NAG Library Function Document

nag_bivariate_normal_dist (g01hac)

1 Purpose

nag_bivariate_normal_dist (g01hac) returns the lower tail probability for the bivariate Normal distribution.

2 Specification

```
#include &lt;nag.h&gt;
#include &lt;nagg01.h&gt;
double nag_bivariate_normal_dist (double x, double y, double rho,
       NagError *fail)
```

3 Description

For the two random variables (X,Y) following a bivariate Normal distribution with

$$E[X]=0, \quad E[Y]=0, \quad E\big[X^2\big]=1, \quad E\big[Y^2\big]=1 \quad \text{and} \quad E[XY]=\rho,$$

the lower tail probability is defined by:

$$P(X \le x, Y \le y : \rho) = \frac{1}{2\pi\sqrt{1-\rho^2}} \int_{-\infty}^{y} \int_{-\infty}^{x} \exp\left(-\frac{\left(X^2-2\rho XY+Y^2\right)}{2\left(1-\rho^2\right)}\right) dX dY.$$

For a more detailed description of the bivariate Normal distribution and its properties see Abramowitz and Stegun (1972) and Kendall and Stuart (1969). The method used is described by Genz (2004).

4 References

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

Genz A (2004) Numerical computation of rectangular bivariate and trivariate Normal and t probabilities *Statistics and Computing* **14** 151–160

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

5 Arguments

1: **x** – double *Input*

On entry: x, the first argument for which the bivariate Normal distribution function is to be evaluated.

2: **y** – double *Input*

On entry: y, the second argument for which the bivariate Normal distribution function is to be evaluated.

3: **rho** – double *Input*

On entry: ρ, the correlation coefficient.

Constraint: $-1.0 \le$ **rho** ≤ 1.0.

4: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

On any of the error conditions listed below nag_bivariate_normal_dist (g01hac) returns 0.0.

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_REAL_ARG_GT

On entry, **rho** $= \langle value \rangle$.
Constraint: **rho** ≤ 1.0.

NE_REAL_ARG_LT

On entry, **rho** $= \langle value \rangle$.
Constraint: **rho** ≥ -1.0.

7 Accuracy

Accuracy of the hybrid algorithm implemented here is discussed in Genz (2004). This algorithm should give a maximum absolute error of less than 5×10^{-16}.

8 Parallelism and Performance

Not applicable.

9 Further Comments

The probabilities for the univariate Normal distribution can be computed using nag_cumul_normal (s15abc) and nag_cumul_normal_complem (s15acc).

10 Example

This example reads values of x and y for a bivariate Normal distribution along with the value of ρ and computes the lower tail probabilities.

10.1 Program Text

```
/* nag_bivariate_normal_dist (g01hac) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg01.h>

int main(void)
{
  Integer  exit_status = 0;
  double   prob, rho, x, y;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  printf(
          "nag_bivariate_normal_dist (g01hac) Example Program Results\n");
  printf("    x       y      rho    prob\n\n");
#ifdef _WIN32
  while (scanf_s("%lf %lf %lf", &x, &y, &rho) != EOF)
#else
  while (scanf("%lf %lf %lf", &x, &y, &rho) != EOF)
#endif
    {
      /* nag_bivariate_normal_dist (g01hac).
       * Probability for the bivariate Normal distribution
       */
      prob = nag_bivariate_normal_dist(x, y, rho, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf(
                  "Error from nag_bivariate_normal_dist (g01hac).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%8.3f%8.3f%8.3f%8.4f\n", x, y, rho, prob);
    }

 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_bivariate_normal_dist (g01hac) Example Program Data
  1.7  23.1   0.0
  0.0   0.0   0.1
  3.3  11.1   0.54
  9.1   9.1   0.17
```

10.3 Program Results

```
nag_bivariate_normal_dist (g01hac) Example Program Results
    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
```