NAG Library Function Document

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 Wichura (1988). \(p\) 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
\[
x_p = \text{sign } q \left[ \frac{E(u)}{F(u)} + u \right],
\]
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 References

5 Arguments
1: tail – Nag_TailProbability  
   Input  
   On entry: indicates which tail the supplied probability represents.
   tail = Nag_LowerTail  
   The lower probability, i.e., \( P(X \leq x_p) \).
   tail = Nag_UpperTail  
   The upper probability, i.e., \( P(X \geq x_p) \).
   tail = Nag_TwoTailSignif  
   The two tail (significance level) probability, i.e., \( P(X \geq |x_p|) + P(X \leq -|x_p|) \).
   tail = Nag_TwoTailConfid  
   The two tail (confidence interval) probability, i.e., \( P(X \leq |x_p|) - P(X \leq -|x_p|) \).
   Constraint: tail = Nag_LowerTail, Nag_UpperTail, Nag_TwoTailSignif or Nag_TwoTailConfid.

2: p – double  
   Input  
   On entry: \( p \), the probability from the standard Normal distribution as defined by tail.
   Constraint: \( 0.0 < p < 1.0 \).

3: fail – NagError *  
   Input/Output  
   The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings
If on exit fail.code = NE_NOERROR, then nag_deviates_normal (g01fac) returns 0.0.

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 \( \langle \text{value} \rangle \) 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.
On entry, \( p = \text{value} \).
Constrait: \( p < 1.0 \).

### NE_REAL_ARG_LE

On entry, \( p = \text{value} \).
Constrait: \( p > 0.0 \).

### 7 Accuracy

The accuracy is mainly limited by the *machine precision*.

### 8 Parallelism and Performance

Not applicable.

### 9 Further Comments

None.

### 10 Example

Four values of \( \text{tail} \) and \( p \) are input and the deviates calculated and printed.

### 10.1 Program Text

```c
/* nag_deviates_normal (g01fac) Example Program.
 * Copyright 2014 Numerical Algorithms Group.
 * Mark 4, 1996.
 */
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg01.h>

int main(void)
{
    Integer exit_status = 0;
    double p;
    double dev;
    Integer i;
    char nag_enum_arg[40];
    Nag_TailProbability tail;
    NagError fail;

    INIT_FAIL(fail);

    printf("nag_deviates_normal (g01fac) Example Program Results\n");
    /* Skip heading in data file */
    #ifdef _WIN32
        scanf_s("%39s %lf ", nag_enum_arg, _countof(nag_enum_arg), &p);
    #else
        scanf("%39s %lf ", nag_enum_arg, &p);
    #endif
    for (i = 1; i <= 4; ++i)
    {
        #ifdef _WIN32
            scanf_s("%9s %lf ", nag_enum_arg, _countof(nag_enum_arg), &p);
        #else
            scanf("%9s %lf ", nag_enum_arg, &p);
        #endif
    }
    
    return exit_status;
}
```

---

Mark 25
#else
    scanf("%39s %lf ", nag_enum_arg, &p);
#endif

/* nag_enum_name_to_value (x04nac).
   * Converts NAG enum member name to value
   */
    tail = (Nag_TailProbability) nag_enum_name_to_value(nag_enum_arg);

/* nag_deviates_normal (g01fac).
   * Deviates for the Normal distribution
   */
    dev = nag_deviates_normal(tail, p, &fail);
    if (fail.code != NE_NOERROR)
     {
        printf("Error from nag_deviates_normal (g01fac).
   fail.message\n",
            fail.message);
             exit_status = 1;
             goto END;
    }
    printf(" %-17s %5.3f %6.4f\n", nag_enum_arg, p,
        dev);
}

END:
    return exit_status;
}

10.2 Program Data

nag_deviates_normal (g01fac) Example Program Data
Nag_LowerTail 0.975
Nag_UpperTail 0.025
Nag_TwoTailConfid 0.95
Nag_TwoTailSignif 0.05

10.3 Program Results

nag_deviates_normal (g01fac) Example Program Results

<table>
<thead>
<tr>
<th>Tail</th>
<th>Probability</th>
<th>Deviate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nag_LowerTail</td>
<td>0.975</td>
<td>1.9600</td>
</tr>
<tr>
<td>Nag_UpperTail</td>
<td>0.025</td>
<td>1.9600</td>
</tr>
<tr>
<td>Nag_TwoTailConfid</td>
<td>0.950</td>
<td>1.9600</td>
</tr>
<tr>
<td>Nag_TwoTailSignif</td>
<td>0.050</td>
<td>1.9600</td>
</tr>
</tbody>
</table>