1. **Purpose**

*nag_corr_cov (g02bxc)* calculates the Pearson product-moment correlation coefficients and the variance-covariance matrix for a set of data. Weights may be used.

2. **Specification**

```c
#include <nag.h>
#include <nagg02.h>

void nag_corr_cov(Integer n, Integer m, double x[], Integer tdx,
                   Integer sx[], double wt[], double *sw, double wmean[],
                   double std[], double r[], Integer tdr, double v[],
                   Integer tdv, NagError *fail)
```

3. **Description**

For *n* observations on *m* variables a one-pass updating algorithm (see West 1979) is used to compute the means, the standard deviations, the variance-covariance matrix, and the Pearson product-moment correlation matrix for *p* selected variables. Suitable weights may be used to indicate multiple observations and to remove missing values.

The quantities are defined by:

(a) The means

\[
\bar{x}_j = \frac{\sum_{i=1}^{n} w_i x_{ij}}{\sum_{i=1}^{n} w_i} \quad j = 1, \ldots, p
\]

(b) The variance-covariance matrix

\[
C_{jk} = \frac{\sum_{i=1}^{n} w_i (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sum_{i=1}^{n} w_i - 1} \quad j, k = 1, \ldots, p
\]

(c) The standard deviations

\[
s_j = \sqrt{C_{jj}} \quad j = 1, \ldots, p
\]

(d) The Pearson product-moment correlation coefficients

\[
R_{jk} = \frac{C_{jk}}{\sqrt{C_{jj}C_{kk}}} \quad j, k = 1, \ldots, p
\]

where *x*<sub>ij</sub> is the value of the *i*th observation on the *j*th variable and *w*<sub>i</sub> is the weight for the *i*th observation which will be 1 in the unweighted case.

Note that the denominator for the variance-covariance is \(\sum_{i=1}^{n} w_i - 1\), so the weights should be scaled so that the sum of weights reflects the true sample size.

4. **Parameters**

- **n**
  - Input: the number of observations in the data set, *n*.
  - Constraint: *n* > 1.

- **m**
  - Input: the total number of variables, *m*.
  - Constraint: *m* ≥ 1.
NAG C Library Manual

\textbf{nag\_corr\_cov}

\begin{verbatim}
x[n][tdx]
  Input: the data \( x[i-1][j-1] \) must contain the \( i \)th observation on the \( j \)th variable, \( x_{ij} \), for 
  \( i = 1, \ldots, n \); \( j = 1, \ldots, m \).

  tdx
  Input: the second dimension of the array \( x \) as declared in the function from which nag\_corr\_cov
  is called.
  Constraint: \( tdx \geq m \).

sx[m]
  Input: indicates which \( p \) variables to include in the analysis.
  If \( sx[j-1] > 0 \), the \( j \)th variable is to be included.
  If \( sx[j-1] = 0 \), the \( j \)th variable is not to be included.
  If \( sx \) is set to the null pointer (Integer *)0 then all variables are included in the analysis, i.e.,
  \( p = m \).
  Constraint: \( sx[i] \geq 0 \), for \( i = 1, \ldots, m \).

wt[n]
  Input: the optional frequency weighting for each observation. \( wt[i-1] \) contains the weight for
  the \( i \)th data value. Usually \( wt[i-1] \) will be an integral value corresponding to the number of
  observations associated with the \( i \)th data value, or zero if the \( i \)th data value is to be ignored.
  If \( wt \) is set to the null pointer (double *)0 then \( wt \) is not referenced.
  Constraint: \( wt[i-1] \geq 0.0 \), for \( i = 1, \ldots, n \).

sw
  Output: the sum of weights if \( wt \) is not the null pointer, otherwise \( sw \) contains the number
  of observations, \( n \).

wmean[m]
  Output: the sample means. \( wmean[j-1] \) contains the mean for the \( j \)th variable.

std[m]
  Output: the standard deviations. \( std[j-1] \) contains the standard deviation for the \( j \)th
  variable.

r[m][tdr]
  Output: the matrix of Pearson product-moment correlation coefficients. \( r[j-1][k-1] \) contains
  the correlation between variables \( j \) and \( k \), for \( j, k = 1, \ldots, p \).

tdr
  Input: the second dimension of the array \( r \) as declared in the function from which nag\_corr\_cov
  is called.
  Constraint: \( tdr \geq m \).

v[m][tdv]
  Output: the variance-covariance matrix. \( v[j-1][k-1] \) contains the covariance between
  variables \( j \) and \( k \), for \( j, k = 1, \ldots, p \).

tdv
  Input: the second dimension of the array \( r \) as declared in the function from which nag\_corr\_cov
  is called.
  Constraint: \( tdv \geq m \).

fail
  The NAG error parameter, see the Essential Introduction to the NAG C Library.

5. Error Indications and Warnings

\textbf{NE\_INT\_ARG\_LE}
  On entry, \( n \) must be greater than 1: \( n = \langle value \rangle \).

\textbf{NE\_INT\_ARG\_LT}
  On entry, \( m \) must not be less than 1: \( m = \langle value \rangle \).
\end{verbatim}
6. Further Comments

Correlation coefficients based on ranks can be computed using nag_kendall_coeffs (g02brc).

6.1. Accuracy


6.2. References


7. See Also

nag_kendall_coeffs (g02brc)

8. Example

A program to calculate the means, standard deviations, variance-covariance matrix and a matrix of Pearson product-moment correlation coefficients for a set of 3 observations of 3 variables.

8.1. Program Text

```c
/* nag_corr_cov(g02bxc) Example Program
 */

#include <nag.h>
#include <stdio.h>
#include <nagg02.h>
```

#define NMAX 5
#define MMAX 5
#define TDX MMAX
#define TDV MMAX
#define TDR MMAX

main()
{
    double x[NMAX][TDX], r[MMAX][TDR], v[MMAX][TDV];
    double wt[NMAX], *wtptr;
    double sw, wmean[MMAX], std[MMAX];
    Integer i, j, n, m;
    char w;
    Integer tdx, tdr, tdv;
    Vprintf("g02bxc Example Program Results\n");
    /* Skip heading in data file */
    Vscanf("%*[\n]");
    tdx = TDX;
    tdr = TDR;
    tdv = TDV;
    test = 0;

    while ((scanf("%ld%ld %c", &m, &n, &w) != EOF))
    {
        if (m>=1 && m<=MMAX && n>=1 && n<=NMAX)
        {
            for(i=0; i<n; i++)
                Vscanf("%lf", &wt[i]);
            for(i=0; i<n; i++)
                for(j=0; j<m; j++)
                    Vscanf("%lf", &x[i][j]);
            if (w == 'w')
                wtptr = wt;
            else
                wtptr = (double *)0;
            g02bxc(n, m, (double *)x, tdx, (Integer *)0, wtptr, &sw, wmean, std,
                (double *)r, tdr, (double *)v, tdv, NAGERR_DEFAULT);
            if (wtptr)
                Vprintf("Case %ld --- Using weights\n", ++test);
            else
                Vprintf("Case %ld --- Not using weights\n", ++test);
            Vprintf ("Input data\n");
            for(i=0; i<n; i++)
                Vprintf("%6.1f%6.1f%6.1f%6.1f\n", x[i][0], x[i][1], x[i][2], wt[i]);
            Vprintf("\n");
            Vprintf("Sample means.\n");
            for(i=0; i<m; i++)
                Vprintf("%6.1f\n", wmean[i]);
            Vprintf("Standard deviation.\n");
            for(i=0; i<m; i++)
                Vprintf("%6.1f\n", std[i]);
            Vprintf("Correlation matrix.\n");
            for(i=0; i<m; i++)
            {
                for(j=0; j<m; j++)
                    Vprintf("%7.4f ", r[i][j]);
                Vprintf("\n");
            }
            Vprintf("Variance matrix.\n");
            for(i=0; i<m; i++)
            {
            }
        }
    }
}
for(j=0; j<m; j++)
    Vprintf(" %7.3f ", v[i][j]);
    Vprintf("\n");
}
Vprintf("\nSum of weights %6.1f\n", sw);
else
{
    Vfprintf(stderr, "One or both of m and n are out of range:\n    m = %-3ld while n = %-3ld\n", m, n);
    exit(EXIT_FAILURE);
}
exit(EXIT_SUCCESS);

8.2. Program Data

g02bxc Example Program Data
3 3 w
  9.1231  3.7011  4.5230
  0.9310  0.0900  0.8870
  0.0009  0.0099  0.0999
  0.1300  1.3070  0.3700

3 3 w
  0.1300  1.3070  0.3700
  9.1231  3.7011  4.5230
  0.9310  0.0900  0.8870
  0.0009  0.0099  0.0999

3 3 u
  0.717  9.370  0.013
  1.119  0.133  9.700
  11.100 23.510 11.117
  0.900  9.013  8.710

3 3 w
  0.717  19.370  0.013
  1.119  0.133  9.700
  11.100 23.510 11.117
  0.900  9.013  78.710

3 3 u
  0.717  19.370  0.013
  1.119  0.133  9.700
  11.100 3.510 13.117
  0.900  0.013  78.710

3 3 w
  0.717  19.370  0.913
  1.119  0.133  9.700
  17.100 93.510 13.117
  30.900  0.013  78.710

8.3. Program Results

g02bxc Example Program Results

Case 1 --- Using weights

Input data
  0.9  0.1  0.9  9.1
  0.0  0.0  0.1  3.7
  0.1  1.3  0.4  4.5

Sample means.
  0.5
  0.4
  0.6
Standard deviation.
0.4
0.6
0.3

Correlation matrix.
\[
\begin{array}{ccc}
1.0000 & -0.4932 & 0.9839 \\
-0.4932 & 1.0000 & -0.3298 \\
0.9839 & -0.3298 & 1.0000 \\
\end{array}
\]

Variance matrix.
\[
\begin{array}{ccc}
0.197 & -0.123 & 0.149 \\
-0.123 & 0.316 & -0.063 \\
0.149 & -0.063 & 0.117 \\
\end{array}
\]

Sum of weights 17.3

Case 2 --- Using weights

Input data
\[
\begin{array}{cccc}
9.1 & 3.7 & 4.5 & 0.1 \\
0.9 & 0.1 & 0.9 & 1.3 \\
0.0 & 0.0 & 0.1 & 0.4 \\
\end{array}
\]

Sample means.
1.3
0.3
1.0

Standard deviation.
3.3
1.4
1.5

Correlation matrix.
\[
\begin{array}{ccc}
1.0000 & 0.9908 & 0.9903 \\
0.9908 & 1.0000 & 0.9624 \\
0.9903 & 0.9624 & 1.0000 \\
\end{array}
\]

Variance matrix.
\[
\begin{array}{ccc}
10.851 & 4.582 & 5.044 \\
4.582 & 1.971 & 2.089 \\
5.044 & 2.089 & 2.391 \\
\end{array}
\]

Sum of weights 1.8

Case 3 --- Not using weights

Input data
\[
\begin{array}{cccc}
1.1 & 0.1 & 9.7 & 0.7 \\
11.1 & 23.5 & 11.1 & 9.4 \\
0.9 & 9.0 & 8.7 & 0.0 \\
\end{array}
\]

Sample means.
4.4
10.9
9.8

Standard deviation.
5.8
11.8
1.2

Correlation matrix.
\[
\begin{array}{ccc}
1.0000 & 0.9193 & 0.9200 \\
0.9193 & 1.0000 & 0.6915 \\
0.9200 & 0.6915 & 1.0000 \\
\end{array}
\]
Variance matrix.
\[
\begin{bmatrix}
33.951 & 63.208 & 6.485 \\
63.208 & 139.250 & 9.871 \\
6.485 & 9.871 & 1.464 \\
\end{bmatrix}
\]

Sum of weights 3.0

Case 4 --- Using weights

Input data
\[
\begin{bmatrix}
1.1 & 0.1 & 9.7 & 0.7 \\
11.1 & 23.5 & 11.1 & 19.4 \\
0.9 & 9.0 & 78.7 & 0.0 \\
\end{bmatrix}
\]

Sample means.
\[
\begin{bmatrix}
10.7 \\
22.7 \\
11.1 \\
\end{bmatrix}
\]

Standard deviation.
\[
\begin{bmatrix}
1.9 \\
4.5 \\
1.8 \\
\end{bmatrix}
\]

Correlation matrix.
\[
\begin{bmatrix}
1.0000 & 0.9985 & 0.0173 \\
0.9985 & 1.0000 & 0.0716 \\
0.0173 & 0.0716 & 1.0000 \\
\end{bmatrix}
\]

Variance matrix.
\[
\begin{bmatrix}
3.672 & 8.538 & 0.059 \\
8.538 & 19.909 & 0.570 \\
0.059 & 0.570 & 3.185 \\
\end{bmatrix}
\]

Sum of weights 20.1

Case 5 --- Not using weights

Input data
\[
\begin{bmatrix}
1.1 & 0.1 & 9.7 & 0.7 \\
11.1 & 3.5 & 13.1 & 19.4 \\
0.9 & 0.0 & 78.7 & 0.0 \\
\end{bmatrix}
\]

Sample means.
\[
\begin{bmatrix}
4.4 \\
1.2 \\
33.8 \\
\end{bmatrix}
\]

Standard deviation.
\[
\begin{bmatrix}
5.8 \\
2.0 \\
38.9 \\
\end{bmatrix}
\]

Correlation matrix.
\[
\begin{bmatrix}
1.0000 & 0.9999 & -0.4781 \\
0.9999 & 1.0000 & -0.4881 \\
-0.4781 & -0.4881 & 1.0000 \\
\end{bmatrix}
\]

Variance matrix.
\[
\begin{bmatrix}
33.951 & 11.567 & -108.343 \\
11.567 & 3.941 & -37.687 \\
-108.343 & -37.687 & 1512.750 \\
\end{bmatrix}
\]

Sum of weights 3.0
Case 6 --- Using weights

Input data

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>0.1</td>
<td>9.7</td>
<td>0.7</td>
</tr>
<tr>
<td>17.1</td>
<td>93.5</td>
<td>13.1</td>
<td>19.4</td>
</tr>
<tr>
<td>30.9</td>
<td>0.0</td>
<td>78.7</td>
<td>0.9</td>
</tr>
</tbody>
</table>

Sample means.
17.2
86.3
15.9

Standard deviation.
4.2
25.6
13.7

Correlation matrix.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>-0.0461</td>
<td>0.7426</td>
</tr>
<tr>
<td>-0.0461</td>
<td>1.0000</td>
<td>-0.7033</td>
</tr>
<tr>
<td>0.7426</td>
<td>-0.7033</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

Variance matrix.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>17.846</td>
<td>-4.989</td>
<td>43.123</td>
</tr>
<tr>
<td>-4.989</td>
<td>656.407</td>
<td>-247.692</td>
</tr>
<tr>
<td>43.123</td>
<td>-247.692</td>
<td>188.970</td>
</tr>
</tbody>
</table>

Sum of weights 21.0