NAG Library Function Document

nag_deviates_gamma_dist (g01ffc)

1 Purpose

nag_deviates_gamma_dist (g01ffc) returns the deviate associated with the given lower tail probability of the gamma distribution.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
double nag_deviates_gamma_dist (double p, double a, double b, double tol,
       NagError *fail)
```

3 Description

The deviate, g_p, associated with the lower tail probability, p, of the gamma distribution with shape parameter α and scale parameter β, is defined as the solution to

$$P\big(G \leq g_p : \alpha, \beta\big) = p = \frac{1}{\beta^{\alpha}\Gamma(\alpha)}\int_0^{g_p} e^{-G/\beta}G^{\alpha-1}\,dG, \quad 0 \leq g_p < \infty; \alpha, \beta > 0.$$

The method used is described by Best and Roberts (1975) making use of the relationship between the gamma distribution and the χ^2-distribution.

Let $y = 2\dfrac{g_p}{\beta}$. The required y is found from the Taylor series expansion

$$y = y_0 + \sum_r \frac{C_r(y_0)}{r!}\left(\frac{E}{\phi(y_0)}\right)^r,$$

where y_0 is a starting approximation

$$C_1(u) = 1,$$

$$C_{r+1}(u) = \left(r\Psi + \frac{d}{du}\right)C_r(u),$$

$$\Psi = \tfrac{1}{2} - \frac{\alpha - 1}{u},$$

$$E = p - \int_0^{y_0} \phi(u)\,du,$$

$$\phi(u) = \frac{1}{2^{\alpha}\Gamma(\alpha)}e^{-u/2}u^{\alpha-1}.$$

For most values of p and α the starting value

$$y_{01} = 2\alpha\left(z\sqrt{\frac{1}{9\alpha}} + 1 - \frac{1}{9\alpha}\right)^3$$

is used, where z is the deviate associated with a lower tail probability of p for the standard Normal distribution.

For p close to zero,

$$y_{02} = (p\alpha 2^{\alpha}\Gamma(\alpha))^{1/\alpha}$$

is used.

For large p values, when $y_{01} > 4.4\alpha + 6.0$,

$$y_{03} = -2\left[\ln(1-p) - (\alpha-1)\ln\left(\tfrac{1}{2}y_{01}\right) + \ln(\Gamma(\alpha))\right]$$

is found to be a better starting value than y_{01}.

For small α ($\alpha \leq 0.16$), p is expressed in terms of an approximation to the exponential integral and y_{04} is found by Newton–Raphson iterations.

Seven terms of the Taylor series are used to refine the starting approximation, repeating the process if necessary until the required accuracy is obtained.

4 References

Best D J and Roberts D E (1975) Algorithm AS 91. The percentage points of the χ^2 distribution *Appl. Statist.* **24** 385–388

5 Arguments

1: **p** – double *Input*

On entry: p, the lower tail probability from the required gamma distribution.

Constraint: $0.0 \leq \mathbf{p} < 1.0$.

2: **a** – double *Input*

On entry: α, the shape parameter of the gamma distribution.

Constraint: $0.0 < \mathbf{a} \leq 10^6$.

3: **b** – double *Input*

On entry: β, the scale parameter of the gamma distribution.

Constraint: $\mathbf{b} > 0.0$.

4: **tol** – double *Input*

On entry: the relative accuracy required by you in the results. The smallest recommended value is $50 \times \delta$, where $\delta = \max\left(10^{-18}, \textbf{\textit{machine precision}}\right)$. If nag_deviates_gamma_dist (g01ffc) is entered with **tol** less than $50 \times \delta$ or greater or equal to 1.0, then $50 \times \delta$ is used instead.

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

The NAG error argument (see Section 2.7 in How to Use the NAG Library and its Documentation).

6 Error Indicators and Warnings

On any of the error conditions listed below, except **fail.code** = NE_ALG_NOT_CONV, nag_deviates_gamma_dist (g01ffc) returns 0.0.

NE_ALG_NOT_CONV

The algorithm has failed to converge in 100 iterations. A larger value of **tol** should be tried. The result may be a reasonable approximation.

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 2.3.1.2 in How to Use the NAG Library and its Documentation for further information.

NE_GAM_NOT_CONV

The series used to calculate the gamma function has failed to converge. This is an unlikely error exit.

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 2.7.6 in How to Use the NAG Library and its Documentation for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 2.7.5 in How to Use the NAG Library and its Documentation for further information.

NE_PROBAB_CLOSE_TO_TAIL

The probability is too close to 0.0 for the given **a** to enable the result to be calculated.

NE_REAL_ARG_GE

On entry, $\mathbf{p} = \langle value \rangle$.
Constraint: $\mathbf{p} < 1.0$.

NE_REAL_ARG_GT

On entry, $\mathbf{a} = \langle value \rangle$.
Constraint: $\mathbf{a} \leq 10^6$.

NE_REAL_ARG_LE

On entry, $\mathbf{a} = \langle value \rangle$.
Constraint: $\mathbf{a} > 0.0$.

On entry, $\mathbf{b} = \langle value \rangle$.
Constraint: $\mathbf{b} > 0.0$.

NE_REAL_ARG_LT

On entry, $\mathbf{p} = \langle value \rangle$.
Constraint: $\mathbf{p} \geq 0.0$.

7 Accuracy

In most cases the relative accuracy of the results should be as specified by **tol**. However, for very small values of α or very small values of p there may be some loss of accuracy.

8 Parallelism and Performance

nag_deviates_gamma_dist (g01ffc) is not threaded in any implementation.

9 Further Comments

None.

10 Example

This example reads lower tail probabilities for several gamma distributions, and calculates and prints the corresponding deviates until the end of data is reached.

10.1 Program Text

```
/* nag_deviates_gamma_dist (g01ffc) Example Program.
 *
 * NAGPRODCODE Version.
 *
 * Copyright 2016 Numerical Algorithms Group.
 *
 * Mark 26, 2016.
 */

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

int main(void)
{
  Integer exit_status = 0;
  double a, b, p, g;
  double tol = 0.0;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  printf("nag_deviates_gamma_dist (g01ffc) Example Program Results\n");
  printf("    p       a       b           g\n\n");
#ifdef _WIN32
  while (scanf_s("%lf %lf %lf", &p, &a, &b) != EOF)
#else
  while (scanf("%lf %lf %lf", &p, &a, &b) != EOF)
#endif
  {
    /* nag_deviates_gamma_dist (g01ffc).
     * Deviates for the gamma distribution
     */
    g = nag_deviates_gamma_dist(p, a, b, tol, &fail);
    if (fail.code != NE_NOERROR) {
      printf("Error from nag_deviates_gamma_dist (g01ffc).\n%s\n",
             fail.message);
      exit_status = 1;
      goto END;
    }
    printf("%8.3f%8.3f%8.3f%10.3f\n", p, a, b, g);
  }

END:
  return exit_status;
}
```

10.2 Program Data

```
nag_deviates_gamma_dist (g01ffc) Example Program Data
 0.0100   1.0 20.0
 0.4279   7.5  0.1
 0.8694  45.0 10.0
```

10.3 Program Results

```
nag_deviates_gamma_dist (g01ffc) Example Program Results
    p       a       b           g

   0.010   1.000  20.000     0.201
   0.428   7.500   0.100     0.670
   0.869  45.000  10.000   525.979
```