NAG C 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 standard distributions provided.

2 Specification

#include <nag.h>
#include <nagf08.h>

void nag_1_sample ks_test (Integer n, const double x[],
    Nag_Distributions dist, double par[], Nag_ParamEstimates estima,
    Nag_TestStatistics dtype, 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, ..., 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 of observations from a theoretical distribution specified by the user 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 parameter dtype in Section 4).

For the alternative hypothesis H_1,

\[ D_n = \text{the largest absolute deviation between the sample cumulative distribution function and the theoretical cumulative distribution function}. \]

For the alternative hypothesis H_2,

\[ D_n^+ = \text{the largest positive deviation between the sample cumulative distribution function and the theoretical cumulative distribution function}. \]

For the alternative hypothesis H_3,

\[ D_n^- = \text{the largest positive deviation between the theoretical cumulative distribution function and the sample cumulative distribution function}. \]

The standardized statistic \( Z = D \times \sqrt{n} \) 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 standardised 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 Parameters

1: \( n \) – Integer

*Input*

*On entry:* the number of observations in the sample, \( n \).

*Constraint:* \( n \geq 3 \).

2: \( x[n] \) – const double

*Input*

*On entry:* the sample observations \( x_1, x_2, \ldots, x_n \).

*Constraint:* the sample observations supplied must be consistent, in the usual manner, with the null distribution chosen, as specified by the parameters dist and par. For further details see Section 6.

3: dist – Nag_Distributions

*Input*

*On entry:* the theoretical (null) distribution from which it is suspected the data may arise, as follows:

- \( \text{dist} = \text{Nag\_Uniform} \), uniform distribution over \( (a, b) - U(a, b) \).
- \( \text{dist} = \text{Nag\_Normal} \), Normal distribution with mean \( \mu \) and variance \( \sigma^2 - N(\mu, \sigma^2) \).
- \( \text{dist} = \text{Nag\_Gamma} \), gamma distribution with shape parameter \( \alpha \) and scale parameter \( \beta \), where the mean = \( \alpha \beta \).
- \( \text{dist} = \text{Nag\_Beta} \), beta distribution with shape parameters \( \alpha \) and \( \beta \), where the mean = \( \alpha / (\alpha + \beta) \).
- \( \text{dist} = \text{Nag\_Binomial} \), binomial distribution with the number of trials, \( m \), and the probability of a success, \( p \).
- \( \text{dist} = \text{Nag\_Exponential} \), exponential distribution with parameter \( \lambda \), where the mean = \( 1 / \lambda \).
- \( \text{dist} = \text{Nag\_Poisson} \), poisson distribution with parameter \( \mu \), where the mean = \( \mu \).

*Constraint:* dist = Nag\_Uniform, Nag\_Normal, Nag\_Gamma, Nag\_Beta, Nag\_Binomial, Nag\_Exponential or Nag\_Poisson.

4: \( \text{par}[2] \) – double

*Input/Output*

*On entry:* if estima = Nag\_ParaSupplied, 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 a 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 \( \text{estima} = \text{Nag_ParaEstimated} \), \( \text{par} \) need not be set except when the null distribution requested is the 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 \( \text{estima} = \text{Nag_ParaEstimated} \), then \( \text{par}[0] \) and \( \text{par}[1] \) are set to values as 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 \( \text{eps} = \text{the machine precision} \), 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 \( \text{par}[0] \leq 10^6 \).

5: \( \text{estima} \) – Nag_ParaEstimates

**Input**

**On entry:** \( \text{estima} \) must specify whether values of the parameters of the null distribution are known or are to be estimated from the data:  
if \( \text{estima} = \text{Nag_ParaSupplied} \), values of the parameters will be supplied in the array \( \text{par} \) described above;  
if \( \text{estima} = \text{Nag_ParaEstimated} \), parameters are to be estimated from the data except when the null distribution requested is the binomial distribution in which case the first parameter, \( m \), must be supplied in \( \text{par}[0] \) and only the second parameter, \( p \) is estimated from the data.

**Constraint:** \( \text{estima} = \text{Nag_ParaSupplied} \) or \( \text{Nag_ParaEstimated} \).

6: \( \text{dtype} \) – Nag_TextStatistics

**Input**

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

\( \text{dtype} = \text{Nag_TextStatisticsDAbs} \) : Computes \( D_n \), to test \( H_0 \) against \( H_1 \),
\( \text{dtype} = \text{Nag_TextStatisticsDPos} \) : Computes \( D^+_n \), to test \( H_0 \) against \( H_2 \),
\( \text{dtype} = \text{Nag_TextStatisticsDNeg} \) : Computes \( D^-_n \), to test \( H_0 \) against \( H_3 \).

**Constraint:** \( \text{dtype} = \text{Nag_TextStatisticsDAbs}, \text{Nag_TextStatisticsDPos} \) or \( \text{Nag_TextStatisticsDNeg} \).

7: \( d \) – double *

**Output**

**On exit:** the Kolmogorov–Smirnov test statistic (\( D_n \), \( D^+_n \) or \( D^-_n \) according to the value of \( \text{dtype} \)).

8: \( z \) – double *

**Output**

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

9: \( p \) – 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 \( \text{dtype} \) (see Section 3).
5  Error Indicators and Warnings

NE_INT_ARG_LT
On entry, n must not be less than 3: n = <value>.

NE_BAD_PARAM
On entry, parameter dist had an illegal value.
On entry, parameter estima had an illegal value.
On entry, parameter dtype had an illegal value.

NE_G08CB_PARAM
On entry, the parameters supplied for the specified null distribution are out of range. This error will
only occur if estima = Nag_ParaEstimates.

NE_G08CB_DATA
The data supplied in x could not arise from the chosen null distribution, as specified by the
parameters dist and par.

NE_G08CB_SAMPLE
The whole sample is constant i.e., the variance is zero. This error may only occur if (dist =
Nag_Uniform, Nag_Normal, Nag_Gamma or Nag_Beta) and estima = Nag_ParaEstimatesE.

NE_G08CB_VARIANCE
The variance of the binomial distribution (dist = Nag_Binomial) is too large. That is np(1 − p) >
1.0e6.

NE_G08CB_INCOMP_GAMMA
When dist = Nag_Gamma, in the computation of the incomplete gamma function by
nag_incomplete_gamma (s14bac) the convergence of the Taylor’s series or Legendre continued
fraction fails within 600 iterations.

NE_ALLOC_FAIL
Memory allocation failed.

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
The time taken by the routine 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 parameter x must be consistent with the chosen null distribution as follows:

when dist = Nag_Uniform, then par[0] ≤ x[i] ≤ par[1], for i = 1, 2,..., n;
when dist = Nag_Normal, then there are no constraints on the x[i]’s;
when dist = Nag_Gamma, then x[i] ≥ 0.0, for i = 1, 2,..., 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 \( i = 1, 2, \ldots, n \).

6.1 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.

6.2 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

7 See Also
None.

8 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.

8.1 Program Text

```c
/* nag_l_sample_ks_test (g08cbc) Example Program. */
/* Copyright 2000 Numerical Algorithms Group. */
/* Mark 6, 2000. */
*/

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nag08.h>

int main (void)
```

{
    double d, p, *par=0, *x=0, z;
    Integer i, n, np, ntype;
    Integer exit_status=0;
    Nag_TestStatistics ntype_enum;
    NagError fail;

    INIT_FAIL(fail);
    Vprintf("g08cbc Example Program Results\n");

    /* Skip heading in data file */
    Vscanf("%*[\n"]);

    Vscanf("%ld", &n);
    x = NAG_ALLOC(n, double);

    Vprintf("\n");
    for (i = 1; i <= n; ++i)
        Vscanf("%lf", &x[i - 1]);
    Vscanf("%ld", &np);
    if (!((par = NAG_ALLOC(np, double))
        {
            Vprintf("Allocation failure\n");
            exit_status = -1;
            goto END;
        }
    }

    for (i = 1; i <= np; ++i)
        Vscanf("%lf", &par[i - 1]);
    Vscanf("%ld", &ntype);
    if (ntype == 1)
        ntype_enum = Nag_TestStatisticsDAbs;
    else if (ntype == 2)
        ntype_enum = Nag_TestStatisticsDPos;
    else if (ntype == 3)
        ntype_enum = Nag_TestStatisticsDNeg;
    else
        ntype_enum = (Nag_TestStatistics)-999;

    go8cbc(n, x, Nag_Uniform, par, Nag_ParaSupplied, ntype_enum, &d, &z, &p, &fail);
    if (fail.code != NE_NOERROR)
        {
            Vprintf("Error from go8cbc.\n%s\n", fail.message);
            exit_status = 1;
            goto END;
        }

    Vprintf("Test against uniform distribution on (0,2)\n");
    Vprintf("\n");
    Vprintf("Test statistic D = %8.4f\n", d);
    Vprintf("Z statistic = %8.4f\n", z);
    Vprintf("Tail probability = %8.4f\n", p);
    Vprintf("\n");
    Vscanf("%ld", &np);
        for (i = 1; i <= np; ++i)
        Vscanf("%lf", &par[i - 1]);
        Vscanf("%ld", &ntype);
    if (ntype == 1)
        ntype_enum = Nag_TestStatisticsDAbs;
}
else if (ntype == 2)
    ntype_enum = Nag_TestStatisticsDPos;
else if (ntype == 3)
    ntype_enum = Nag_TestStatisticsDNeg;

g08cbc(n, x, Nag_Normal, par, Nag_ParaEstimated, ntype_enum, &d, &z, &p, &fail);
if (fail.code != NE_NOERROR)
{
    Vprintf("Error from g08cbc.\n\n", fail.message);
    exit_status = 1;
    goto END;
}

Vprintf("Test against Normal distribution with parameters estimated from the data\n");
Vprintf("\n");
Vprintf("%s\n", fail.message);
Vprintf("%s6.4f\n", Mean = " , par[0], " and variance = ", par[1]);
Vprintf("Test statistic D = %8.4f\n", d);
Vprintf("Z statistic = %8.4f\n", z);
Vprintf("Tail probability = %8.4f\n", p);
END:
if (x) NAG_FREE(x);
if (par) NAG_FREE(par);
return exit_status;

8.2 Program Data

g08cbc Example Program Data
30
0.01 0.30 0.20 0.90 1.20 0.09 1.30 0.18 0.90 0.48
1.98 0.03 0.50 0.07 0.70 0.60 0.95 1.00 0.31 1.45
1.04 1.25 0.15 0.75 0.85 0.22 1.56 0.81 0.57 0.55
2 0.0 2.0 1
2 0.0 1.0 1

8.3 Program Results

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