NAG Toolbox

nag_correg_linregm_service_select (g02ce)

1 Purpose

nag_correg_linregm_service_select (g02ce) takes selected elements from two vectors (typically vectors of means and standard deviations) to form two smaller vectors, and selected rows and columns from two matrices (typically either matrices of sums of squares and cross-products of deviations from means and Pearson product-moment correlation coefficients, or matrices of sums of squares and cross-products about zero and correlation-like coefficients) to form two smaller matrices, allowing reordering of elements in the process.

2 Syntax

```
[xbar2, std2, ssp2, r2, ifail] = nag_correg_linregm_service_select(xbar, std, ssp, r, korder, 'n', n, 'm', m)
[xbar2, std2, ssp2, r2, ifail] = g02ce(xbar, std, ssp, r, korder, 'n', n, 'm', m)
```

3 Description

Input to the function consists of:

(a) A vector of means:

$$(\bar{x}_1, \bar{x}_2, \bar{x}_3, \ldots, \bar{x}_n),$$

where n is the number of input variables.

(b) A vector of standard deviations:

$$(s_1, s_2, s_3, \ldots, s_n).$$

(c) A matrix of sums of squares and cross-products of deviations from means:

$$\begin{pmatrix} S_{11} & S_{12} & S_{13} & . & . & . & S_{1n} \\ S_{21} & S_{22} & & & & & S_{2n} \\ S_{31} & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ S_{n1} & S_{n2} & . & . & . & . & S_{nn} \end{pmatrix}.$$

(d) A matrix of correlation coefficients:

$$\begin{pmatrix} R_{11} & R_{12} & R_{13} & . & . & . & R_{1n} \\ R_{21} & R_{22} & & & & & R_{2n} \\ R_{31} & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ R_{n1} & R_{n2} & . & . & . & . & R_{nn} \end{pmatrix}.$$

(e) The number of variables, m, in the required subset, and their row/column numbers in the input data, $i_1, i_2, i_3, \ldots, i_m$,

$$i \leq i_k \leq n \quad \text{for } k = 1, 2, \ldots, m \quad (n \geq 2, m \geq 1 \text{ and } m \leq n).$$

New vectors and matrices are output containing the following information:

(i) A vector of means:

$$(\bar{x}_{i_1}, \bar{x}_{i_2}, \bar{x}_{i_3}, \ldots, \bar{x}_{i_m}).$$

(ii) A vector of standard deviations:

$$(s_{i_1}, s_{i_2}, s_{i_3}, \ldots, s_{i_m}).$$

(iii) A matrix of sums of squares and cross-products of deviations from means:

$$\begin{pmatrix} S_{i_1i_1} & S_{i_1i_2} & S_{i_1i_3} & . & . & . & S_{i_1i_m} \\ S_{i_2i_1} & S_{i_2i_2} & & & & & . \\ S_{i_3i_1} & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ S_{i_mi_1} & S_{i_mi_2} & . & . & . & . & S_{i_mi_m} \end{pmatrix}.$$

(iv) A matrix of correlation coefficients:

$$\begin{pmatrix} R_{i_1i_1} & R_{i_1i_2} & R_{i_1i_3} & . & . & . & R_{i_1i_m} \\ R_{i_2i_1} & R_{i_2i_2} & & & & & . \\ R_{i_3i_1} & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ . & & & & & & . \\ R_{i_mi_1} & R_{i_mi_2} & . & . & . & . & R_{i_mi_m} \end{pmatrix}.$$

Note: for sums of squares of cross-products of deviations about zero and correlation-like coefficients S_{ij} and R_{ij} should be replaced by $\tilde{S}_{ij}$ and $\tilde{R}_{ij}$ in the description of the input and output above.

4 References

None.

5 Parameters

5.1 Compulsory Input Parameters

1: **xbar**(**n**) – REAL (KIND=nag_wp) array

xbar(i) must be set to $\bar{x}_i$, the mean of variable i, for $i = 1, 2, \ldots, n$.

2: **std**(**n**) – REAL (KIND=nag_wp) array

std(i) must be set to s_i, the standard deviation of variable i, for $i = 1, 2, \ldots, n$.

3: **ssp**($ldssp$, **n**) – REAL (KIND=nag_wp) array

$ldssp$, the first dimension of the array, must satisfy the constraint $ldssp \geq$ **n**.

ssp(i, j) must be set to the sum of cross-products of deviations from means S_{ij} (or about zero, $\tilde{S}_{ij}$) for variables i and j, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$.

4: **r**(ldr, **n**) – REAL (KIND=nag_wp) array

ldr, the first dimension of the array, must satisfy the constraint $ldr \geq$ **n**.

r(i, j) must be set to the Pearson product-moment correlation coefficient R_{ij} (or the correlation-like coefficient, $\tilde{R}_{ij}$) for variables i and j, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, n$.

5: **korder**(**m**) – INTEGER array

korder(i) must be set to the number of the original variable which is to be the ith variable in the output vectors and matrices, for $i = 1, 2, \ldots, m$.

Constraint: $1 \leq$ **korder**(i) $\leq$ **n**, for $i = 1, 2, \ldots, m$.

5.2 Optional Input Parameters

1: **n** – INTEGER

Default: the dimension of the arrays **xbar**, **std** and the first dimension of the arrays **ssp**, **r** and the second dimension of the arrays **ssp**, **r**. (An error is raised if these dimensions are not equal.)

n, the number of variables in the input data.

Constraint: **n** ≥ 2.

2: **m** – INTEGER

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

The number of variables m, required in the reduced vectors and matrices.

Constraint: $1 \leq$ **m** $\leq$ **n**.

5.3 Output Parameters

1: **xbar2**(**m**) – REAL (KIND=nag_wp) array

The mean of variable i, **xbar**(i), where $i =$ **korder**(k), for $k = 1, 2, \ldots, m$. (The array **xbar2** must differ from **xbar** and **std**.)

2: **std2**(**m**) – REAL (KIND=nag_wp) array

The standard deviation of variable i, **std**(i), where $i =$ **korder**(k), for $k = 1, 2, \ldots, m$. (The array **std2** must differ from both **xbar** and **std**.)

3: **ssp2**($ldssp2$, **m**) – REAL (KIND=nag_wp) array

ssp2(k, l) contains the value of **ssp**(i, j), where $i =$ **korder**(k) and $j =$ **korder**(l), for $k = 1, 2, \ldots, m$ and $l = 1, 2, \ldots, m$. (The array **ssp2** must differ from both **ssp** and **r**.)

That is to say: on exit, **ssp2**(k, l) contains the sum of cross-products of deviations from means S_{ij} (or about zero, $\tilde{S}_{ij}$).

4: **r2**($ldr2$, **m**) – REAL (KIND=nag_wp) array

r2(k, l) contains the value of **r**(i, j), where $i =$ **korder**(k) and $j =$ **korder**(l), for $k = 1, 2, \ldots, m$ and $l = 1, 2, \ldots, m$. (The array **r2** must differ from both **ssp** and **r**.)

That is to say: on exit, **r2**(k, l) contains the Pearson product-moment coefficient R_{ij} (or the correlation-like coefficient, $\tilde{R}_{ij}$).

5: **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, **n** < 2,
or **m** < 1.

ifail $= 2$

On entry, **n** $<$ **m**.

ifail $= 3$

On entry, $ldssp <$ **n**,
or $ldr <$ **n**,
or $ldssp <$ **m**,
or $ldr2 <$ **m**.

ifail $= 4$

On entry, **korder**$(i) < 1$,
or **korder**$(i) >$ **n** for some $i = 1, 2, \ldots, m$.

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

Not applicable.

8 Further Comments

The time taken by nag_correg_linregm_service_select (g02ce) depends on n and m.

The function is intended primarily for use when a subset of variables from a larger set of variables is to be used in a regression, and is described accordingly. There is however no reason why the function should not also be used to select specific rows and columns from vectors and arrays which contain any other non-statistical information; the matrices need not be symmetric.

The function may be used either with sums of squares and cross-products of deviations from means and Pearson product-moment correlation coefficients in connection with a regression involving a constant, or with sums of squares and cross-products about zero and correlation-like coefficients in connection with a regression with no constant.

9 Example

This example reads in the means, standard deviations, sums of squares and cross-products, and correlation coefficients for four variables. New vectors and matrices are created containing the means, standard deviations, sums of squares and cross-products, and correlation coefficients for the fourth, first and second variables (in that order). Finally these new vectors and matrices are printed.

9.1 Program Text

```
     function g02ce_example

fprintf('g02ce example results\n\n');

% Data
xbar = [  5.8;        2.8;       1.8;        5.4   ];
std  = [  5.0695;     1.924;     2.5884;     4.98  ];
ssp  = [102.8,      -29.2,     -14.2,      -57.6;
        -29.2,       14.8,      -6.2,        6.4;
        -14.2,       -6.2,      28.6,       42.4;
        -57.6,        6.4,      42.4,       99.2   ];
r    = [  1,         -0.7486,   -0.2619,    -0.5704;
          -0.7486,     1,        -0.3014,     0.167;
          -0.2619,    -0.3014,    1,          0.796;
          -0.5704,     0.167,     0.796,      1     ];

% Reordering
korder = [nag_int(4); 1; 2];

% Display data
fprintf('Original vector xbar :   ');
fprintf(' %10.4f', xbar);
fprintf('\n\nOriginal vector std  :   ');
fprintf(' %10.4f', std);
fprintf('\n\n');
disp('Original matrix SSP  :');
disp(ssp);
disp('Original matrix R    :');
disp(r);

% Calculate summaries on reduced number of variables
[xbar2, std2, ssp2, r2, ifail] = ...
  g02ce( ...
         xbar, std, ssp, r, korder);

% Display results
fprintf('New vector xbar2 :   ');
fprintf(' %10.4f', xbar2);
fprintf('\n\nNew vector std2  :   ');
fprintf(' %10.4f', std2);
fprintf('\n\n');
disp('New matrix ssp2  :');
disp(ssp2);
disp('New matrix r2    :');
disp(r2);
```

9.2 Program Results

```
     g02ce example results

Original vector xbar :        5.8000     2.8000     1.8000     5.4000

Original vector std  :        5.0695     1.9240     2.5884     4.9800

Original matrix SSP  :
  102.8000  -29.2000  -14.2000  -57.6000
  -29.2000   14.8000   -6.2000    6.4000
  -14.2000   -6.2000   28.6000   42.4000
  -57.6000    6.4000   42.4000   99.2000

Original matrix R    :
    1.0000   -0.7486   -0.2619   -0.5704
   -0.7486    1.0000   -0.3014    0.1670
   -0.2619   -0.3014    1.0000    0.7960
   -0.5704    0.1670    0.7960    1.0000

New vector xbar2 :        5.4000     5.8000     2.8000
```

```
New vector std2  :        4.9800     5.0695     1.9240

New matrix ssp2  :
   99.2000  -57.6000    6.4000
  -57.6000  102.8000  -29.2000
    6.4000  -29.2000   14.8000

New matrix r2    :
    1.0000   -0.5704    0.1670
   -0.5704    1.0000   -0.7486
    0.1670   -0.7486    1.0000
```