# nag_deviates_normal (g01fac)

### 1.   Purpose

**nag_deviates_normal (g01fac)** returns the deviate associated with the given probability of the standard Normal distribution.

### 2.   Specification

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

double nag_deviates_normal(Nag_TailProbability tail, double p, NagError *fail)
```

### 3.   Description

The deviate, $x_p$ associated with the lower tail probability, $p$, for the standard Normal distribution is defined as the solution to:

$$P(X \leq x_p) = p = \int_{-\infty}^{x_p} Z(X) dX$$

where

$$Z(X) = \frac{1}{\sqrt{2\pi}} e^{-X^2/2}, \quad -\infty < X < \infty.$$

The method used is an extension of that of Beasley and Springer (1977). $p$ is first replaced by $q = p - 0.5$.

(a)   if $|q| \leq 0.3$, $x_p$ is computed by a rational Chebyshev approximation

$$x_p = s \frac{A(s^2)}{B(s^2)}$$

where $s = \sqrt{2\pi}.q$ and $A$, $B$ are polynomials of degree 7.

(b)   if $0.3 < |q| \leq 0.42$, $x_p$ is computed by a rational Chebyshev approximation

$$x_p = \text{sign}\, q \left( \frac{C(t)}{D(t)} \right)$$

where $t = |q| - 0.3$ and $C$, $D$ are polynomials of degree 5.

(c)   if $|q| > 0.42$, $x_p$ is computed as

$$x_p = \text{sign}\, q \left\{ \left( \frac{E(u)}{F(u)} \right) + u \right\}$$

where $u = \sqrt{-2 \times \log(\min(p, 1-p))}$ and $E$, $F$ are polynomials of degree 6.

For the upper tail probability $-x_p$ is returned while for the two tail probabilities the value $x_{p^*}$ is returned where $p^*$ is the required tail probability computed from the input value of $p$.

## 4. Parameters

**tail**

Input: indicates which tail the supplied probability represents.
If **tail** = **Nag_LowerTail**, the lower tail probability, i.e., $P(X \leq x_p)$.
If **tail** = **Nag_UpperTail**, the upper tail probability, i.e., $P(X \geq x_p)$.
If **tail** = **Nag_TwoTailSignif**, the two tail (significance level) probability, i.e., $P(X \geq |x_p|) + P(X \leq -|x_p|)$.
If **tail** = **Nag_TwoTailConfid**, the two tail (confidence interval) probability, i.e., $P(X \leq |x_p|) - P(X \leq -|x_p|)$.
Constraint: **tail** = **Nag_UpperTail**, **Nag_LowerTail**, **Nag_TwoTailSignif** or **Nag_TwoTailConfid**.

**p**

Input: the probability, $p$, from the standard Normal distribution as defined by **tail**.
Constraint: $0.0 < \mathbf{p} < 1.0$.

**fail**

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

## 5. Error Indications and Warnings

If **fail.code** $\neq$ **NE_NOERROR**, then nag_deviates_normal returns 0.0.

**NE_BAD_PARAM**

On entry, parameter **tail** had an illegal value.

**NE_REAL_ARG_LE**

On entry, **p** must not be less than or equal to 0.0: **p** = $\langle value \rangle$.

**NE_REAL_ARG_GE**

On entry, **p** must not be greater than or equal to 1.0: **p** = $\langle value \rangle$.

**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 consult NAG for assistance.

## 6. Further Comments

### 6.1. Accuracy

Accuracy is mainly limited by the **machine precision**.

### 6.2. References

Abramowitz M and Stegun I A (1972) *Handbook of Mathematical Functions* Ch. 7.1, p.297 and Ch. 26.2, p. 931 Dover Publications, New York.
Beasley J D and Springer S G (1977) Algorithm AS111. The Percentage Points of the Normal Distribution *Appl. Statist.* **26** 118–120.
Hastings N A J and Peacock J B (1977) *Statistical Distributions* Ch. 21, pp.96–101 Butterworth.

## 7. See Also

None

## 8. Example

Four values of **tail** and **x** are input and the probabilities calculated and printed.

### 8.1. Program Text

```
/* nag_deviates_normal(g01fac) Example Program.
 *
 * Copyright 1996 Numerical Algorithms Group.
 *
 * Mark 4, 1996.
 *
```

```
 */

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

main()
{

  double p;
  double dev;
  Integer i;
  char tail_char;
  Nag_TailProbability tail;

  Vprintf(" g01fac Example Program Results\n");
  /*     Skip heading in data file */
  Vscanf("%*[^\n] ");
  Vprintf("\n  Tail    Probability     Deviate \n\n");
  for (i = 1; i <= 4; ++i)
    {
      Vscanf("%c %lf ", &tail_char, &p);
      switch (tail_char)
        {
        case 'L':
          tail=Nag_LowerTail;
          break;
        case 'U':
          tail=Nag_UpperTail;
          break;
        case 'C':
          tail=Nag_TwoTailConfid;
          break;
        case 'S':
          tail=Nag_TwoTailSignif;
        }

      dev = g01fac(tail, p, NAGERR_DEFAULT);
      Vprintf("   %c         %5.3f          %6.4f\n", tail_char, p, dev);
    }
  exit(EXIT_SUCCESS);
}
```

**8.2.  Program Data**

```
g01fac Example Program Data
L 0.975
U 0.025
C 0.95
S 0.05
```

**8.3.  Program Results**

```
 g01fac Example Program Results

  Tail    Probability    Deviate

   L         0.975        1.9600
   U         0.025        1.9600
   C         0.950        1.9600
   S         0.050        1.9600
```