nag_deviates_f_dist (g01fdc)

1. Purpose

nag_deviates_f_dist (g01fdc) returns the deviate associated with the given lower tail probability of the F or variance-ratio distribution with real degrees of freedom.

2. Specification

```
#include <nag.h>
#include <nagg01.h>

double nag_deviates_f_dist(double p, double df1, double df2, NagError *fail)
```

3. Description

The deviate, f_p, associated with the lower tail probability, p, of the F-distribution with degrees of freedom ν_1 and ν_2 is defined as the solution to

$$\begin{aligned} P(F \leq f_p : \nu_1, \nu_2) &= p \\ &= \frac{\nu_1^{\nu_1/2} \nu_1 \nu_2^{\nu_2/2} \nu_2 \Gamma((\nu_1+\nu_2)/2)}{\Gamma(\nu_1/2)\Gamma(\nu_2/2)} \int_0^{f_p} F^{(\nu_1-2)/2} (\nu_2 + \nu_1 F)^{(\nu_1+\nu_2)/2} \, dF \end{aligned}$$

where $\nu_1, \nu_2 > 0$; $0 \leq f_p < \infty$.

The value of f_p is computed by means of a transformation to a beta distribution, $P_\beta(B \leq \beta : a, b)$

$$P(F \leq f : \nu_1, \nu_2) = P_\beta \left(B \leq \frac{\nu_1 f}{\nu_1 f + \nu_2} : \nu_1/2, \nu_2/2 \right)$$

and using a call to nag_deviates_beta (g01fec).

For very large values of both ν_1 and ν_2, greater than 10^5, a normal approximation is used. If only one of ν_1 or ν_2 is greater than 10^5 then a χ^2 approximation is used, see Abramowitz and Stegun (1965).

4. Parameters

p

Input: the probability, p, from the required F-distribution.
Constraint: $0.0 \leq \mathbf{p} < 1.0$.

df1

Input: the degrees of freedom of the numerator variance, ν_1.
Constraint: $\mathbf{df1} > 0.0$.

df2

Input: the degrees of freedom of the denominator variance, ν_2.
Constraint: $\mathbf{df2} > 0.0$.

fail

The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

On any of the error conditions listed below except **NAG_SOL_NOT_CONV** nag_deviates_f_dist returns 0.0.

NE_REAL_ARG_LT

On entry, **p** must not be less than 0.0: $\mathbf{p} = \langle \mathit{value} \rangle$.

NE_REAL_ARG_GE

On entry, **p** must not be greater than or equal to 1.0: **p** = ⟨*value*⟩.

NE_REAL_ARG_LE

On entry, **df1** must not be less than or equal to 0.0: **df1** = ⟨*value*⟩.
On entry, **df2** must not be less than or equal to 0.0: **df2** = ⟨*value*⟩.

NE_SOL_NOT_CONV

The solution has failed to converge.
However, the result should be a reasonable approximation.
Alternatively, nag_deviates_f_dist can be used with a suitable setting of the parameter **tol**.

NE_PROBAB_CLOSE_TO_TAIL

The probability is too close to 0.0 or 1.0.
The value of f_p cannot be computed. This will only occur when the large sample approximations are used.

6. Further Comments

For higher accuracy nag_deviates_beta (g01fec) can be used along with the transformations given in Section 3.

6.1. Accuracy

The result should be accurate to 5 significant digits.

6.2. References

Abramowitz M and Stegun I A (1965) *Handbook of Mathematical Functions* Dover Publications, New York ch 26.

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

7. See Also

nag_deviates_beta (g01fec)

8. Example

Lower tail probabilities are read for several F-distributions, and the corresponding deviates calculated and printed, until the end of data is reached.

8.1. Program Text

```
/* nag_deviates_f_dist(g01fdc) Example Program
 *
 * Copyright 1990 Numerical Algorithms Group.
 *
 * Mark 1, 1990.
 */

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

main()
{
  double df1, df2, f, p;
  static NagError fail;

  /*  Skip heading in data file */
  Vscanf("%*[^\n]");
  Vprintf("g01fdc Example Program Results\n");
  Vprintf("     p      df1     df2      f\n\n");
  while (scanf("%lf %lf %lf", &p, &df1, &df2) != EOF)
    {
      f = g01fdc(p, df1, df2, &fail);
```

```
        if (fail.code==NE_NOERROR)
          Vprintf("%8.3f%8.3f%8.3f%8.3f\n", p, df1, df2, f);
        else
          Vprintf("%8.3f%8.3f%8.3f%8.3f\n Note: %s\n",p,df1,df2,f,
                  fail.message);
      }
    exit(EXIT_SUCCESS);
  }
```

8.2. Program Data

```
g01fdc Example Program Data
0.9837  10.0   25.5
0.9000  1.0    1.0
0.5342  20.25  1.0
```

8.3. Program Results

```
g01fdc Example Program Results
     p      df1     df2      f

   0.984  10.000  25.500   2.837
   0.900   1.000   1.000  39.866
   0.534  20.250   1.000   2.500
```