**NAG Library Function Document**

*nag_1_sample_ks_test (g08cbc)*

1 **Purpose**

*nag_1_sample_ks_test (g08cbc)* performs the one sample Kolmogorov–Smirnov test, using one of the distributions provided.

2 **Specification**

```c
#include <nag.h>
#include <nagg08.h>

void nag_1_sample_ks_test (Integer n, const double x[], Nag_Distributions dist, double par[], Nag_ParaEstimates estima,
                          Nag_TestStatistics ntype, double *d, double *z, double *p,
                          NagError *fail)
```

3 **Description**

The data consist of a single sample of \( n \) observations denoted by \( x_1, x_2, \ldots, x_n \). Let \( S_n(x_{(i)}) \) and \( F_0(x_{(i)}) \) represent the sample cumulative distribution function and the theoretical (null) cumulative distribution function respectively at the point \( x_{(i)} \) where \( x_{(i)} \) is the \( i \)th smallest sample observation.

The Kolmogorov–Smirnov test provides a test of the null hypothesis \( H_0 \): the data are a random sample from a theoretical distribution specified by you against one of the following alternative hypotheses:

(i) \( H_1 \): the data cannot be considered to be a random sample from the specified null distribution.

(ii) \( H_2 \): the data arise from a distribution which dominates the specified null distribution. In practical terms, this would be demonstrated if the values of the sample cumulative distribution function \( S_n(x) \) tended to exceed the corresponding values of the theoretical cumulative distribution function \( F_0(x) \).

(iii) \( H_3 \): the data arise from a distribution which is dominated by the specified null distribution. In practical terms, this would be demonstrated if the values of the theoretical cumulative distribution function \( F_0(x) \) tended to exceed the corresponding values of the sample cumulative distribution function \( S_n(x) \).

One of the following test statistics is computed depending on the particular alternative null hypothesis specified (see the description of the argument \texttt{ntype} in Section 5).

For the alternative hypothesis \( H_1 \).

\[ D_n = \text{the largest absolute deviation between the sample cumulative distribution function and the theoretical cumulative distribution function. Formally } D_n = \max\{D_n^+, D_n^-\}. \]

For the alternative hypothesis \( H_2 \).

\[ D_n^+ = \text{the largest positive deviation between the sample cumulative distribution function and the theoretical cumulative distribution function. Formally } D_n^+ = \max\{S_n(x_{(i)}) - F_0(x_{(i)}), 0\} \text{ for both discrete and continuous null distributions.} \]

For the alternative hypothesis \( H_3 \).

\[ D_n^- = \text{the largest positive deviation between the theoretical cumulative distribution function and the sample cumulative distribution function. Formally if the null distribution is discrete then } D_n^- = \max\{F_0(x_{(i)}) - S_n(x_{(i)}), 0\} \text{ and if the null distribution is continuous then } D_n^- = \max\{F_0(x_{(i)}) - S_n(x_{(i-1)}), 0\}. \]
The standardized statistic \( Z = \frac{D}{\sqrt{n}} \) is also computed where \( D \) may be \( D_n, D_n^+ \) or \( D_n^- \) depending on the choice of the alternative hypothesis. This is the standardized value of \( D \) with no correction for continuity applied and the distribution of \( Z \) converges asymptotically to a limiting distribution, first derived by Kolmogorov (1933), and then tabulated by Smirnov (1948). The asymptotic distributions for the one-sided statistics were obtained by Smirnov (1933).

The probability, under the null hypothesis, of obtaining a value of the test statistic as extreme as that observed, is computed. If \( n \leq 100 \) an exact method given by Conover (1980), is used. Note that the method used is only exact for continuous theoretical distributions and does not include Conover’s modification for discrete distributions. This method computes the one-sided probabilities. The two-sided probabilities are estimated by doubling the one-sided probability. This is a good estimate for small \( p \), that is \( p \leq 0.10 \), but it becomes very poor for larger \( p \). If \( n > 100 \) then \( p \) is computed using the Kolmogorov–Smirnov limiting distributions, see Feller (1948), Kendall and Stuart (1973), Kolmogorov (1933), Smirnov (1933) and Smirnov (1948).

4 References


Kolmogorov A N (1933) Sulla determinazione empirica di una legge di distribuzione Giornale dell’Istituto Italiano degli Attuari 4 83–91


Smirnov N (1933) Estimate of deviation between empirical distribution functions in two independent samples Bull. Moscow Univ. 2(2) 3–16


5 Arguments

1: \( n \) – Integer

   \textit{Input}

   \textit{On entry:} \( n \), the number of observations in the sample.

   \textit{Constraint:} \( n \geq 3 \).

2: \( x[n] \) – const double

   \textit{Input}

   \textit{On entry:} the sample observations \( x_1, x_2, \ldots, x_n \).

   \textit{Constraint:} the sample observations supplied must be consistent, in the usual manner, with the null distribution chosen, as specified by the arguments \texttt{dist} and \texttt{par}. For further details see Section 9.

3: \( \texttt{dist} \) – Nag_Distributions

   \textit{Input}

   \textit{On entry:} the theoretical (null) distribution from which it is suspected the data may arise.

   \( \texttt{dist} = \texttt{Nag\_Uniform} \)

   The uniform distribution over \((a, b)\).

   \( \texttt{dist} = \texttt{Nag\_Normal} \)

   The Normal distribution with mean \( \mu \) and variance \( \sigma^2 \).

   \( \texttt{dist} = \texttt{Nag\_Gamma} \)

   The gamma distribution with shape parameter \( \alpha \) and scale parameter \( \beta \), where the mean = \( \alpha \beta \).
The beta distribution with shape parameters $\alpha$ and $\beta$, where the mean $= \alpha / (\alpha + \beta)$.

The binomial distribution with the number of trials, $m$, and the probability of a success, $p$.

The exponential distribution with parameter $\lambda$, where the mean $= 1 / \lambda$.

The Poisson distribution with parameter $\mu$, where the mean $= \mu$.

The negative binomial distribution with the number of trials, $m$, and the probability of success, $p$.

The generalized Pareto distribution with shape parameter $\xi$ and scale $\beta$.

**Constraint:** $\text{dist} = \text{Nag Uniform, Nag Normal, Nag Gamma, Nag Beta, Nag Binomial, Nag Exponential, Nag Poisson, Nag NegBinomial or Nag GenPareto}$.

4: $\text{par}[2] = \text{double}$

*Input/Output*

**On entry:** if $\text{estima} = \text{Nag ParaSupplied}$, $\text{par}$ must contain the known values of the parameter(s) of the null distribution as follows.

If a uniform distribution is used, then $\text{par}[0]$ and $\text{par}[1]$ must contain the boundaries $a$ and $b$ respectively.

If a Normal distribution is used, then $\text{par}[0]$ and $\text{par}[1]$ must contain the mean, $\mu$, and the variance, $\sigma^2$, respectively.

If a gamma distribution is used, then $\text{par}[0]$ and $\text{par}[1]$ must contain the parameters $\alpha$ and $\beta$ respectively.

If a beta distribution is used, then $\text{par}[0]$ and $\text{par}[1]$ must contain the parameters $\alpha$ and $\beta$ respectively.

If a binomial distribution is used, then $\text{par}[0]$ and $\text{par}[1]$ must contain the parameters $m$ and $p$ respectively.

If an exponential distribution is used, then $\text{par}[0]$ must contain the parameter $\lambda$.

If a Poisson distribution is used, then $\text{par}[0]$ must contain the parameter $\mu$.

If a negative binomial distribution is used, $\text{par}[0]$ and $\text{par}[1]$ must contain the parameters $m$ and $p$ respectively.

If a generalized Pareto distribution is used, $\text{par}[0]$ and $\text{par}[1]$ must contain the parameters $\xi$ and $\beta$ respectively.

If $\text{estima} = \text{Nag ParaEstimated}$, $\text{par}$ need not be set except when the null distribution requested is either the binomial or the negative binomial distribution in which case $\text{par}[0]$ must contain the parameter $m$.

**On exit:** if $\text{estima} = \text{Nag ParaSupplied}$, $\text{par}$ is unchanged; if $\text{estima} = \text{Nag ParaEstimated}$, and $\text{dist} = \text{Nag Binomial}$ or $\text{dist} = \text{Nag NegBinomial}$ then $\text{par}[1]$ is estimated from the data; otherwise $\text{par}[0]$ and $\text{par}[1]$ are estimated from the data.

**Constraints:**

if $\text{dist} = \text{Nag Uniform}$, $\text{par}[0] < \text{par}[1]$;
if $\text{dist} = \text{Nag Normal}$, $\text{par}[1] > 0.0$;
if $\text{dist} = \text{Nag Gamma}$, $\text{par}[0] > 0.0$ and $\text{par}[1] > 0.0$;
if $\text{dist} = \text{Nag Beta}$, $\text{par}[0] > 0.0$ and $\text{par}[1] > 0.0$ and $\text{par}[0] \leq 10^6$ and $\text{par}[1] \leq 10^6$;
if \( \text{dist} = \text{Nag\_Binomial} \), \( \text{par}[0] \geq 1.0 \) and \( 0.0 < \text{par}[1] < 1.0 \) and
\( \text{par}[0] \times \text{par}[1] \times (1.0 - \text{par}[1]) \leq 10^6 \) and \( \text{par}[0] < 1/\text{eps} \), where
\( \text{eps} = \text{machine precision} \), see nag\_machine\_precision (X02AJC);
if \( \text{dist} = \text{Nag\_Exponential} \), \( \text{par}[0] > 0.0 \);
if \( \text{dist} = \text{Nag\_Poisson} \), \( \text{par}[0] > 0.0 \) and \( \text{par}[0] \leq 10^6 \);
if \( \text{dist} = \text{Nag\_NegBinomial} \), \( \text{par}[0] \geq 1.0 \) and \( 0.0 < \text{par}[1] < 1.0 \) and
\( \text{par}[0] \times (1.0 - \text{par}[1])/(\text{par}[1] \times \text{par}[1]) \leq 10^6 \) and \( \text{par}[0] < 1/\text{eps} \), where
\( \text{eps} = \text{machine precision} \), see nag\_machine\_precision (X02AJC);
if \( \text{dist} = \text{Nag\_GenPareto} \), \( \text{par}[1] > 0.5 \):
estima – Nag\_ParaEstimates

Input

On entry: \estima must specify whether values of the parameters of the null distribution are known or are to be estimated from the data.

\estima = Nag\_ParaSupplied
Values of the parameters will be supplied in the array \par described above.

\estima = Nag\_ParaEstimated
Parameters are to be estimated from the data except when the null distribution requested is
the binomial distribution or the negative binomial distribution in which case the first parameter, \( m \), must be supplied in \par[0] and only the second parameter, \( p \), is estimated from the data.

Constraint: \estima = Nag\_ParaSupplied or Nag\_ParaEstimated.

ntype – Nag\_TestStatistics

Input

On entry: the test statistic to be calculated, i.e., the choice of alternative hypothesis.

\ntype = Nag\_TestStatisticsDAbs
Computes \( D_n \), to test \( H_0 \) against \( H_1 \),
\ntype = Nag\_TestStatisticsDPos
Computes \( D_n^+ \), to test \( H_0 \) against \( H_2 \),
\ntype = Nag\_TestStatisticsDNeg
Computes \( D_n^- \), to test \( H_0 \) against \( H_3 \).

Constraint: \ntype = Nag\_TestStatisticsDAbs, Nag\_TestStatisticsDPos or Nag\_TestStatisticsDNeg.

d – double *

Output

On exit: the Kolmogorov–Smirnov test statistic (\( D_n \), \( D_n^+ \) or \( D_n^- \) according to the value of \ntype).

z – double *

Output

On exit: a standardized value, \( Z \), of the test statistic, \( D \), without any correction for continuity.

p – double *

Output

On exit: the probability, \( p \), associated with the observed value of \( D \) where \( D \) may be \( D_n \), \( D_n^+ \) or \( D_n^- \) depending on the value of \ntype (see Section 3).

fail – NagError *

Input/Output

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

6 Error Indicators and Warnings

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 value \rangle had an illegal value.

NE_G08CB_DATA
On entry, \texttt{dist} = \texttt{Nag\_Beta} and at least one observation is illegal.
Constraint: $0 \leq x[i-1] \leq 1$, for $i = 1, 2, \ldots, n$.

On entry, \texttt{dist} = \texttt{Nag\_Binomial} and all observations are zero or $m$.
Constraint: at least one $0 < x[i-1] < \texttt{par}[0]$, for $i = 1, 2, \ldots, n$.

On entry, \texttt{dist} = \texttt{Nag\_Binomial} and at least one observation is illegal.
Constraint: $0 \leq x[i-1] \leq \texttt{par}[0]$, for $i = 1, 2, \ldots, n$.

On entry, \texttt{dist} = \texttt{Nag\_Binomial} and at least one observation is illegal.
Constraint: $0 \leq x[i-1] \leq \texttt{par}[0]$, for $i = 1, 2, \ldots, n$.

On entry, \texttt{dist} = \texttt{Nag\_Exponential} or \texttt{Nag\_Poisson} and all observations are zero.
Constraint: at least one $x[i-1] > 0$, for $i = 1, 2, \ldots, n$.

On entry, \texttt{dist} = \texttt{Nag\_Uniform} and at least one observation is illegal.
Constraint: $\texttt{par}[0] \leq x[i-1] \leq \texttt{par}[1]$, for $i = 1, 2, \ldots, n$.

NE_G08CB_PARAM
On entry, \texttt{dist} = \texttt{Nag\_Binomial} and $m = \texttt{par}[0] = \langle value \rangle$.
Note that $m$ must always be supplied.
Constraint: for the binomial distribution, $1 \leq \texttt{par}[0] < 1/\texttt{eps}$, where $\texttt{eps} = \texttt{machine precision}$, see nag\_machine\_precision (X02AJC).

On entry, \texttt{dist} = \texttt{Nag\_GenPareto} and \texttt{estima} = \texttt{Nag\_ParaEstimated}.
The parameter estimates are invalid; the data may not be from the generalized Pareto distribution.

On entry, \texttt{dist} = \texttt{Nag\_NegBinomial} and $m = \texttt{par}[0] = \langle value \rangle$.
Note that $m$ must always be supplied.
Constraint: for the negative binomial distribution, $1 \leq \texttt{par}[0] < 1/\texttt{eps}$, where $\texttt{eps} = \texttt{machine precision}$, see nag\_machine\_precision (X02AJC).

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[0] = \langle value \rangle$; $\texttt{par}[1] = \langle value \rangle$.
Constraint: for the gamma distribution, $0 < \texttt{par}[0]$ and $\texttt{par}[1] \leq 1000000$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[0] = \langle value \rangle$; $\texttt{par}[1] = \langle value \rangle$.
Constraint: for the gamma distribution, $\texttt{par}[0]$ and $\texttt{par}[1] > 0$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[0] = \langle value \rangle$; $\texttt{par}[1] = \langle value \rangle$.
Constraint: for the generalized Pareto distribution with $\texttt{par}[0] < 0$, $0 \leq x[i-1] \leq -\texttt{par}[1]/\texttt{par}[0]$, for $i = 1, 2, \ldots, n$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[0] = \langle value \rangle$; $\texttt{par}[1] = \langle value \rangle$.
Constraint: for the uniform distribution, $\texttt{par}[0] < \texttt{par}[1]$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[0] = \langle value \rangle$.
Constraint: for the exponential distribution, $\texttt{par}[0] > 0$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[0] = \langle value \rangle$.
Constraint: for the Poisson distribution, $0 < \texttt{par}[0] < 1000000$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[1] = \langle value \rangle$.
Constraint: for the binomial distribution, $0 < \texttt{par}[1] < 1$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[1] = \langle value \rangle$.
Constraint: for the generalized Pareto distribution, $\texttt{par}[1] > 0$.

On entry, \texttt{estima} = \texttt{Nag\_ParaSupplied} and $\texttt{par}[1] = \langle value \rangle$.
Constraint: for the negative binomial distribution, $0 < \texttt{par}[1] < 1$. 

Mark 25
On entry, estima = Nag_ParaSupplied and \( \text{par}[1] = \langle \text{value} \rangle \).
Constraint: for the Normal distribution, \( \text{par}[1] > 0 \).

**NE_G08CB_SAMPLE**

On entry, \( \text{dist} = \text{Nag\_Uniform}, \text{Nag\_Normal}, \text{Nag\_Gamma}, \text{Nag\_Beta} \text{ or Nag\_GenPareto}, \)
estima = Nag\_ParaEstimated and the whole sample is constant. Thus the variance is zero.

**NE_G08CB_VARIANCE**

On entry, \( \text{dist} = \text{Nag\_Binomial}, \text{par}[0] = \langle \text{value} \rangle, \text{par}[1] = \langle \text{value} \rangle \).
The variance \( \text{par}[0] \times \text{par}[1] \times (1 - \text{par}[1]) \) exceeds 1000000.

On entry, \( \text{dist} = \text{Nag\_NegBinomial}, \text{par}[0] = \langle \text{value} \rangle, \text{par}[1] = \langle \text{value} \rangle \).
The variance \( \text{par}[0] \times (1 - \text{par}[1])/(\text{par}[1] \times \text{par}[1]) \) exceeds 1000000.

**NE_INT_ARG_LT**

On entry, \( n = \langle \text{value} \rangle \).
Constraint: \( n \geq 3 \).

**NE_INTERNAL_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the
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.

7 **Accuracy**

The approximation for \( p \), given when \( n > 100 \), has a relative error of at most 2.5% for most cases. The
two-sided probability is approximated by doubling the one-sided probability. This is only good for small
\( p \), i.e., \( p < 0.10 \) but very poor for large \( p \). The error is always on the conservative side, that is the tail
probability, \( p \), is over estimated.

8 **Parallelism and Performance**

nag\_1\_sample\_ks\_test (g08cbc) is threaded by NAG for parallel execution in multithreaded
implementations of the NAG Library.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the
OpenMP environment used within this function. Please also consult the Users’ Note for your
implementation for any additional implementation-specific information.

9 **Further Comments**

The time taken by nag\_1\_sample\_ks\_test (g08cbc) increases with \( n \) until \( n > 100 \) at which point it drops
and then increases slowly with \( n \). The time may also depend on the choice of null distribution and on
whether or not the parameters are to be estimated.

The data supplied in the argument \( x \) must be consistent with the chosen null distribution as follows:

- when \( \text{dist} = \text{Nag\_Uniform} \), then \( \text{par}[0] \leq x_i \leq \text{par}[1] \), for \( i = 1, 2, \ldots, n \);
- when \( \text{dist} = \text{Nag\_Normal} \), then there are no constraints on the \( x_i \)'s;
- when \( \text{dist} = \text{Nag\_Gamma} \), then \( x_i \geq 0.0 \), for \( i = 1, 2, \ldots, n \);
when $\text{dist} = \text{Nag\_Beta}$, then $0.0 \leq x_i \leq 1.0$, for $i = 1, 2, \ldots, n$;
when $\text{dist} = \text{Nag\_Binomial}$, then $0.0 \leq x_i \leq \text{par}[0]$, for $i = 1, 2, \ldots, n$;
when $\text{dist} = \text{Nag\_Exponential}$, then $x_i \geq 0.0$, for $i = 1, 2, \ldots, n$;
when $\text{dist} = \text{Nag\_Poisson}$, then $x_i \geq 0.0$, for $i = 1, 2, \ldots, n$;
when $\text{dist} = \text{Nag\_NegBinomial}$, then $x_i \geq 0.0$, for $i = 1, 2, \ldots, n$;
when $\text{dist} = \text{Nag\_GenPareto}$ and $\text{par}[0] \geq 0.0$, then $x_i \geq 0.0$, for $i = 1, 2, \ldots, n$;
when $\text{dist} = \text{Nag\_GenPareto}$ and $\text{par}[0] < 0.0$, then $0.0 \leq x_i \leq -\text{par}[1]/\text{par}[0]$, for $i = 1, 2, \ldots, n$.

10 Example

The following example program reads in a set of data consisting of 30 observations. The Kolmogorov–Smirnov test is then applied twice, firstly to test whether the sample is taken from a uniform distribution, $U(0,2)$, and secondly to test whether the sample is taken from a Normal distribution where the mean and variance are estimated from the data. In both cases we are testing against $H_1$; that is, we are doing a two tailed test. The values of $d$, $z$ and $p$ are printed for each case.

10.1 Program Text

/ * nag_1_sample_ks_test (g08cbc) Example Program. 
 * 
 * Copyright 2014 Numerical Algorithms Group. 
 * 
 * Mark 6, 2000. 
 */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg08.h>

int main(void)
{
    Integer exit_status = 0;
    Integer i, n, np;
    double d, p, *par = 0, *x = 0, z;
    char nag_enum_arg[40];
    Nag_TestStatistics ntype;
    NagError fail;

    INIT_FAIL(fail);

    printf("nag_1_sample_ks_test (g08cbc) Example Program Results\n");

    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n"]);
#else
    scanf("%*[\n"]);
#endif
#ifdef _WIN32
    scanf_s("%"NAG_IFMT"",&n);
#else
    scanf("%"NAG_IFMT","&n);
#endif
    x = NAG_ALLOC(n, double);

    printf("\n");
    for (i = 1; i <= n; ++i)
#ifdef _WIN32
    scanf_s("%lf", &x[i - 1]);
#else
scanf("%lf", &x[i - 1]);
#endif
#else
scanf("%"NAG_IFMT"", &np);
#endif
if (!(par = NAG_ALLOC(np, double)))
{
printf("Allocation failure\n");
exit_status = -1;
goto END;
}
for (i = 1; i <= np; ++i)
#else
scanf("%lf", &par[i - 1]);
#endif
#define _WIN32
scanf_s("%lf", &par[i - 1]);
#endif
#define _WIN32
scanf_s("%39s", nag_enum_arg, _countof(nag_enum_arg));
#endif
ntype = (Nag_TestStatistics) nag_enum_name_to_value(nag_enum_arg);
/* nag_enum_name_to_value (x04nac).
* Converts NAG enum member name to value
*/
ntype = (Nag_TestStatistics) nag_enum_name_to_value(nag_enum_arg);
#endif
#define _WIN32
scanf_s("%1f", &par[i - 1]);
#endif
#define _WIN32
scanf("%"NAG_IFMT"", &np);
#endif
for (i = 1; i <= np; ++i)
#else
scanf("%"NAG_IFMT"", &np);
#endif
#define _WIN32
scanf_s("%39s", nag_enum_arg, _countof(nag_enum_arg));
#endif
ntype = (Nag_TestStatistics) nag_enum_name_to_value(nag_enum_arg);
/* nag_1_sample_ks_test (g08cbc). 
* Performs the one-sample Kolmogorov-Smirnov test for 
* standard distributions 
*/
nag_1_sample_ks_test(n, x, Nag_Normal, par, Nag_ParaEstimated, ntype, &d, &z, &p, &fail);
if (fail.code != NE_NOERROR)
{
printf("Error from nag_1_sample_ks_test (g08cbc).\n%s\n", fail.message);
exit_status = 1;
goto END;
}
printf("Test against uniform distribution on (0,2)\n");
printf("\n");
printf("Test statistic D = %.4f\n", d);
printf("Z statistic = %.4f\n", z);
printf("Tail probability = %.4f\n", p);
printf("\n");
#endif
#define _WIN32
scanf("%"NAG_IFMT"", &np);
#endif
for (i = 1; i <= np; ++i)
#else
scanf("%"NAG_IFMT"", &np);
#endif
#define _WIN32
scanf_s("%1f", &par[i - 1]);
#endif
#define _WIN32
scanf("%39s", nag_enum_arg, _countof(nag_enum_arg));
#endif
ntype = (Nag_TestStatistics) nag_enum_name_to_value(nag_enum_arg);
/* nag_1_sample_ks_test (g08cbc), see above. */
nag_1_sample_ks_test(n, x, Nag_Normal, par, Nag_ParaEstimated, ntype, &d, &z, &p, &fail);
if (fail.code != NE_NOERROR)


printf("Error from nag_1_sample_kstest (g08cbc).\n%s\n", fail.message);
exit_status = 1;
goto END;
}

printf("Test against Normal distribution with parameters estimated" 
" from the data\n\n");
printf("Mean = %6.4f and variance = %6.4f\n", par[0], par[1]);
printf("Test statistic D = %8.4f\n", d);
printf("Z statistic = %8.4f\n", z);
printf("Tail probability = %8.4f\n", p);

END:
NAG_FREE(x);
NAG_FREE(par);

return exit_status;
}

10.2 Program Data

nag_1_sample_kstest (g08cbc) Example Program Data

<table>
<thead>
<tr>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.01</td>
</tr>
<tr>
<td>0.30</td>
</tr>
<tr>
<td>0.20</td>
</tr>
<tr>
<td>0.90</td>
</tr>
<tr>
<td>1.20</td>
</tr>
<tr>
<td>0.09</td>
</tr>
<tr>
<td>1.30</td>
</tr>
<tr>
<td>0.18</td>
</tr>
<tr>
<td>0.90</td>
</tr>
<tr>
<td>0.48</td>
</tr>
<tr>
<td>1.98</td>
</tr>
<tr>
<td>0.03</td>
</tr>
<tr>
<td>0.50</td>
</tr>
<tr>
<td>0.07</td>
</tr>
<tr>
<td>0.70</td>
</tr>
<tr>
<td>0.60</td>
</tr>
<tr>
<td>0.95</td>
</tr>
<tr>
<td>1.00</td>
</tr>
<tr>
<td>0.31</td>
</tr>
<tr>
<td>1.45</td>
</tr>
<tr>
<td>1.04</td>
</tr>
<tr>
<td>1.25</td>
</tr>
<tr>
<td>0.15</td>
</tr>
<tr>
<td>0.75</td>
</tr>
<tr>
<td>0.85</td>
</tr>
<tr>
<td>0.22</td>
</tr>
<tr>
<td>1.56</td>
</tr>
<tr>
<td>0.81</td>
</tr>
<tr>
<td>0.57</td>
</tr>
<tr>
<td>0.55</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>2.0</td>
</tr>
<tr>
<td>Nag_TestStatisticsDAbs</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>0.0</td>
</tr>
<tr>
<td>1.0</td>
</tr>
<tr>
<td>Nag_TestStatisticsDAbs</td>
</tr>
</tbody>
</table>

10.3 Program Results

nag_1_sample_kstest (g08cbc) Example Program Results

Test against uniform distribution on (0,2)
Test statistic D = 0.2800
Z statistic = 1.5336
Tail probability = 0.0143

Test against Normal distribution with parameters estimated from the data
Mean = 0.6967 and variance = 0.2564
Test statistic D = 0.1108
Z statistic = 0.6068
Tail probability = 0.8925