NAG Library Function Document

nag_prob_normal (g01eac)

1 Purpose

nag_prob_normal (g01eac) returns a one or two tail probability for the standard Normal distribution.

2 Specification

```c
#include <nag.h>
#include <nagg01.h>
double nag_prob_normal (Nag_TailProbability tail, double x, NagError *fail)
```

3 Description

The lower tail probability for the standard Normal distribution, \( P(X \leq x) \) is defined by:

\[
P(X \leq x) = \int_{-\infty}^{x} Z(X) \, dX,
\]

where

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

The relationship

\[
P(X \leq x) = \frac{1}{2} \text{erfc}\left(\frac{x}{\sqrt{2}}\right)
\]

is used, where erfc is the complementary error function, and is computed using nag_erfc (s15adc). For the upper tail probability the relationship \( P(X \geq x) = P(X \leq -x) \) is used and for the two tail significance level probability twice the probability obtained from the absolute value of \( x \) is returned.

When the two tail confidence probability is required the relationship

\[
P(X \leq |x|) - P(X \leq -|x|) = \text{erf}\left(\frac{|x|}{\sqrt{2}}\right),
\]

is used, where erf is the error function, and is computed using nag_erf (s15aec).

4 References


5 Arguments

1:  tail – Nag_TailProbability

   Input
   
   On entry: indicates which tail the returned probability should represent.

   ```c
   tail = Nag_LowerTail
   The lower tail probability is returned, i.e., \( P(X \leq x) \).
   ```

   ```c
   tail = Nag_UpperTail
   The upper tail probability is returned, i.e., \( P(X \geq x) \).
   ```

Mark 25
\textbf{tail} = \textsc{Nag\_TwoTailSignif}

The two tail (significance level) probability is returned, i.e., \( P(X \geq |x|) + P(X \leq -|x|) \).

\textbf{tail} = \textsc{Nag\_TwoTailConfid}

The two tail (confidence interval) probability is returned, i.e., \( P(X \leq |x|) - P(X \leq -|x|) \).

\textit{Constraint}: \textbf{tail} = \textsc{Nag\_LowerTail}, \textsc{Nag\_UpperTail}, \textsc{Nag\_TwoTailSignif} or \textsc{Nag\_TwoTailConfid}.

2: \hspace{1cm} \textbf{x} \hspace{0.5cm} – \textbf{double} \hspace{0.5cm} \textit{Input}

\textit{On entry}: \textbf{x}, the value of the standard Normal variate.

3: \hspace{1cm} \textbf{fail} \hspace{0.5cm} – \textsc{NagError} * \hspace{0.5cm} \textit{Input/Output}

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

6 \hspace{1cm} \textbf{Error Indicators and Warnings}

\textbf{NE\_ALLOC\_FAIL}

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

\textbf{NE\_BAD\_PARAM}

On entry, argument \textit{\textless value\textgreater} had an illegal value.

\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.

7 \hspace{1cm} \textbf{Accuracy}

Accuracy is limited by \textit{machine precision}. For detailed error analysis see \texttt{nag\_erfc} (s15adc) and \texttt{nag\_erf} (s15aec).

8 \hspace{1cm} \textbf{Parallelism and Performance}

Not applicable.

9 \hspace{1cm} \textbf{Further Comments}

None.

10 \hspace{1cm} \textbf{Example}

Four values of \textbf{tail} and \textbf{x} are input and the probabilities calculated and printed.
#include <nag.h>
#include <nag_stdlib.h>
#include <stdio.h>
#include <nag_g01.h>

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

    INIT_FAIL(fail);

    printf("nag_prob_normal (g01eac) Example Program Results\n");

    /* Skip heading in data file */
    ifndef _WIN32
        scanf_s("%*[\n] ");
    else
        scanf("%*[\n] ");
    endif

    printf("\n Tail X Probability \n\n");
    for (i = 1; i <= 4; ++i)
    {
        ifndef _WIN32
            scanf_s("%39s %lf ", nag_enum_arg, _countof(nag_enum_arg), &x);
        else
            scanf("%39s %lf ", nag_enum_arg, &x);
        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_prob_normal (g01eac).
         * Probabilities for the standard Normal distribution
         */
        prob = nag_prob_normal(tail, x, &fail);
        if (fail.code != NE_NOERROR)
        {
            printf("Error from nag_prob_normal (g01eac).\n", fail.message);
            exit_status = 1;
            goto END;
        }

        printf(" %-17s %4.2f %6.4f\n", nag_enum_arg, x, prob);
    }

    END:
    return exit_status;
}
10.2 Program Data

*nag_prob_normal* (g01eac) Example Program Data
Nag_LowerTail  1.96
Nag_UpperTail  1.96
Nag_TwoTailConfid  1.96
Nag_TwoTailSignif  1.96

10.3 Program Results

*nag_prob_normal* (g01eac) Example Program Results

<table>
<thead>
<tr>
<th>Tail</th>
<th>X</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nag_LowerTail</td>
<td>1.96</td>
<td>0.9750</td>
</tr>
<tr>
<td>Nag_UpperTail</td>
<td>1.96</td>
<td>0.0250</td>
</tr>
<tr>
<td>Nag_TwoTailConfid</td>
<td>1.96</td>
<td>0.9500</td>
</tr>
<tr>
<td>Nag_TwoTailSignif</td>
<td>1.96</td>
<td>0.0500</td>
</tr>
</tbody>
</table>