NAG Toolbox

nag_stat_normal_scores_var (g01dc)

1 Purpose

nag_stat_normal_scores_var (g01dc) computes an approximation to the variance-covariance matrix of an ordered set of independent observations from a Normal distribution with mean 0.0 and standard deviation 1.0.

2 Syntax

```
[vec, ifail] = nag_stat_normal_scores_var(n, exp1, exp2, sumssq)
[vec, ifail] = g01dc(n, exp1, exp2, sumssq)
```

3 Description

nag_stat_normal_scores_var (g01dc) is an adaptation of the Applied Statistics Algorithm AS 128, see Davis and Stephens (1978). An approximation to the variance-covariance matrix, V, using a Taylor series expansion of the Normal distribution function is discussed in David and Johnson (1954).

However, convergence is slow for extreme variances and covariances. The present function uses the David–Johnson approximation to provide an initial approximation and improves upon it by use of the following identities for the matrix.

For a sample of size n, let m_i be the expected value of the ith largest order statistic, then:

(a) for any $i = 1, 2, \ldots, n$, $\sum_{j=1}^{n} V_{ij} = 1$

(b) $V_{12} = V_{11} + m_n^2 - m_n m_{n-1} - 1$

(c) the trace of V is $tr(V) = n - \sum_{i=1}^{n} m_i^2$

(d) $V_{ij} = V_{ji} = V_{rs} = V_{sr}$ where $r = n + 1 - i$, $s = n + 1 - j$ and $i, j = 1, 2, \ldots, n$. Note that only the upper triangle of the matrix is calculated and returned column-wise in vector form.

4 References

David F N and Johnson N L (1954) Statistical treatment of censored data, Part 1. Fundamental formulae *Biometrika* **41** 228–240

Davis C S and Stephens M A (1978) Algorithm AS 128: approximating the covariance matrix of Normal order statistics *Appl. Statist.* **27** 206–212

5 Parameters

5.1 Compulsory Input Parameters

1: **n** – INTEGER

n, the sample size.

Constraint: **n** > 0.

2: **exp1** – REAL (KIND=nag_wp)

The expected value of the largest Normal order statistic, m_n, from a sample of size n.

3: **exp2** – REAL (KIND=nag_wp)

The expected value of the second largest Normal order statistic, m_{n-1}, from a sample of size n.

4: **sumssq** – REAL (KIND=nag_wp)

The sum of squares of the expected values of the Normal order statistics from a sample of size n.

5.2 Optional Input Parameters

None.

5.3 Output Parameters

1: $\mathbf{vec}(\mathbf{n} \times (\mathbf{n}+\mathbf{1})/\mathbf{2})$ – REAL (KIND=nag_wp) array

The upper triangle of the n by n variance-covariance matrix packed by column. Thus element V_{ij} is stored in $\mathbf{vec}(i + j \times (j-1)/2)$, for $1 \leq i \leq j \leq n$.

2: **ifail** – INTEGER

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail $= 1$

On entry, $\mathbf{n} < 1$.

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

For $n \leq 20$, where comparison with the exact values can be made, the maximum error is less than 0.0001.

8 Further Comments

The time taken by nag_stat_normal_scores_var (g01dc) is approximately proportional to n^2.

The arguments **exp1** $(= m_n)$, **exp2** $(= m_{n-1})$ and **sumssq** $(= \sum_{j=1}^{n} m_j^2)$ may be found from the expected values of the Normal order statistics obtained from nag_stat_normal_scores_exact (g01da) (exact) or nag_stat_normal_scores_approx (g01db) (approximate).

9 Example

A program to compute the variance-covariance matrix for a sample of size 6. nag_stat_normal_scores_exact (g01da) is called to provide values for **exp1**, **exp2** and **sumssq**.

9.1 Program Text

```
function g01dc_example

fprintf('g01dc example results\n\n');

n    = nag_int(6);
etol = 1e-4;

% Compute normal scores
[pp, errest, ifail] = g01da(n, etol);

sumssq = dot(pp,pp);
exp1   = pp(n);
exp2   = pp(n-1);

% Compute approximate variance-covariance matrix
[vec, ifail] = g01dc(n, exp1, exp2, sumssq);

fprintf('Sample size = %4d\n', n);
fprintf('Variance-covariance matrix\n');
k = 1;
for j = 1:n
  fprintf('%8.3f',vec(k:(k+j-1)));
  fprintf('\n');
  k = k + j;
end
```

9.2 Program Results

```
g01dc example results

Sample size =    6
Variance-covariance matrix
   0.416
   0.209   0.280
   0.139   0.189   0.246
   0.102   0.140   0.183   0.246
   0.077   0.106   0.140   0.189   0.280
   0.056   0.077   0.102   0.139   0.209   0.416
```