NAG Library Function Document

nag_deviates_beta (g01fec)

1 Purpose

nag_deviates_beta (g01fec) returns the deviate associated with the given lower tail probability of the beta distribution.

2 Specification

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

3 Description

The deviate, \( \beta_p \), associated with the lower tail probability, \( p \), of the beta distribution with parameters \( a \) and \( b \) is defined as the solution to

\[
P(B \leq \beta_p : a, b) = p = \frac{\Gamma(a + b)}{\Gamma(a)\Gamma(b)} \int_0^{\beta_p} B^{a-1}(1 - B)^{b-1} dB, \quad 0 \leq \beta_p \leq 1, a, b > 0.
\]

The algorithm is a modified version of the Newton–Raphson method, following closely that of Cran et al. (1977).

An initial approximation, \( \beta_0 \), to \( \beta_p \) is found (see Cran et al. (1977)), and the Newton–Raphson iteration

\[
\beta_i = \beta_{i-1} - \frac{f(\beta_{i-1})}{f'(\beta_{i-1})},
\]

where \( f(\beta) = P(B \leq \beta : a, b) - p \) is used, with modifications to ensure that \( \beta \) remains in the range \((0, 1)\).

4 References


5 Arguments

1: \( p \) – double

\textit{Input}

\textit{On entry:} \( p \), the lower tail probability from the required beta distribution.

\textit{Constraint:} \( 0.0 \leq p \leq 1.0 \).

2: \( a \) – double

\textit{Input}

\textit{On entry:} \( a \), the first parameter of the required beta distribution.

\textit{Constraint:} \( 0.0 < a \leq 10^6 \).
3: \( b \) – double

    \textit{Input}

    \textit{On entry:} \( b \), the second parameter of the required beta distribution.
    \textit{Constraint:} \( 0.0 < b \leq 10^6. \)

4: \( \text{tol} \) – double

    \textit{Input}

    \textit{On entry:} the relative accuracy required by you in the result. If \( \text{nag\_deviates\_beta} \) is entered with \( \text{tol} \) greater than or equal to 1.0 or less than \( 10 \times \text{machine precision} \) (see \( \text{nag\_machine\_precision} \)), then the value of \( 10 \times \text{machine precision} \) is used instead.

5: \( \text{fail} \) – NagError*

    \textit{Input/Output}

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

6 \textbf{Error Indicators and Warnings}

On any of the error conditions listed below except \texttt{fail.code = NE\_RES\_NOT\_ACC} or \texttt{NE\_SOL\_NOT\_CONV} \( \text{nag\_deviates\_beta} \) returns 0.0.

\textbf{NE\_ALLOC\_FAIL}

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

\textbf{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.

\textbf{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.

\textbf{NE\_REAL\_ARG\_GT}

    On entry, \( a = \langle \text{value} \rangle \) and \( b = \langle \text{value} \rangle \).
    Constraint: \( a \leq 10^6. \)
    On entry, \( a = \langle \text{value} \rangle \) and \( b = \langle \text{value} \rangle \).
    Constraint: \( b \leq 10^6. \)
    On entry, \( p = \langle \text{value} \rangle \).
    Constraint: \( p \leq 1.0. \)

\textbf{NE\_REAL\_ARG\_LE}

    On entry, \( a = \langle \text{value} \rangle \) and \( b = \langle \text{value} \rangle \).
    Constraint: \( a > 0.0. \)
    On entry, \( a = \langle \text{value} \rangle \) and \( b = \langle \text{value} \rangle \).
    Constraint: \( b > 0.0. \)

\textbf{NE\_REAL\_ARG\_LT}

    On entry, \( p = \langle \text{value} \rangle \).
    Constraint: \( p \geq 0.0. \)
The requested accuracy has not been achieved. Use a larger value of tol. There is doubt concerning the accuracy of the computed result. 100 iterations of the Newton–Raphson method have been performed without satisfying the accuracy criterion (see Section 9). The result should be a reasonable approximation of the solution.

The solution has failed to converge. However, the result should be a reasonable approximation. Requested accuracy not achieved when calculating beta probability. You should try setting tol larger.

7 Accuracy
The required precision, given by tol, should be achieved in most circumstances.

8 Parallelism and Performance
Not applicable.

9 Further Comments
The typical timing will be several times that of nag_prob_beta_dist (g01ec) and will be very dependent on the input argument values. See nag_prob_beta_dist (g01ec) for further comments on timings.

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

10.1 Program Text
/* nag_deviates_beta (g01fc) Example Program. *
 * Copyright 2014 Numerical Algorithms Group.
 * * Mark 2 revised, 1992.
 */
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg01.h>

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

    INIT_FAIL(fail);

    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
    printf("nag_deviates_beta (g01fc) Example Program Results\n");
    printf(" Probability A B Deviate\n");
#ifdef _WIN32
    while (scanf_s("\%f \%f \%f", &p, &a, &b) != EOF)
#else
while (scanf("%lf %lf %lf", &p, &a, &b) != EOF)
#else
{
  tol = 0.0;
  /* nag_deviates_beta (g01fec).
   * Deviates for the beta distribution
   */
  x = nag_deviates_beta(p, a, b, tol, &fail);
  if (fail.code != NE_NOERROR)
  {
    printf("Error from nag_deviates_beta (g01fec).\n%s\n", fail.message);
    exit_status = 1;
    if (fail.code != NE_RES_NOT_ACC && fail.code != NE_SOL_NOT_CONV)
    {
      goto END;
    }
  }
  printf("%9.4f%10.3f%10.3f%10.4f\n", p, a, b, x);
}
END:
return exit_status;
}

10.2 Program Data

nag_deviates_beta (g01fec) Example Program Data
0.5000 1.0 2.0
0.9900 1.5 1.5
0.2500 20.0 10.0

10.3 Program Results

nag_deviates_beta (g01fec) Example Program Results
Probability  A    B    Deviate

<table>
<thead>
<tr>
<th>Probability</th>
<th>A</th>
<th>B</th>
<th>Deviate</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5000</td>
<td>1.000</td>
<td>2.000</td>
<td>0.2929</td>
</tr>
<tr>
<td>0.9900</td>
<td>1.500</td>
<td>1.500</td>
<td>0.9672</td>
</tr>
<tr>
<td>0.2500</td>
<td>20.000</td>
<td>10.000</td>
<td>0.6105</td>
</tr>
</tbody>
</table>