NAG Library Function Document

nag_prob_beta_dist (g01eec)

1 Purpose

nag_prob_beta_dist (g01eec) computes the upper and lower tail probabilities and the probability density function of the beta distribution with parameters a and b.

2 Specification

```
#include <nag.h>
#include <nagg01.h>
void nag_prob_beta_dist (double x, double a, double b, double tol, double *p,
     double *q, double *pdf, NagError *fail)
```

3 Description

The probability density function of the beta distribution with parameters a and b is:

$$f(B : a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} B^{a-1}(1-B)^{b-1}, \quad 0 \leq B \leq 1; a, b > 0.$$

The lower tail probability, $P(B \leq \beta : a, b)$ is defined by

$$P(B \leq \beta : a, b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \int_0^\beta B^{a-1}(1-B)^{b-1}\, dB = I_\beta(a, b), \quad 0 \leq \beta \leq 1; a, b > 0.$$

The function $I_x(a, b)$, also known as the incomplete beta function is calculated using nag_incomplete_beta (s14ccc).

4 References

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

5 Arguments

1: **x** – double *Input*

On entry: β, the value of the beta variate.

Constraint: $0.0 \leq \mathbf{x} \leq 1.0$.

2: **a** – double *Input*

On entry: a, the first parameter of the required beta distribution.

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

3: **b** – double *Input*

On entry: b, the second parameter of the required beta distribution.

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

4: **tol** – double *Input*

On entry: this argument is no longer referenced, but is included for backwards compatability.

5: **p** – double * *Output*

On exit: the lower tail probability, $P(B \leq \beta : a, b)$.

6: **q** – double * *Output*

On exit: the upper tail probability, $P(B \geq \beta : a, b)$.

7: **pdf** – double * *Output*

On exit: the probability density function, $f(B : a, b)$.

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

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

6 Error Indicators and Warnings

NE_ALLOC_FAIL

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

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

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, $\mathbf{x} = \langle value \rangle$.
Constraint: $\mathbf{x} \leq 1.0$.

NE_REAL_ARG_LE

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

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

NE_REAL_ARG_LT

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

7 Accuracy

The accuracy is limited by the error in the incomplete beta function. See Section 7 in nag_incomplete_beta (s14ccc) for further details.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

This example reads values from a number of beta distributions and computes the associated upper and lower tail probabilities and the corresponding value of the probability density function.

10.1 Program Text

```
/* nag_prob_beta_dist (g01eec) 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   a, b, p, pdf, q, tol, x;
  NagError fail;

  INIT_FAIL(fail);

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n]");
#else
  scanf("%*[^\n]");
#endif
  printf("nag_prob_beta_dist (g01eec) Example Program Results\n");
  printf("   x         a           b            p            q"
         "           pdf\n\n");
#ifdef _WIN32
  while (scanf_s("%lf %lf %lf %lf", &x, &a, &b, &tol) != EOF)
#else
  while (scanf("%lf %lf %lf %lf", &x, &a, &b, &tol) != EOF)
#endif
    {
      /* nag_prob_beta_dist (g01eec).
       * Upper and lower tail probabilities and probability
       * density function for the beta distribution
       */
      nag_prob_beta_dist(x, a, b, tol, &p, &q, &pdf, &fail);
      if (fail.code != NE_NOERROR)
        {
          printf("Error from nag_prob_beta_dist (g01eec).\n%s\n",
                  fail.message);
          exit_status = 1;
          goto END;
        }
      printf("%7.4f%13.4e%13.4e%13.4e%13.4e%13.4e\n", x, a, b, p, q,
              pdf);
```

```
    }

 END:
  return exit_status;
}
```

10.2 Program Data

```
nag_prob_beta_dist (g01eec) Example Program Data
 0.25  1.0  2.0  1.9
 0.75  1.5  1.5  0.0001
 0.5   2.0  1.0  1.01
```

10.3 Program Results

```
nag_prob_beta_dist (g01eec) Example Program Results
   x         a            b            p            q          pdf

 0.2500   1.0000e+00   2.0000e+00   4.3750e-01   5.6250e-01   1.5000e+00
 0.7500   1.5000e+00   1.5000e+00   8.0450e-01   1.9550e-01   1.1027e+00
 0.5000   2.0000e+00   1.0000e+00   2.5000e-01   7.5000e-01   1.0000e+00
```