NAG Library Function Document

nag_rank_ci_2var (g07ebc)

1 Purpose
nag_rank_ci_2var (g07ebc) calculates a rank based (nonparametric) estimate and confidence interval for
the difference in location between two independent populations.

2 Specification
#include <nag.h>
#include <nagg07.h>

void nag_rank_ci_2var (Nag_RCIMethod method, Integer n, const double x[],
Integer m, const double y[], double clevel, double *theta,
double *thetal, double *thetau, double *estcl, double *ulower,
double *uupper, NagError *fail)

3 Description
Consider two random samples from two populations which have the same continuous distribution except
for a shift in the location. Let the random sample, $x = (x_1, x_2, \ldots, x_n)^T$, have distribution $F(x)$ and the
random sample, $y = (y_1, y_2, \ldots, y_m)^T$, have distribution $F(x - \theta)$.

nag_rank_ci_2var (g07ebc) finds a point estimate, $\hat{\theta}$, of the difference in location $\theta$ together with an
associated confidence interval. The estimates are based on the ordered differences $y_j - x_i$. The estimate $\hat{\theta}$
is defined by

$$\hat{\theta} = \text{median}\{y_j - x_i, \ i = 1, 2, \ldots, n; j = 1, 2, \ldots, m\}.$$  

Let $d_k$, for $k = 1, 2, \ldots, nm$, denote the $nm$ (ascendingly) ordered differences $y_j - x_i$, for $i = 1, 2, \ldots, n$
and $j = 1, 2, \ldots, m$. Then

- if $nm$ is odd, $\hat{\theta} = d_k$ where $k = (nm - 1)/2$;
- if $nm$ is even, $\hat{\theta} = (d_k + d_{k+1})/2$ where $k = nm/2$.

This estimator arises from inverting the two sample Mann–Whitney rank test statistic, $U(\theta_0)$, for testing
the hypothesis that $\theta = \theta_0$. Thus $U(\theta_0)$ is the value of the Mann–Whitney $U$ statistic for the two
independent samples $\{x_i + \theta_0\}$, for $i = 1, 2, \ldots, n$ and $\{y_j\}$, for $j = 1, 2, \ldots, m$. Effectively $U(\theta_0)$ is
a monotonically increasing step function of $\theta_0$ with

$$\text{mean}\ (U) = \mu = \frac{nm}{2},$$

$$\text{var}\ (U) = \sigma^2 \frac{nm(n + m + 1)}{12}.$$  

The estimate $\hat{\theta}$ is the solution to the equation $U(\hat{\theta}) = \mu$; two methods are available for solving this
equation. These methods avoid the computation of all the ordered differences $d_k$; this is because for large
$n$ and $m$ both the storage requirements and the computation time would be high.

The first is an exact method based on a set partitioning procedure on the set of all differences $y_j - x_i$, for
$i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$. This is adapted from the algorithm proposed by Monahan (1984) for
the computation of the Hodges–Lehmann estimator for a single population.

The second is an iterative algorithm, based on the Illinois method which is a modification of the \textit{regula falsi} method, see McKean and Ryan (1977). This algorithm has proved suitable for the function $U(\theta_0)$
which is asymptotically linear as a function of $\theta_0$. 

Mark 25
The confidence interval limits are also based on the inversion of the Mann–Whitney test statistic. Given a desired percentage for the confidence interval, \(1 - \alpha\), expressed as a proportion between 0.0 and 1.0 initial estimates of the upper and lower confidence limits for the Mann–Whitney \(U\) statistic are found;

\[
U_l = \mu - 0.5 + (\sigma \times \Phi^{-1}(\alpha/2)) \\
U_u = \mu + 0.5 + (\sigma \times \Phi^{-1}((1 - \alpha)/2))
\]

where \(\Phi^{-1}\) is the inverse cumulative Normal distribution function.

\(U_l\) and \(U_u\) are rounded to the nearest integer values. These estimates are refined using an exact method, without taking ties into account, if \(n + m \leq 40\) and \(\max(n, m) \leq 30\) and a Normal approximation otherwise, to find \(U_l\) and \(U_u\) satisfying

\[
\begin{align*}
P(U \leq U_l) &\leq \alpha/2 \\
P(U \leq U_l + 1) &> \alpha/2
\end{align*}
\]

and

\[
\begin{align*}
P(U \geq U_u) &\leq \alpha/2 \\
P(U \geq U_u - 1) &> \alpha/2.
\end{align*}
\]

The function \(U(\theta_0)\) is a monotonically increasing step function. It is the number of times a score in the second sample, \(y_j\), precedes a score in the first sample, \(x_i + \theta\), where we only count a half if a score in the second sample actually equals a score in the first.

Let \(U_l = k\); then \(\theta_l = d_{k+1}\). This is the largest value \(\theta_l\) such that \(U(\theta_l) = U_l\).

Let \(U_u = nm - k\); then \(\theta_u = d_{nm-k}\). This is the smallest value \(\theta_u\) such that \(U(\theta_u) = U_u\).

As in the case of \(\hat{\theta}\), these equations may be solved using either the exact or iterative methods to find the values \(\theta_l\) and \(\theta_u\).

Then \((\theta_l, \theta_u)\) is the confidence interval for \(\theta\). The confidence interval is thus defined by those values of \(\theta_0\) such that the null hypothesis, \(\theta = \theta_0\), is not rejected by the Mann–Whitney two sample rank test at the \((100 \times \alpha)\)% level.

### 4 References

Lehmann E L (1975) *Nonparametrics: Statistical Methods Based on Ranks* Holden–Day


### 5 Arguments

1. `method` – Nag_RCI\_Method

   *Input*

   *On entry:* specifies the method to be used.

   - `method = Nag_RCI\_Exact`
     - The exact algorithm is used.

   - `method = Nag_RCI\_Approx`
     - The iterative algorithm is used.

   *Constraint:* `method = Nag_RCI\_Exact` or `Nag_RCI\_Approx`.  

---

**g07ebc.2**

*Mark 25*
2: n – Integer
   *Input*
   
   *On entry:* \( n \), the size of the first sample.
   
   *Constraint:* \( n \geq 1 \).

3: x[n] – const double
   *Input*
   
   *On entry:* the observations of the first sample, \( x_i \), for \( i = 1, 2, \ldots, n \).

4: m – Integer
   *Input*
   
   *On entry:* \( m \), the size of the second sample.
   
   *Constraint:* \( m \geq 1 \).

5: y[m] – const double
   *Input*
   
   *On entry:* the observations of the second sample, \( y_j \), for \( j = 1, 2, \ldots, m \).

6: clevel – double
   *Input*
   
   *On entry:* the confidence interval required, \( 1 - \alpha \); e.g., for a 95% confidence interval set \( clevel = 0.95 \).
   
   *Constraint:* \( 0.0 < \text{clevel} < 1.0 \).

7: theta – double *
   *Output*
   
   *On exit:* the estimate of the difference in the location of the two populations, \( \hat{\theta} \).

8: theta1 – double *
   *Output*
   
   *On exit:* the estimate of the lower limit of the confidence interval, \( \theta_l \).

9: theta2 – double *
   *Output*
   
   *On exit:* the estimate of the upper limit of the confidence interval, \( \theta_u \).

10: estcl – double *
    *Output*
    
    *On exit:* an estimate of the actual percentage confidence of the interval found, as a proportion between \((0.0, 1.0)\).

11: ulower – double *
    *Output*
    
    *On exit:* the value of the Mann–Whitney \( U \) statistic corresponding to the lower confidence limit, \( U_l \).

12: uupper – double *
    *Output*
    
    *On exit:* the value of the Mann–Whitney \( U \) statistic corresponding to the upper confidence limit, \( U_u \).

13: 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 \(<\text{value}\)> had an illegal value.

**NE_CONVERGENCE**

Warning. The iterative procedure to find an estimate of the lower confidence limit has not converged in 100 iterations.

Warning. The iterative procedure to find an estimate of Theta has not converged in 100 iterations.

Warning. The iterative procedure to find an estimate of the upper confidence limit has not converged in 100 iterations.

**NE_INT**

On entry, \(m = \langle\text{value}\rangle\).

Constraint: \(m \geq 1\).

On entry, \(n = \langle\text{value}\rangle\).

Constraint: \(n \geq 1\).

**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.

**NE_REAL**

On entry, \(clevel = \langle\text{value}\rangle\).

**NE_SAMPLE_IDEN**

Not enough information to compute an interval estimate since each sample has identical values.

The common difference is returned in \(\text{theta}, \text{thetal}\) and \(\text{thetau}\).

**Accuracy**

\text{nag\_rank\_ci\_2var (g07ebc)} should return results accurate to five significant figures in the width of the confidence interval, that is the error for any one of the three estimates should be less than \(0.00001 \times (\text{thetau} - \text{thetal})\).

**Parallelism and Performance**

\text{nag\_rank\_ci\_2var (g07ebc)} is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

\text{nag\_rank\_ci\_2var (g07ebc)} makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

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 increases with the sample sizes $n$ and $m$.

10 Example
The following program calculates a 95% confidence interval for the difference in location between the two populations from which the two samples of sizes 50 and 100 are drawn respectively.

10.1 Program Text
/* nag_rank_ci_2var (g07ebc) Example Program.
 * Copyright 2014 Numerical Algorithms Group.
 */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg07.h>

int main(void)
{
    /* Scalars */
    double  clevel, estcl, theta, thetal, thetau, ulower, uupper;
    Integer  exit_status, i, m, n;
    NagError fail;
    
    /* Arrays */
    double  *wrk = 0, *x = 0, *y = 0;
    Integer  *iwrk = 0;

    INIT_FAIL(fail);
    exit_status = 0;
    printf("nag_rank_ci_2var (g07ebc) Example Program Results\n");

    /* Skip Heading in data file */
    #ifdef _WIN32
        scanf_s("%*[\n] %"NAG_IFMT"%"NAG_IFMT"%*[\n] ", &n, &m);
    #else
        scanf("%*[\n] %"NAG_IFMT"%"NAG_IFMT"%*[\n] ", &n, &m);
    #endif
    
    /* Allocate memory */
    if (!(wrk = NAG_ALLOC(600, double)) ||
        !(x = NAG_ALLOC(n, double)) ||
        !(y = NAG_ALLOC(m, double)) ||
        !(iwrk = NAG_ALLOC(300, integer)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }

    /* Skip data file */
    #ifdef _WIN32
        scanf_s("%*[\n] ");
    #else
        scanf("%*[\n] ");
    #endif

    for (i = 1; i <= n; ++i)
    {
        if (_WIN32)
            scanf_s("%lf", &x[i - 1]);
        else
            scanf("%lf", &x[i - 1]);
    }
```c
#ifdef _WIN32
    scanf_s(" %*[\n] ");
#else
    scanf(" %*[\n] ");
#endif

for (i = 1; i <= m; ++i)
#ifdef _WIN32
    scanf_s("%lf", &y[i - 1]);
#else
    scanf("%lf", &y[i - 1]);
#endif
#endif
#ifdef _WIN32
    scanf_s(" %*[\n] ");
#else
    scanf(" %*[\n] ");
#endif
#ifdef _WIN32
    scanf_s(" %lf%*[\n] ", &clevel);
#else
    scanf(" %lf%*[\n] ", &clevel);
#endif

nag_rank_ci_2var(Nag_RCI_Approx, n, x, m, y, clevel, &theta, &thetal, &thetau, &estcl, &ulower, &uupper, &fail);

if (fail.code != NE_NOERROR)
{
    printf("Error from nag_rank_ci_2var (g07ebc).\n%s\n", fail.message);
    exit_status = 1;
    goto END;
}

printf(" Location estimator  Confidence Interval\n");
printf("\n");
printf(" %10.4f  ( %6.4f , %6.4f )\n", theta, thetal, thetau);
printf("\n");
printf(" Corresponding Mann-Whitney U statistics\n");
printf("\n");
printf("  Lower : %8.2f\n  Upper : %8.2f\n", ulower, uupper);
END:
NAG_FREE(wrk);
NAG_FREE(x);
NAG_FREE(y);
NAG_FREE(iwrk);
return exit_status;

10.2 Program Data

nag_rank_ci_2var (g07ebc) Example Program Data
50 100
First sample of N observations
-0.582 0.157 -0.523 -0.769 2.338 1.664 -0.981 1.549 1.131 -0.460
-0.484 1.932 0.306 -0.602 -0.979 0.132 0.256 -0.094 1.065 -1.084
-0.969 -0.524 0.239 1.512 -0.782 -0.252 -1.163 1.376 1.674 0.831
1.478 -1.486 -0.608 -0.429 -2.002 0.482 -1.584 -0.105 0.429 0.568
0.944 2.558 -1.801 0.242 0.763 -0.461 -1.497 -1.353 0.301 1.941
Second sample of M observations
1.995 0.007 0.997 1.089 2.004 0.171 0.294 2.448 0.214 0.773
2.960 0.025 0.638 0.937 -0.568 -0.711 0.931 2.601 1.121 -0.251
-0.050 1.341 2.282 0.745 1.633 0.944 2.370 0.293 0.895 0.938
0.199 0.812 1.253 0.590 1.522 -0.685 1.259 0.571 1.579 0.568
0.381 0.829 0.277 0.656 2.497 1.779 1.922 -0.174 2.132 2.793
```
10.3 Program Results

nag_rank_ci_2var (g07ebc) Example Program Results

Location estimator        Confidence Interval
                           0.9505                 ( 0.5650 , 1.3050 )

Corresponding Mann-Whitney U statistics

Lower :  2007.00
Upper :  2993.00