

NAG Toolbox

nag_nonpar_randtest_pairs (g08eb)

1 Purpose

nag_nonpar_randtest_pairs (g08eb) performs a pairs test on a sequence of observations in the interval $[0, 1]$.

2 Syntax

```
[ncount, ex, chi, df, prob, wrk, ifail] = nag_nonpar_randtest_pairs(c1, x, lag, ncount, wrk, 'n', n, 'msize', msize)
```

```
[ncount, ex, chi, df, prob, wrk, ifail] = g08eb(c1, x, lag, ncount, wrk, 'n', n, 'msize', msize)
```

3 Description

nag_nonpar_randtest_pairs (g08eb) computes the statistics for performing a pairs test which may be used to investigate deviations from randomness in a sequence, $x = \{x_i : i = 1, 2, \dots, n\}$, of $[0, 1]$ observations.

For a given lag, $l \geq 1$, an m by m matrix, C , of counts is formed as follows. The element c_{jk} of C is the number of pairs (x_i, x_{i+l}) such that

$$\frac{j-1}{m} \leq x_i < \frac{j}{m}$$

$$\frac{k-1}{m} \leq x_{i+l} < \frac{k}{m}$$

where $i = 1, 3, 5, \dots, n-1$ if $l = 1$, and $i = 1, 2, \dots, l, 2l+1, 2l+2, \dots, 3l, 4l+1, \dots, n-l$, if $l > 1$.

Note that all pairs formed are non-overlapping pairs and are thus independent under the assumption of randomness.

Under the assumption that the sequence is random, the expected number of pairs for each class (i.e., each element of the matrix of counts) is the same; that is, the pairs should be uniformly distributed over the unit square $[0, 1]^2$. Thus the expected number of pairs for each class is just the total number of pairs,

$\sum_{j,k=1}^m c_{jk}$, divided by the number of classes, m^2 .

The χ^2 test statistic used to test the hypothesis of randomness is defined as

$$X^2 = \sum_{j,k=1}^m \frac{(c_{jk} - e)^2}{e},$$

where $e = \sum_{j,k=1}^m c_{jk} / m^2 =$ expected number of pairs in each class.

The use of the χ^2 -distribution as an approximation to the exact distribution of the test statistic, X^2 , improves as the length of the sequence relative to m increases and hence the expected value, e , increases.

nag_nonpar_randtest_pairs (g08eb) may be used in two different modes:

- (i) a single call to `nag_nonpar_randtest_pairs` (g08eb) which computes all test statistics after counting the pairs;
- (ii) multiple calls to `nag_nonpar_randtest_pairs` (g08eb) with the final test statistics only being computed in the last call.

The second mode is necessary if all the data do not fit into the memory. See argument `cl` in Section 5 for details on how to invoke each mode.

4 References

Dagpunar J (1988) *Principles of Random Variate Generation* Oxford University Press

Knuth D E (1981) *The Art of Computer Programming (Volume 2)* (2nd Edition) Addison–Wesley

Morgan B J T (1984) *Elements of Simulation* Chapman and Hall

Ripley B D (1987) *Stochastic Simulation* Wiley

5 Parameters

5.1 Compulsory Input Parameters

- 1: `cl` – CHARACTER(1)

Indicates the type of call to `nag_nonpar_randtest_pairs` (g08eb).

`cl` = 'S'

This is the one and only call to `nag_nonpar_randtest_pairs` (g08eb) (single call mode). All data are to be input at once. All test statistics are computed after the counting of pairs is complete.

`cl` = 'F'

This is the first call to the function. All initializations are carried out and the counting of pairs begins. The final test statistics are not computed since further calls will be made to `nag_nonpar_randtest_pairs` (g08eb).

`cl` = 'I'

This is an intermediate call during which the counts of pairs are updated. The final test statistics are not computed since further calls will be made to `nag_nonpar_randtest_pairs` (g08eb).

`cl` = 'L'

This is the last call to `nag_nonpar_randtest_pairs` (g08eb). The test statistics are computed after the final counting of runs is complete.

Constraint: `cl` = 'S', 'F', 'I' or 'L'.

- 2: `x(n)` – REAL (KIND=nag_wp) array

The sequence of observations.

Constraint: $0.0 \leq x(i) \leq 1.0$, for $i = 1, 2, \dots, n$.

- 3: `lag` – INTEGER

l , the lag to be used in choosing pairs.

If `lag` = 1, then we consider the pairs $(x(i), x(i+1))$, for $i = 1, 3, \dots, n-1$, where n is the number of observations.

If `lag` > 1, then we consider the pairs $(x(i), x(i+l))$, for $i = 1, 2, \dots, l, 2l+1, 2l+2, \dots, 3l, 4l+1, \dots, n-l$, where n is the number of observations. `lag` must not be changed between calls to `nag_nonpar_randtest_pairs` (g08eb).

Constraints:

lag ≥ 1 ;
if **cl** = 'S', **lag** $< n$.

4: **ncount**(*ldc*, **msize**) – INTEGER array

ldc, the first dimension of the array, must satisfy the constraint $ldc \geq \mathbf{msize}$.

If **cl** = 'S' or 'F', **ncount** need not be set.

If **cl** = 'I' or 'L', **ncount** must contain the values returned by the previous call to `nag_nonpar_randtest_pairs` (g08eb).

5: **wrk**($2 \times \mathbf{lag}$) – REAL (KIND=nag_wp) array

wrk is used to store information between successive calls to `nag_nonpar_randtest_pairs` (g08eb) and therefore must not be changed.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the array **x**.

n, the number of observations.

Constraints:

if **cl** = 'S', **n** ≥ 2 ;
otherwise **n** ≥ 1 .

2: **msize** – INTEGER

Default: the first dimension of the array **ncount** and the second dimension of the array **ncount**. (An error is raised if these dimensions are not equal.)

m, the size of the matrix of counts.

msize must not be changed between calls to `nag_nonpar_randtest_pairs` (g08eb).

Constraint: **msize** ≥ 2 .

5.3 Output Parameters

1: **ncount**(*ldc*, **msize**) – INTEGER array

Is an **msize** by **msize** matrix containing the counts of the number of pairs in each cell, c_{ij} , for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, m$.

2: **ex** – REAL (KIND=nag_wp)

If **cl** = 'S' or 'L' (i.e., if it is a final exit) then **ex** contains the expected number of counts in each cell, *e*.

Otherwise **ex** is not set.

3: **chi** – REAL (KIND=nag_wp)

If **cl** = 'S' or 'L' (i.e., if it is a final exit) then **chi** contains the χ^2 test statistic, X^2 , for testing the null hypothesis of randomness.

Otherwise **chi** is not set.

4: **df** – REAL (KIND=nag_wp)

If **cl** = 'S' or 'L' (i.e., if it is a final exit) then **df** contains the degrees of freedom for the χ^2 statistic.

Otherwise **df** is not set.

5: **prob** – REAL (KIND=nag_wp)

If **cl** = 'S' or 'L' (i.e., if it is a final exit) then **prob** contains the upper tail probability associated with the χ^2 test statistic, i.e., the significance level.

Otherwise **prob** is not set.

6: **wrk**($2 \times \mathbf{lag}$) – REAL (KIND=nag_wp) array

7: **ifail** – INTEGER

ifail = 0 unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Note: nag_nonpar_randtest_pairs (g08eb) may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the function:

ifail = 1

On entry, **cl** = *value*.

Constraint: **cl** = 'S', 'F', 'T' or 'L'.

ifail = 2

Constraint: if **cl** = 'S', $\mathbf{n} \geq 2$, otherwise $\mathbf{n} \geq 1$.

ifail = 3

Constraint: **msize** ≥ 2

ifail = 4

Constraint: **lag** > 0 and if **cl** = 'S', **lag** $< \mathbf{n}$.

ifail = 5

Constraint: *ldc* $\geq \mathbf{msize}$.

ifail = 6

On entry, at least one element of **x** is out of range.

Constraint: $0 \leq \mathbf{x}(i) \leq 1$, for $i = 1, 2, \dots, \mathbf{n}$.

ifail = 7

No pairs were found. This will occur if the value of **lag** is greater than or equal to the total number of observations.

ifail = 8 (*warning*)

msize is too large relative to the number of pairs, therefore the expected value for at least one cell is less than or equal to 5.0.

ifail = -99

An unexpected error has been triggered by this routine. Please contact NAG.

ifail = -399

Your licence key may have expired or may not have been installed correctly.

ifail = -999

Dynamic memory allocation failed.

7 Accuracy

The computations are believed to be stable. The computation of **prob** given the values of **chi** and **df** will obtain a relative accuracy of five significant figures for most cases.

8 Further Comments

If after forming the pairs in an initial or intermediate call to `nag_nonpar_randtest_pairs` (g08eb) there is an observation left over at the end of the sequence, this observation is used at the beginning of the new sequence provided by the following call to `nag_nonpar_randtest_pairs` (g08eb). Clearly an observation left over from an only or final call to `nag_nonpar_randtest_pairs` (g08eb) is ignored.

The time taken by the function increases with the number of observations n , and also depends to some extent on whether the call to `nag_nonpar_randtest_pairs` (g08eb) is an only, first, intermediate or last call.

9 Example

The following program performs the pairs test on 500 pseudorandom numbers. `nag_nonpar_randtest_pairs` (g08eb) is called 5 times with 100 observations on each call. **lag** = 1 is used and the pairs are tallied into a 5 by 5 matrix.

9.1 Program Text

```
function g08eb_example

fprintf('g08eb example results\n\n');

% Initialize the base generator to a repeatable sequence
seed = [nag_int(324213)];
genid = nag_int(1);
subid = nag_int(1);
[state, ifail] = g05kf( ...
                    genid, subid, seed);

lag = nag_int(1);
wrk = zeros(2*lag, 1);
m = 5;
ncount = zeros(m, m, nag_int_name);
n = nag_int(100);
nsampl = 5;
cl = 'F';

for i=1:nsampl
    % Generate a sample from U(0,1)
    [state, x, ifail] = g05sq( ...
                            n, 0, 1, state);

    % Process the sample
    [ncount, ex, chi, df, prob, wrk, ifail] = ...
    g08eb( ...
          cl, x, lag, ncount, wrk);

    % Adjust CL
```

```
cl = 'I';
if i==nsampl-1
    cl = 'L';
end
end

[ifail] = x04ea( ...
    'General', ' ', ncount, 'Count matrix');
fprintf('\nExpected value = %8.2f\n', ex);
fprintf('Chisq          = %10.4f\n', chi);
fprintf('DF            = %7.1f\n', df);
fprintf('Probability    = %10.4f\n', prob);
```

9.2 Program Results

g08eb example results

```
Count matrix
   1  2  3  4  5
1  7 10  5 16  8
2  9 10  7  6  8
3 13 15 10 10 12
4 10 21  7  5 13
5 13  5 10 12  8
```

```
Expected value =    10.00
Chisq          =   34.8000
DF            =    24.0
Probability    =    0.0714
```
