analysis see S15ADF and S15AEF.

8 Further Comments

None.

9 Example

Four values of TAIL, X, XMU and XSTD are input and the probabilities calculated and printed.

9.1 Program Text

```
    Program g01safe
!     G01SAF Example Program Text

!     Mark 24 Release. NAG Copyright 2012.

!     .. Use Statements ..
      Use nag_library, Only: g01saf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lout, ltail, lx, lxmu, lxstd
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: p(:), x(:), xmu(:), xstd(:)
      Integer, Allocatable             :: ivalid(:)
      Character (1), Allocatable       :: tail(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: max, mod, repeat
!     .. Executable Statements ..
      Write (nout,*) 'G01SAF Example Program Results'
      Write (nout,*)

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

!     Read in the input vectors
      Read (nin,*) ltail
      Allocate (tail(ltail))
      Read (nin,*) tail(1:ltail)

      Read (nin,*) lx
      Allocate (x(lx))
      Read (nin,*) x(1:lx)

      Read (nin,*) lxmu
      Allocate (xmu(lxmu))
      Read (nin,*) xmu(1:lxmu)

      Read (nin,*) lxstd
      Allocate (xstd(lxstd))
      Read (nin,*) xstd(1:lxstd)

!     Allocate memory for output
      lout = max(ltail,lx,lxmu,lxstd)
      Allocate (p(lout),ivalid(lout))

!     Calculate probability
      ifail = -1
      Call g01saf(ltail,tail,lx,x,lxmu,xmu,lxstd,xstd,p,ivalid,ifail)

      If (ifail==0 .Or. ifail==1) Then
!       Display title
```

```
        Write (nout,*) &
          '   TAIL      X        XMU      XSTD       P     IVALID'
        Write (nout,*) repeat('-',56)

!       Display results
        Do i = 1, lout
          Write (nout,99999) tail(mod(i-1,ltail)+1), x(mod(i-1,lx)+1), &
            xmu(mod(i-1,lxmu)+1), xstd(mod(i-1,lxstd)+1), p(i), ivalid(i)
        End Do
      End If

99999 Format (5X,A1,3(4X,F6.2),4X,F6.3,4X,I3)
    End Program g01safe
```

9.2 Program Data

```
G01SAF Example Program Data
4                       :: LTAIL
'L' 'U' 'C' 'S'         :: TAIL
1                       :: LX
1.96                    :: X
1                       :: LXMU
0.0                     :: XMU
1                       :: LXSTD
1.0                     :: XSTD
```

9.3 Program Results

```
 G01SAF Example Program Results

    TAIL      X        XMU      XSTD       P     IVALID
 ------------------------------------------------------------
     L      1.96      0.00      1.00     0.975      0
     U      1.96      0.00      1.00     0.025      0
     C      1.96      0.00      1.00     0.950      0
     S      1.96      0.00      1.00     0.050      0
```