NAG Toolbox

nag_stat_prob_normal_vector (g01sa)

1 Purpose

nag_stat_prob_normal_vector (g01sa) returns a number of one or two tail probabilities for the Normal distribution.

2 Syntax

```
[p, ivalid, ifail] = nag_stat_prob_normal_vector(tail, x, xmu, xstd, 'ltail', ltail, 'lx', lx, 'lxmu', lxmu, 'lxstd', lxstd)
[p, ivalid, ifail] = g01sa(tail, x, xmu, xstd, 'ltail', ltail, 'lx', lx, 'lxmu', lxmu, 'lxstd', lxstd)
```

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

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

is used, where erfc is the complementary error function, and is computed using nag_specfun_erfc_real (s15ad).

When the two tail confidence probability is required the relationship

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

is used, where erf is the error function, and is computed using nag_specfun_erf_real (s15ae).

The input arrays to this function are designed to allow maximum flexibility in the supply of vector arguments 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

5.1 Compulsory Input Parameters

1: **tail**(**ltail**) – CHARACTER(1) array

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 $j = ((i-1) \bmod \mathbf{ltail}) + 1$, for $i = 1, 2, \ldots, \max(\mathbf{lx}, \mathbf{ltail}, \mathbf{lxmu}, \mathbf{lxstd})$:

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 $j = 1, 2, \ldots,$ **ltail**.

2: **x**(**lx**) – REAL (KIND=nag_wp) array

x_i, the Normal variate values with $x_i =$ **x**(j), $j = ((i-1) \bmod \mathbf{lx}) + 1$.

3: **xmu**(**lxmu**) – REAL (KIND=nag_wp) array

μ_i, the means with $\mu_i =$ **xmu**(j), $j = ((i-1) \bmod \mathbf{lxmu}) + 1$.

4: **xstd**(**lxstd**) – REAL (KIND=nag_wp) array

σ_i, the standard deviations with $\sigma_i =$ **xstd**(j), $j = ((i-1) \bmod \mathbf{lxstd}) + 1$.

Constraint: **xstd**$(j) > 0.0$, for $j = 1, 2, \ldots,$ **lxstd**.

5.2 Optional Input Parameters

1: **ltail** – INTEGER

Default: the dimension of the array **tail**.

The length of the array **tail**.

Constraint: **ltail** > 0.

2: **lx** – INTEGER

Default: the dimension of the array **x**.

The length of the array **x**.

Constraint: **lx** > 0.

3: **lxmu** – INTEGER

Default: the dimension of the array **xmu**.

The length of the array **xmu**.

Constraint: **lxmu** > 0.

4: **lxstd** – INTEGER

Default: the dimension of the array **xstd**.

The length of the array **xstd**.

Constraint: **lxstd** > 0.

5.3 Output Parameters

1: **p**(:) – REAL (KIND=nag_wp) array

The dimension of the array **p** will be $\max(\mathbf{lx}, \mathbf{ltail}, \mathbf{lxmu}, \mathbf{lxstd})$

p_i, the probabilities for the Normal distribution.

2: **ivalid**(:) – INTEGER array

The dimension of the array **ivalid** will be $\max(\mathbf{lx}, \mathbf{ltail}, \mathbf{lxmu}, \mathbf{lxstd})$

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, $\sigma_i \leq 0.0$.

3: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$ (*warning*)

On entry, at least one value of **tail** or **xstd** was invalid.
Check **ivalid** for more information.

ifail $= 2$

Constraint: **ltail** > 0.

ifail $= 3$

Constraint: **lx** > 0.

ifail $= 4$

Constraint: **lxmu** > 0.

ifail $= 5$

Constraint: **lxstd** > 0.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Accuracy is limited by ***machine precision***. For detailed error analysis see nag_specfun_erfc_real (s15ad) and nag_specfun_erf_real (s15ae).

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

```
function g01sa_example

fprintf('g01sa example results\n\n');

x    = [1.96; 1.96; 1.96; 1.96];
xmu  = [0; 0; 0; 0];
xstd = [1; 1; 1; 1];
tail = {'L'; 'U'; 'C'; 'S'};

% calculate probability
[prob, ivalid, ifail] = g01sa( ...
                              tail, x, xmu, xstd);

fprintf('tail    x       xmu      xstd    probability\n');
lx    = numel(x);
lxmu  = numel(xmu);
lxstd = numel(xstd);
ltail = numel(tail);
len   = max ([lx, lxmu, lxstd, ltail]);
for i=0:len-1
  fprintf(' %c %8.2f %8.2f %8.2f %8.3f\n', tail{mod(i,ltail)+1}, ...
          x(mod(i,lx)+1), xmu(mod(i,lxmu)+1), xstd(mod(i,lxstd)+1), prob(i+1));
end
```

9.2 Program Results

```
g01sa example results

tail    x       xmu      xstd    probability
 L     1.96     0.00     1.00    0.975
 U     1.96     0.00     1.00    0.025
 C     1.96     0.00     1.00    0.950
 S     1.96     0.00     1.00    0.050
```