Module 20.1: nag_normal_dist

Probabilities and Deviate for a Normal Distribution

`nag_normal_dist` provides procedures for computing probabilities and the deviate for various parts of a Normal distribution.

Contents

Procedure: nag_normal_prob

1 Description

nag_normal_prob calculates lower tail, upper tail or two tail probability for a univariate Normal distribution. It allows an unstandardized normal variate to be specified.

2 Usage

```
USE nag_normal_dist
[value =] nag_normal_prob(tail, x  [, optional arguments])
```

The function result is a scalar of type real(kind=*wp*).

3 Arguments

3.1 Mandatory Arguments

tail — character(len=1), intent(in)

Input: the type of tail probability to be returned:

if tail = 'L' or 'l', the lower tail probability is returned;

if tail = 'U' or 'u', the upper tail probability is returned;

if tail = 'S' or 's', the two tail (significance level) probability is returned;

if tail = 'C' or 'c', the two tail (confidence interval) probability is returned.

Constraints: tail = 'L', 'l', 'U' or 'u', 'S', 's', 'C' or 'c'.

x — real(kind=*wp*), intent(in)

Input: the value of the normal variate.

Note: if both x_mean and x_std are not specified, x is assumed to be a standard normal variate.

3.2 Optional Arguments

Note. Optional arguments must be supplied by keyword, not by position. The order in which they are described below may differ from the order in which they occur in the argument list.

x_mean — real(kind=*wp*), intent(in), optional

Input: the mean of the normal variate, x.

Default: x_mean = 0.0.

x_std — real(kind=*wp*), intent(in), optional

Input: the standard deviation of the normal variate, x.

Default: x_std = 1.0.

Constraints: x_std > 0.0.

error — type(nag_error), intent(inout), optional

The NAG *fl*90 error-handling argument. See the Essential Introduction, or the module document nag_error_handling (1.2). You are recommended to omit this argument if you are unsure how to use it. If this argument is supplied, it *must* be initialized by a call to nag_set_error before this procedure is called.

4 Error Codes

Fatal errors (error%level = 3):

error%code	Description
301	An input argument has an invalid value.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 1 of this module document.

6 Further Comments

6.1 Mathematical Background

Given the variate x with mean μ and standard deviation σ, the (lower tail, upper tail, and two tail) probabilities for various parts of a Normal distribution are calculated by first converting x to $z = (x - \mu)/\sigma$ and then solving the following.

For lower tail probability:

$$P(Z \leq z) = \int_{-\infty}^{z} f(Z)\, dZ.$$

For upper tail probability:

$$P(Z \geq z) = P(Z \leq -z).$$

For two tail significance level probability:

$$\mathrm{Prob} = P(Z \geq |z|) + P(Z \leq -|z|).$$

For two tail confidence interval probability:

$$\mathrm{Prob} = P(Z \leq |z|) - P(Z \leq -|z|),$$

where the probability density function (PDF) of z is defined by

$$f(Z) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-Z^2/2}.$$

6.2 Accuracy

The procedure uses the complementary error function erfc:

$$P(Z \leq z) = \frac{1}{2} \operatorname{erfc}\left(\frac{-z}{\sqrt{2}}\right).$$

Procedure: nag_normal_deviate

1 Description

nag_normal_deviate returns the deviate associated with the lower tail, upper tail or two tail probability of a standard Normal distribution.

2 Usage

```
USE nag_normal_dist
[value =] nag_normal_deviate(tail, p  [, optional arguments])
```

The function result is a scalar of type real(kind=*wp*).

3 Arguments

3.1 Mandatory Arguments

tail — character(len=1), intent(in)

Input: indicates which tail the supplied probability represents:

if tail = 'L' or 'l', p contains the lower tail probability;

if tail = 'U' or 'u', p contains the upper tail probability;

if tail = 'S' or 's', p contains the two tail (significance level) probability;

if tail = 'C' or 'c', p contains the two tail (confidence interval) probability.

Constraints: tail = 'L', 'l', 'U' or 'u', 'S', 's', 'C' or 'c'.

p — real(kind=*wp*), intent(in)

Input: the probability (as defined by tail) for the standard Normal distribution.

Constraints: $0.0 < \mathtt{p} < 1.0$.

3.2 Optional Argument

error — type(nag_error), intent(inout), optional

The NAG *fl*90 error-handling argument. See the Essential Introduction, or the module document nag_error_handling (1.2). You are recommended to omit this argument if you are unsure how to use it. If this argument is supplied, it *must* be initialized by a call to nag_set_error before this procedure is called.

4 Error Codes

Fatal errors (error%level = 3):

error%code	Description
301	An input argument has an invalid value.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 1 of this module document.

6 Further Comments

6.1 Mathematical Background

Given a specific probability from a standard Normal distribution, the deviate x_p associated with the lower tail probability p is defined as the solution to

$$P(X \leq x_p) = p = \int_{-\infty}^{x_p} f(Z)\, dZ$$

where

$$f(Z) = \frac{1}{\sqrt{2\pi}} \mathrm{e}^{-Z^2/2}, \;\; -\infty < X < \infty.$$

6.2 Algorithmic Detail

The method used is an extension of that of Beasley and Springer [2]. p is first replaced by $q = p - 0.5$.

(a) If $|q| \leq 0.3$, x_p is computed by a rational Chebyshev approximation

$$x_p = s \frac{A(s^2)}{B(s^2)}$$

where $s = \sqrt{2\pi}.q$ and A, B are polynomials of degree 7.

(b) If $0.3 < |q| \leq 0.42$, x_p is computed by a rational Chebyshev approximation

$$x_p = \mathrm{sign} q \left(\frac{C(t)}{D(t)} \right)$$

where $t = |q| - 0.3$ and C, D are polynomials of degree 5.

(c) If $|q| > 0.42$, x_p is computed as

$$x_p = \mathrm{sign} q \left[\left(\frac{E(u)}{F(u)} \right) + u \right]$$

where $u = \sqrt{-2 \log(\min(p, 1-p))}$ and E, F are polynomials of degree 6.

For the upper tail probability $-x_p$ is returned while for the two tail probabilities the value x_{p^*} is returned where p^* is the required tail probability computed from the input value of p.

6.3 Accuracy

The accuracy is limited by EPSILON(1.0_*wp*).

Procedure: nag_bivar_normal_prob

1 Description

nag_bivar_normal_prob returns the lower tail probability for a bivariate Normal distribution. It allows the means and standard deviations of the variables to be specified.

2 Usage

```
USE nag_normal_dist
[value =] nag_bivar_normal_prob(x, y, rho  [, optional arguments])
```

The function result is a scalar of type real(kind=*wp*).

3 Arguments

3.1 Mandatory Arguments

x — real(kind=*wp*), intent(in)
y — real(kind=*wp*), intent(in)

Input: the values of x and y for which the bivariate Normal distribution function is to be evaluated.

rho — real(kind=*wp*), intent(in)

Input: the correlation coefficient between x and y.

Constraints: $-1.0 \leq$ rho ≤ 1.0.

3.2 Optional Arguments

Note. Optional arguments must be supplied by keyword, not by position. The order in which they are described below may differ from the order in which they occur in the argument list.

x_mean — real(kind=*wp*), intent(in), optional
y_mean — real(kind=*wp*), intent(in), optional

Input: the means of x and y respectively.

Default: x_mean = 0.0, y_mean = 0.0.

x_std — real(kind=*wp*), intent(in), optional
y_std — real(kind=*wp*), intent(in), optional

Input: the standard deviations of x and y respectively.

Default: x_std = 1.0, y_std = 1.0.

Constraints: x_std > 0.0, y_std > 0.0.

error — type(nag_error), intent(inout), optional

The NAG *fl*90 error-handling argument. See the Essential Introduction, or the module document nag_error_handling (1.2). You are recommended to omit this argument if you are unsure how to use it. If this argument is supplied, it *must* be initialized by a call to nag_set_error before this procedure is called.

4 Error Codes

Fatal errors (error%level = 3):

error%code	Description
301	An input argument has an invalid value.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 2 of this module document.

6 Further Comments

6.1 Mathematical Background

Let the random variables (X and Y) follow a bivariate Normal distribution with

$$E[X] = \mu_1, \quad E[Y] = \mu_2, \quad E(X-\mu_1)^2 = \sigma_1^2, \quad E(Y-\mu_2)^2 = \sigma_2^2,$$

$$E((X-\mu_1)(Y-\mu_2)) = \sigma_{12}, \quad \text{and} \quad \rho = \sigma_{12}/(\sigma_1\sigma_2).$$

Suppose we write

$$u = \frac{x-\mu_1}{\sigma_1} \quad \text{and} \quad v = \frac{y-\mu_2}{\sigma_2},$$

it follows that the lower tail probability is defined by

$$P(U \le u, V \le v : \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{v} \int_{-\infty}^{u} \exp\left[-\frac{(U^2 - 2\rho UV + V^2)}{2(1-\rho^2)}\right] dU dV.$$

For a more detailed description of the bivariate Normal distribution and its properties see Abramowitz and Stegun [1] and Kendall and Stuart [4].

6.2 Algorithmic Detail

The method used here is described by Divgi [3].

6.3 Accuracy

The accuracy is discussed in Divgi [3] but this procedure uses a higher-order polynomial approximation to Mills ratio and this gives higher absolute accuracy of about 10 digits on machines of sufficiently high precision.

Procedure: nag_mv_normal_prob

1 Description

nag_mv_normal_prob calculates the lower tail, upper tail or central probability associated with a multivariate Normal distribution of up to ten dimensions. It allows the means and standard deviations of the variables to be specified.

2 Usage

```
USE nag_normal_dist
[value =] nag_mv_normal_prob(tail, a, correl  [, optional arguments])
```

The function result is a scalar of type real(kind=*wp*).

3 Arguments

Note. All array arguments are assumed-shape arrays. The extent in each dimension must be exactly that required by the problem. Notation such as '$\mathtt{x}(n)$' is used in the argument descriptions to specify that the array x must have exactly n elements.

This procedure derives the value of the following problem parameter from the shape of the supplied arrays.

$1 \leq n \leq 10$ — the number of multivariable distributions

3.1 Mandatory Arguments

tail — character(len=1), intent(in)

Input: the type of tail probability to be returned:

if tail = 'L' or 'l', the lower tail probability is returned;

if tail = 'U' or 'u', the upper tail probability is returned;

if tail = 'C' or 'c', the central probability is returned.

Note: if tail = 'C', then the optional argument b must be supplied.

Constraints: tail = 'L', 'l', 'U' or 'u', 'C' or 'c'.

a(n) — real(kind=*wp*), intent(in)

Input: the vector for which the multivariate Normal distribution function is to be evaluated:

if tail = 'U' or 'C', the vector of lower bounds;

if tail = 'L', the vector of upper bounds.

correl(n, n) — real(kind=*wp*), intent(in)

Input: the correlation matrix of the multivariate normal disribution. Only the lower triangle needs to be supplied. The diagonal elements are ignored and assumed to be unity.

Constraints: correl must be positive-definite.

3.2 Optional Arguments

Note. Optional arguments must be supplied by keyword, not by position. The order in which they are described below may differ from the order in which they occur in the argument list.

b(n) — real(kind=*wp*), intent(in), optional

Input: the vector of upper bounds, if the central probability is returned.

Constraints:

if `tail` = `'C'`, then `b` must be supplied and `b`(i) > `a`(i), for $i = 1, \ldots, n$;

otherwise `b` must not be supplied.

mean(n) — real(kind=*wp*), intent(in), optional

Input: the mean vector of the multivariate Normal distribution.

Default: `mean` = 0.0.

std(n) — real(kind=*wp*), intent(in), optional

Input: the standard deviation vector of the multivariate Normal distribution.

Default: `std` = 1.0.

Constraints: `std` > 0.0.

tol — real(kind=*wp*), intent(in), optional

Input: the relative accuracy required for the probability and if the upper or lower tail probability is requested, then it is also used to determine the cut-off-points.

Constraints:

if $n = 1$, `tol` is not referenced;

if $n > 1$, `tol` > 0.0.

Default: `tol` = 0.0001.

error — type(nag_error), intent(inout), optional

The NAG *fl*90 error-handling argument. See the Essential Introduction, or the module document `nag_error_handling` (1.2). You are recommended to omit this argument if you are unsure how to use it. If this argument is supplied, it *must* be initialized by a call to `nag_set_error` before this procedure is called.

4 Error Codes

Fatal errors (error%level = 3):

error%code	Description
301	An input argument has an invalid value.
302	An array argument has an invalid shape.
303	Array arguments have inconsistent shapes.
304	Invalid presence of an optional argument.
305	Invalid absence of an optional argument.
320	The procedure was unable to allocate enough memory.

Failures (error%level = 2):

error%code	Description
201	Matrix `correl` is not positive definite i.e., it is not a correct correlation matrix. The matrix `correl` must be positive definite.

Warnings (error%level = 1):

error%code	**Description**
101	Unable to achieve the required accuracy.
	A larger value for `tol` should be tried. The returned value may be an approximation to the required result.

5 Examples of Usage

A complete example of the use of this procedure appears in Example 3 of this module document.

6 Further Comments

6.1 Mathematical Background

Let the vector random variable $X = (X_1, X_2, \ldots, X_n)^T$ follow an n-dimensional multivariate Normal distribution with mean vector μ and n by n variance-covariance matrix $\Sigma = \sigma C \sigma$, where *sigma* is the matrix of standard deviation and C is the n by n correlation matrix. The probability density function, $f(X : \mu, \Sigma)$, is given by

$$f(X : \mu, \Sigma) = (2\pi)^{-(1/2)n} |\Sigma|^{-1/2} \exp\left(-\tfrac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu)\right).$$

The lower tail probability is defined by

$$P(X_1 \le a_1, \ldots, X_n \le a_n : \mu, \Sigma) = \int_{-\infty}^{a_1} \cdots \int_{-\infty}^{a_n} f(X : \mu, \Sigma) dX_n \ldots dX_1.$$

The upper tail probability is defined by

$$P(X_1 \ge a_1, \ldots, X_n \ge a_n : \mu, \Sigma) = \int_{a_1}^{\infty} \cdots \int_{a_n}^{\infty} f(X : \mu, \Sigma) dX_n \ldots dX_1.$$

The central probability is defined by

$$P(a_1 \le X_1 \le b_1, \ldots, a_n \le X_n \le b_n : \mu, \Sigma) = \int_{a_1}^{b_1} \cdots \int_{a_n}^{b_n} f(X : \mu, \Sigma) dX_n \ldots dX_1.$$

To evaluate the probability for $n \ge 3$, the probability density function of $X_1, X_2, \ldots, X_n$ is considered as the product of the conditional probability of $X_1, X_2, \ldots, X_{n-2}$ given X_{n-1} and X_n and the marginal bivariate Normal distribution of X_{n-1} and X_n. The bivariate normal probability can be evaluated as described in `nag_bivar_normal_prob` and numerical integration is then used over the remaining $n-2$ dimensions. In the case of $n = 3$ the procedure `nag_quad_1d_gen` is used and for $n > 3$ the procedure `nag_quad_md_rect` is used.
To evaluate the probability for $n = 1$ a direct call to `nag_normal_prob` is made and for $n = 2$ calls to `nag_bivar_normal_prob` are made.

6.2 Accuracy

For $n > 1$, the accuracy is specified by `tol`. When on exit `error%code` = 101 the approximate accuracy achieved is given in the error message. For the upper and lower tail probabilities the infinite limits are approximated by cut-off points for the $n-2$ dimensions over which the numerical integration takes place; these cut-off points are given by $\Phi^{-1}(\texttt{tol}/(10\times\ n))$, where Φ^{-1} is the inverse univariate Normal distribution function.

Example 1: Calculation of Probabilities and the Deviate for a Standard Normal Distribution

This example program shows how `nag_normal_prob` returns various probabilities and the deviate for a standard Normal distribution.

1 Program Text

Note. The listing of the example program presented below is double precision. Single precision users are referred to Section 5.2 of the Essential Introduction for further information.

```
PROGRAM nag_normal_dist_ex01

  ! Example Program Text for nag_normal_dist
  ! NAG fl90, Release 4. NAG Copyright 2000.

  ! .. Use Statements ..
  USE nag_examples_io, ONLY : nag_std_out
  USE nag_normal_dist, ONLY : nag_normal_prob, nag_normal_deviate
  ! .. Implicit None Statement ..
  IMPLICIT NONE
  ! .. Intrinsic Functions ..
  INTRINSIC KIND
  ! .. Parameters ..
  INTEGER, PARAMETER :: wp = KIND(1.0D0)
  ! .. Local Scalars ..
  INTEGER :: i
  REAL (wp) :: dev, prob, x
  ! .. Local Arrays ..
  CHARACTER (1) :: tail(4)
  ! .. Executable Statements ..
  WRITE (nag_std_out,*) 'Example Program Results for nag_normal_dist_ex01'

  WRITE (nag_std_out,*)
  WRITE (nag_std_out,*) ' Tail    X     Probability     Deviate'
  WRITE (nag_std_out,*)

  x = 1.96_wp
  tail = (/ 'L', 'U', 'C', 'S'/)

  DO i = 1, 4

    prob = nag_normal_prob(tail(i),x)
    dev = nag_normal_deviate(tail(i),prob)

    WRITE (nag_std_out,'(3X,A1,F8.2,2F13.4)') tail(i), x, prob, dev
  END DO

END PROGRAM nag_normal_dist_ex01
```

2 Program Data

None.

3 Program Results

```
 Example Program Results for nag_normal_dist_ex01

  Tail    X     Probability     Deviate

   L    1.96       0.9750       1.9600
   U    1.96       0.0250       1.9600
   C    1.96       0.9500       1.9600
   S    1.96       0.0500       1.9600
```

Example 2: Calculation of the Lower Tail Probability for a Bivariate Normal Distribution

This example program shows how `nag_bivar_normal_prob` returns the lower tail probability for a bivariate Normal distribution.

1 Program Text

Note. The listing of the example program presented below is double precision. Single precision users are referred to Section 5.2 of the Essential Introduction for further information.

```
    PROGRAM nag_normal_dist_ex02

      ! Example Program Text for nag_normal_dist
      ! NAG fl90, Release 4. NAG Copyright 2000.

      ! .. Use Statements ..
      USE nag_normal_dist, ONLY : nag_bivar_normal_prob
      USE nag_examples_io, ONLY : nag_std_out, nag_std_in
      ! .. Implicit None Statement ..
      IMPLICIT NONE
      ! .. Intrinsic Functions ..
      INTRINSIC KIND
      ! .. Parameters ..
      INTEGER, PARAMETER :: wp = KIND(1.0D0)
      ! .. Local Scalars ..
      REAL (wp) :: prob, rho, x, y
      ! .. Executable Statements ..
      WRITE (nag_std_out,*) 'Example Program Results for nag_normal_dist_ex02'

      READ (nag_std_in,*)          ! Skip heading in data file

      WRITE (nag_std_out,*)
      WRITE (nag_std_out,*) '         X          Y           RHO      PROB'
      WRITE (nag_std_out,*)

      DO
        READ (nag_std_in,*,end=20) x, y, rho

        prob = nag_bivar_normal_prob(x,y,rho)

        WRITE (nag_std_out,'(1X,3F12.3,F10.4)') x, y, rho, prob
      END DO
20    CONTINUE

    END PROGRAM nag_normal_dist_ex02
```

2 Program Data

```
Example Program Data for nag_normal_dist_ex02
  1.7  23.1   0.0         :x, y, rho
  0.0   0.0   0.1
  3.3  11.1   0.54
  9.1   9.1   0.17
```

3 Program Results

```
 Example Program Results for nag_normal_dist_ex02

          X          Y            RHO       PROB

        1.700      23.100        0.000     0.9554
        0.000       0.000        0.100     0.2659
        3.300      11.100        0.540     0.9995
        9.100       9.100        0.170     1.0000
```

Example 3: Calculation of the Central Probability for a Multivariate Normal Distribution

This example program shows how `nag_mv_normal_prob` returns the central probability for a multivariate Normal distribution.

1 Program Text

Note. The listing of the example program presented below is double precision. Single precision users are referred to Section 5.2 of the Essential Introduction for further information.

```
PROGRAM nag_normal_dist_ex03

  ! Example Program Text for nag_normal_dist
  ! NAG fl90, Release 4. NAG Copyright 2000.

  ! .. Use Statements ..
  USE nag_normal_dist, ONLY : nag_mv_normal_prob
  USE nag_examples_io, ONLY : nag_std_out, nag_std_in
  ! .. Implicit None Statement ..
  IMPLICIT NONE
  ! .. Intrinsic Functions ..
  INTRINSIC ALLOCATED, KIND
  ! .. Parameters ..
  INTEGER, PARAMETER :: wp = KIND(1.0D0)
  ! .. Local Scalars ..
  INTEGER :: n
  REAL (wp) :: prob
  CHARACTER (1) :: tail
  ! .. Local Arrays ..
  REAL (wp), ALLOCATABLE :: a(:), b(:), correl(:,:)
  ! .. Executable Statements ..
  WRITE (nag_std_out,*) 'Example Program Results for nag_normal_dist_ex03'

  READ (nag_std_in,*)          ! Skip heading in data file

  WRITE (nag_std_out,*)

  READ (nag_std_in,*) n, tail

  IF (tail=='C' .OR. tail=='c') THEN
    ALLOCATE (a(n),correl(n,n),b(n))
  ELSE
    ALLOCATE (a(n),correl(n,n))
  END IF

  READ (nag_std_in,*) correl
  READ (nag_std_in,*) a

  IF (tail=='C' .OR. tail=='c') THEN
    READ (nag_std_in,*) b
    prob = nag_mv_normal_prob(tail,a,correl,b=b)
  ELSE
    prob = nag_mv_normal_prob(tail,a,correl)
  END IF
  WRITE (nag_std_out,'(1X,A,F7.4)') 'multivariate probability = ', prob
  DEALLOCATE (a,correl)
  IF (ALLOCATED(b)) DEALLOCATE (b)

END PROGRAM nag_normal_dist_ex03
```

2 Program Data

```
Example Program Data for nag_normal_dist_ex03
 4     'c'              : n,tail
 1.0   0.9   0.9   0.9
 0.9   1.0   0.9   0.9
 0.9   0.9   1.0   0.9
 0.9   0.9   0.9   1.0  : correl(1:n,1:n)
-2.0  -2.0  -2.0  -2.0  : a(1:n)
 2.0   2.0   2.0   2.0  : b(1:n)
```

3 Program Results

```
 Example Program Results for nag_normal_dist_ex03

 multivariate probability =  0.9142
```

Additional Examples

Not all example programs supplied with NAG *fl*90 appear in full in this module document. The following additional examples, associated with this module, are available.

`nag_normal_dist_ex04`

Calculation of the deviate associated with a given probability for a standard Normal distribution.

References

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

[2] Beasley J D and Springer S G (1977) Algorithm AS111. The percentage points of the Normal distribution *Appl. Statist.* **26** 118–120

[3] Divgi D R (1979) Calculation of univariate and bivariate normal probability functions *Ann. Statist.* **7** 903–910

[4] Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* Griffin (3rd Edition)