NAG Toolbox

nag_correg_ssqmat_combine (g02bz)

1 Purpose

nag_correg_ssqmat_combine (g02bz) combines two sets of sample means and sums of squares and cross-products matrices. It is designed to be used in conjunction with nag_correg_ssqmat (g02bu) to allow large datasets to be summarised.

2 Syntax

```
[xsw, xmean, xc, ifail] = nag_correg_ssqmat_combine(xsw, xmean, xc, ysw, ymean, yc, 'mean', mean, 'm', m)
[xsw, xmean, xc, ifail] = g02bz(xsw, xmean, xc, ysw, ymean, yc, 'mean', mean, 'm', m)
```

3 Description

Let X and Y denote two sets of data, each with m variables and n_x and n_y observations respectively. Let μ_x denote the (optionally weighted) vector of m means for the first dataset and C_x denote either the sums of squares and cross-products of deviations from μ_x

$$C_x = \left(X - e\mu_x^{\mathrm{T}}\right)^{\mathrm{T}} D_x \left(X - e\mu_x^{\mathrm{T}}\right)$$

or the sums of squares and cross-products, in which case

$$C_x = X^{\mathrm{T}} D_x X$$

where e is a vector of n_x ones and D_x is a diagonal matrix of (optional) weights and W_x is defined as the sum of the diagonal elements of D. Similarly, let μ_y, C_y and W_y denote the same quantities for the second dataset.

Given $\mu_x, \mu_y, C_x, C_y, W_x$ and W_y nag_correg_ssqmat_combine (g02bz) calculates μ_z, C_z and W_z as if a dataset Z, with m variables and $n_x + n_y$ observations were supplied to nag_correg_ssqmat (g02bu), with Z constructed as

$$Z = \begin{pmatrix} X \\ Y \end{pmatrix}.$$

nag_correg_ssqmat_combine (g02bz) has been designed to combine the results from two calls to nag_correg_ssqmat (g02bu) allowing large datasets, or cases where all the data is not available at the same time, to be summarised.

4 References

Bennett J, Pebay P, Roe D and Thompson D (2009) Numerically stable, single-pass, parallel statistics algorithms *Proceedings of IEEE International Conference on Cluster Computing*

5 Parameters

5.1 Compulsory Input Parameters

1: **xsw** – REAL (KIND=nag_wp)

W_x, the sum of weights, from the first set of data, X. If the data is unweighted then this will be the number of observations in the first dataset.

Constraint: **xsw** ≥ 0.

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

μ_x, the sample means for the first set of data, X.

3: **xc**((**m** × **m** + **m**)/**2**) – REAL (KIND=nag_wp) array

C_x, the sums of squares and cross-products matrix for the first set of data, X, as returned by nag_correg_ssqmat (g02bu).

nag_correg_ssqmat (g02bu), returns this matrix packed by columns, i.e., the cross-product between the jth and kth variable, $k \geq j$, is stored in **xc**$(k \times (k-1)/2 + j)$.

No check is made that C_x is a valid cross-products matrix.

4: **ysw** – REAL (KIND=nag_wp)

W_y, the sum of weights, from the second set of data, Y. If the data is unweighted then this will be the number of observations in the second dataset.

Constraint: **ysw** ≥ 0.

5: **ymean**(**m**) – REAL (KIND=nag_wp) array

μ_y, the sample means for the second set of data, Y.

6: **yc**((**m** × **m** + **m**)/**2**) – REAL (KIND=nag_wp) array

C_y, the sums of squares and cross-products matrix for the second set of data, Y, as returned by nag_correg_ssqmat (g02bu).

nag_correg_ssqmat (g02bu), returns this matrix packed by columns, i.e., the cross-product between the jth and kth variable, $k \geq j$, is stored in **yc**$(k \times (k-1)/2 + j)$.

No check is made that C_y is a valid cross-products matrix.

5.2 Optional Input Parameters

1: **mean_p** – CHARACTER(1)

Default: 'M'

Indicates whether the matrices supplied in **xc** and **yc** are sums of squares and cross-products, or sums of squares and cross-products of deviations about the mean.

mean = 'M'
Sums of squares and cross-products of deviations about the mean have been supplied.

mean = 'Z'
Sums of squares and cross-products have been supplied.

Constraint: **mean** = 'M' or 'Z'.

2: **m** – INTEGER

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

m, the number of variables.

Constraint: **m** ≥ 1.

5.3 Output Parameters

1: **xsw** – REAL (KIND=nag_wp)

W_z, the sum of weights, from the combined dataset, Z. If both datasets are unweighted then this will be the number of observations in the combined dataset.

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

μ_z, the sample means for the combined data, Z.

3: **xc**((**m** × **m** + **m**)/**2**) – REAL (KIND=nag_wp) array

C_z, the sums of squares and cross-products matrix for the combined dataset, Z.

This matrix is again stored packed by columns.

4: **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 = 11

On entry, **mean** = ⟨*value*⟩ was an illegal value.

ifail = 21

Constraint: **m** ≥ 1.

ifail = 31

Constraint: **xsw** ≥ 0.0.

ifail = 61

Constraint: **ysw** ≥ 0.0.

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

Not applicable.

8 Further Comments

None.

9 Example

This example illustrates the use of nag_correg_ssqmat_combine (g02bz) by dividing a dataset into three blocks of 4, 5 and 3 observations respectively. Each block of data is summarised using nag_correg_ssqmat (g02bu) and then the three summaries combined using nag_correg_ssqmat_combine (g02bz).

The resulting sums of squares and cross-products matrix is then scaled to obtain the covariance matrix for the whole dataset.

9.1 Program Text

```
     function g02bz_example

fprintf('g02bz example results\n\n');

x1 = [-1.10  4.06  -0.95  8.53 10.41;
       1.63 -3.22  -1.15 -1.30  3.78;
      -2.23 -8.19  -3.50  4.31 -1.11;
       0.92  0.33  -1.60  5.80 -1.15];

x2 = [ 2.12  5.00 -11.69 -1.22  2.86;
       4.82 -7.23  -4.67  0.83  3.46;
      -0.51 -1.12  -1.76  1.45  0.26;
      -4.32  4.89   1.34 -1.12 -2.49;
       0.02 -0.74   0.94 -0.99 -2.61];

wt = [ 2;    0.89;  0.32; 4.19; 4.33];

x3 = [ 1.37  0.00  -0.53 -7.98  3.32;
       4.15 -2.81  -4.09 -7.96 -2.13;
      13.09 -1.43   5.16 -1.83  1.58];

for b=1:3

  switch b
    case 1
      % first data block: summarise the data into xmean and xc
      [xsw, xmean, xc, ifail] = g02bu( ...
                                       x1);
    case 2
      [ysw, ymean, yc, ifail] = g02bu( ...
                                       x2, 'wt', wt);
    case 3
      [ysw, ymean, yc, ifail] = g02bu( ...
                                       x3);
  end

  if b ~= 1
    % Update the running summaries
    [xsw, xmean, xc, ifail] = g02bz( ...
                                     xsw, xmean, xc, ysw, ymean, yc);
  end
end

% Display results
fprintf('\nMeans\n');
disp(xmean');
mtitle = 'Sums of squares and cross-products';
uplo = 'Upper';
diag = 'Non-unit';
m = nag_int(5);
[ifail] = x04cc( ...
                 uplo, diag, m, xc, mtitle);

if xsw > 1
  % convert to covariance matrix
  fprintf('\n');
  mtitle = 'Covariance Matrix';
  [ifail] = x04cc( ...
                   uplo, diag, m, xc/(xsw-1), mtitle);
end
```

9.2 Program Results

```
     g02bz example results

Means
    0.4369    0.4929   -1.3387   -0.5684    0.0987

 Sums of squares and cross-products
             1          2          3          4          5
 1    304.5052  -123.7700   -27.1830   -60.7092    83.4830
 2              298.9148   -17.3196    -2.1710     5.2072
 3                         332.1639    -3.9445   -96.9299
 4                                     264.7684    79.6211
 5                                                225.5948

 Covariance Matrix
             1          2          3          4          5
 1     17.1746    -6.9808    -1.5332    -3.4241     4.7086
 2                16.8593    -0.9769    -0.1224     0.2937
 3                           18.7346    -0.2225    -5.4670
 4                                      14.9334     4.4908
 5                                                 12.7239
```