NAG Library Function Document

nag_corr_cov (g02bxc)

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
#include <nag.h>
#include <nagg02.h>
void nag_corr_cov (Integer n, Integer m, const double x[], Integer tdx,
const Integer sx[], const 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 the one-pass algorithm of West (1979) as implemented in
nag_sum_sqs (g02buc) 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 \(x_{ij}\) is the value of the \(i\)th observation on the \(j\)th variable and \(w_i\) 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 References

5 Arguments
1: \textbf{n} – Integer \hspace{1cm} \textit{Input}
\textit{On entry:} the number of observations in the dataset, \( n \).
\textit{Constraint:} \( n > 1 \).

2: \textbf{m} – Integer \hspace{1cm} \textit{Input}
\textit{On entry:} the total number of variables, \( m \).
\textit{Constraint:} \( m \geq 1 \).

3: \textbf{x[n \times tdx]} – const double \hspace{1cm} \textit{Input}
\textit{On entry:} the data \( x[(i-1) \times tdx + j-1] \) must contain the \( i \)th observation on the \( j \)th variable, \( x_{ij} \), for \( i = 1,2,\ldots,n \) and \( j = 1,2,\ldots,m \).

4: \textbf{tdx} – Integer \hspace{1cm} \textit{Input}
\textit{On entry:} the stride separating matrix column elements in the array \( x \).
\textit{Constraint:} \( tdx \geq m \).

5: \textbf{sx[m]} – const Integer \hspace{1cm} \textit{Input}
\textit{On entry:} indicates which \( p \) variables to include in the analysis.
\( sx[j-1] > 0 \) \hspace{1cm} The \( j \)th variable is to be included.
\( sx[j-1] = 0 \) \hspace{1cm} The \( j \)th variable is not to be included.
\textbf{sx} is set to NULL. 
All variables are included in the analysis, i.e., \( p = m \).
\textit{Constraint:} \( sx[i] \geq 0 \), for \( i = 1,2,\ldots,m \).

6: \textbf{wt[n]} – const double \hspace{1cm} \textit{Input}
\textit{On entry:} \( w \), the optional frequency weighting for each observation, with \( wt[i-1] = w_i \). Usually \( w_i \) 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 NULL then \( w_i \) is set to 1 for all \( i \).
\textit{Constraint:} if \( wt \) is not NULL, \( \sum_{i=1}^{n} wt[i-1] > 1.0 \), \( wt[i-1] \geq 0.0 \), for \( i = 1,2,\ldots,n \).

7: \textbf{sw} – double * \hspace{1cm} \textit{Output}
\textit{On exit:} the sum of weights if \( wt \) is not NULL, otherwise \textbf{sw} contains the number of observations, \( n \).

8: \textbf{wmean[m]} – double \hspace{1cm} \textit{Output}
\textit{On exit:} the sample means. \( wmean[j-1] \) contains the mean for the \( j \)th variable.
9: \(\text{std}[\text{m}]\) – double

\textit{Output}

On exit: the standard deviations. \(\text{std}[j-1]\) contains the standard deviation for the \(j\)th variable.

10: \(\text{r}[\text{m} \times \text{tdr}]\) – double

\textit{Output}

On exit: the matrix of Pearson product-moment correlation coefficients. \(\text{r}[(j-1) \times \text{tdr} + k - 1]\) contains the correlation between variables \(j\) and \(k\), for \(j, k = 1, \ldots, p\).

11: \(\text{tdr}\) – Integer

\textit{Input}

On entry: the stride separating matrix column elements in the array \(\text{r}\).

\textit{Constraint:} \(\text{tdr} \geq \text{m}\).

12: \(\text{v}[\text{m} \times \text{tdv}]\) – double

\textit{Output}

On exit: the variance-covariance matrix. \(\text{v}[(j-1) \times \text{tdv} + k - 1]\) contains the covariance between variables \(j\) and \(k\), for \(j, k = 1, \ldots, p\).

13: \(\text{tdv}\) – Integer

\textit{Input}

On entry: the stride separating matrix column elements in the array \(\text{v}\).

\textit{Constraint:} \(\text{tdv} \geq \text{m}\).

14: \(\text{fail}\) – NagError*

\textit{Input/Output}

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

**NE_2_INT_ARG_LT**

On entry, \(\text{tdr} = \langle\text{value}\rangle\) while \(\text{m} = \langle\text{value}\rangle\).

The arguments must satisfy \(\text{tdr} \geq \text{m}\).

On entry, \(\text{tdv} = \langle\text{value}\rangle\) while \(\text{m} = \langle\text{value}\rangle\). These arguments must satisfy \(\text{tdv} \geq \text{m}\).

On entry, \(\text{tdx} = \langle\text{value}\rangle\) while \(\text{m} = \langle\text{value}\rangle\). These arguments must satisfy \(\text{tdx} \geq \text{m}\).

**NE_ALLOC_FAIL**

Dynamic memory allocation failed.

**NE_INT_ARG_LE**

On entry, \(\text{n} = \langle\text{value}\rangle\).

**NE_INT_ARG_LT**

On entry, \(\text{m} = \langle\text{value}\rangle\).

\textit{Constraint:} \(\text{m} \geq 1\).

**NE_NEG_SX**

On entry, at least one element of \(\text{sx}\) is negative.

**NE_NEG_WEIGHT**

On entry, at least one of the weights is negative.

**NE_POS_SX**

On entry, no element of \(\text{sx}\) is positive.
On entry, the sum of weights is less than 1.0.

A variable has zero variance.
At least one variable has zero variance. In this case \( v \) and \( \text{std} \) are as calculated, but \( r \) will contain zero for any correlation involving a variable with zero variance.

### 7 Accuracy

For a discussion of the accuracy of the one pass algorithm see Chan et al. (1982) and West (1979).

### 8 Parallelism and Performance

Not applicable.

### 9 Further Comments

Correlation coefficients based on ranks can be computed using \texttt{nag ken spe corr coeff} (g02brc).

### 10 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.

#### 10.1 Program Text

```c
/* \texttt{nag corr cov (g02bxc)} Example Program. */
/* Copyright 2014 Numerical Algorithms Group. */
/* Mark 3, 1992. */
/* Mark 8 revised, 2004. */
/
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg02.h>

#define X(I, J) x[(I) *tdx + J]
#define R(I, J) r[(I) *tdr + J]
#define V(I, J) v[(I) *tdv + J]

int main(void)
{
    Integer exit_status = 0, i, j, m, n, tdr, tdv, tdx, test;
    NagError fail;
    char w;
    double *r = 0, *std = 0, sw, *v = 0, *wmean = 0, *wt = 0, *wtptr, *x = 0;

    INIT_FAIL(fail);

    printf("\texttt{nag corr cov (g02bxc)} Example Program Results\n");

    /* Skip heading in data file */
    #ifdef _WIN32
    scanf_s("%*[\n]");
    #else
    scanf("%*[\n]");
    #endif

    test = 0;
    #ifdef _WIN32
    g02bxc
```

---

**NE_SW_LT_ONE**

On entry, the sum of weights is less than 1.0.

**NE_VAR_EQ_ZERO**

A variable has zero variance.
At least one variable has zero variance. In this case \( v \) and \( \text{std} \) are as calculated, but \( r \) will contain zero for any correlation involving a variable with zero variance.

### 7 Accuracy

For a discussion of the accuracy of the one pass algorithm see Chan et al. (1982) and West (1979).

### 8 Parallelism and Performance

Not applicable.

### 9 Further Comments

Correlation coefficients based on ranks can be computed using \texttt{nag ken spe corr coeff} (g02brc).

### 10 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.

#### 10.1 Program Text

```c
/* \texttt{nag corr cov (g02bxc)} Example Program. */
/* Copyright 2014 Numerical Algorithms Group. */
/* Mark 3, 1992. */
/* Mark 8 revised, 2004. */
/
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg02.h>

#define X(I, J) x[(I) *tdx + J]
#define R(I, J) r[(I) *tdr + J]
#define V(I, J) v[(I) *tdv + J]

int main(void)
{
    Integer exit_status = 0, i, j, m, n, tdr, tdv, tdx, test;
    NagError fail;
    char w;
    double *r = 0, *std = 0, sw, *v = 0, *wmean = 0, *wt = 0, *wtptr, *x = 0;

    INIT_FAIL(fail);

    printf("\texttt{nag corr cov (g02bxc)} Example Program Results\n");

    /* Skip heading in data file */
    #ifdef _WIN32
    scanf_s("%*[\n]");
    #else
    scanf("%*[\n]");
    #endif

    test = 0;
    #ifdef _WIN32
```
while ((scanf_s("%"NAG_IFMT"%"NAG_IFMT" %c", &m, &n, &w, 1) != EOF))
#else
while ((scanf("%"NAG_IFMT"%"NAG_IFMT" %c", &m, &n, &w) != EOF))
#endif
{
    if (m >= 1 && n >= 1)
    {
        if (!(x = NAG_ALLOC(n*m, double)) ||
            !(r = NAG_ALLOC(m*m, double)) ||
            !(v = NAG_ALLOC(m*m, double)) ||
            !(wt = NAG_ALLOC(n, double)) ||
            !(wmean = NAG_ALLOC(m, double)) ||
            !(std = NAG_ALLOC(m, double)))
        {
            printf("Allocation failure\n");
            exit_status = -1;
            goto END;
        }
        tdx = m;
        tdr = m;
        tdv = m;
    }
    else
    {
        printf("Invalid m or n.\n");
        exit_status = 1;
        return exit_status;
    }
#endif 
    for (i = 0; i < n; i++)
#endif _WIN32
    scanf_s("%lf", &wt[i]);
#else
    scanf("%lf", &wt[i]);
#endif
    for (i = 0; i < n; i++)
        for (j = 0; j < m; j++)
#endif _WIN32
    scanf_s("%lf", &X(i, j));
#else
    scanf("%lf", &X(i, j));
#endif
    if (w == 'w')
        wtptr = wt;
    else
        wtptr = (double *) 0;

    /* nag_corr_cov (g02bxc).
    * Product-moment correlation, unweighted/weighted
    * correlation and covariance matrix, allows variables to be
    * disregarded
    */
    nag_corr_cov(n, m, x, tdx, (Integer *) 0, wtptr, &sw, wmean, std,
        tdr, v, tdv, &fail);
    if (fail.code != NE_NOERROR)
    {
        printf("Error from nag_corr_cov (g02bxc).\n%\s\n", fail.message);
        exit_status = 1;
        goto END;
    }
    if (wtptr)
        printf("\nCase %"NAG_IFMT" --- Using weights\n", ++test);
    else
        printf("\nCase %"NAG_IFMT" --- Not using weights\n", ++test);
    printf("\nInput data\n");
    for (i = 0; i < n; i++)
        printf("%6.1f%6.1f%6.1f%6.1f\n",}
X(i, 0), X(i, 1), X(i, 2), wt[i]);

printf("\n");
printf("Sample means.\n");
for (i = 0; i < m; i++)
    printf("%6.1f\n", wmean[i]);
printf("\nStandard deviation.\n");
for (i = 0; i < m; i++)
    printf("%6.1f\n", std[i]);

printf("Correlation matrix.\n");
for (i = 0; i < m; i++)
{
    for (j = 0; j < m; j++)
        printf(" %7.4f ", R(i, j));
    printf("\n");
}

printf("Variance matrix.\n");
for (i = 0; i < m; i++)
{
    for (j = 0; j < m; j++)
        printf(" %7.3f ", V(i, j));
    printf("\n");
}
printf("Sum of weights %6.1f\n", sw);

END:
NAG_FREE(x);
NAG_FREE(r);
NAG_FREE(v);
NAG_FREE(wt);
NAG_FREE(wmean);
NAG_FREE(std);
}
return exit_status;

10.2 Program Data

nag_corr_cov (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
10.3 Program Results

*nag_corr_cov* (g02bxc) Example Program Results

Case 1 --- Using weights

Input data

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

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

3 3 w

\[
\begin{array}{ccc}
0.717 & 19.370 & 0.913 \\
1.119 & 0.133 & 9.700 \\
17.100 & 93.510 & 13.117 \\
30.900 & 0.013 & 78.710 \\
\end{array}
\]
### Case 4 --- 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>11.1</td>
<td>23.5</td>
<td>11.1</td>
<td>9.4</td>
</tr>
<tr>
<td>0.9</td>
<td>9.0</td>
<td>8.7</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Sample means.**

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
</tr>
<tr>
<td>10.9</td>
</tr>
<tr>
<td>9.8</td>
</tr>
</tbody>
</table>

**Standard deviation.**

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.8</td>
</tr>
<tr>
<td>11.8</td>
</tr>
<tr>
<td>1.2</td>
</tr>
</tbody>
</table>

**Correlation matrix.**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0000</td>
<td>0.9193</td>
<td>0.9200</td>
<td></td>
</tr>
<tr>
<td>0.9193</td>
<td>1.0000</td>
<td>0.6915</td>
<td></td>
</tr>
<tr>
<td>0.9200</td>
<td>0.6915</td>
<td>1.0000</td>
<td></td>
</tr>
</tbody>
</table>

**Variance matrix.**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>33.951</td>
<td>63.208</td>
<td>6.485</td>
<td></td>
</tr>
<tr>
<td>63.208</td>
<td>139.250</td>
<td>9.871</td>
<td></td>
</tr>
<tr>
<td>6.485</td>
<td>9.871</td>
<td>1.464</td>
<td></td>
</tr>
</tbody>
</table>

**Sum of weights**

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3.0</td>
</tr>
</tbody>
</table>

### Case 5 --- Not 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>11.1</td>
<td>3.5</td>
<td>13.1</td>
<td>19.4</td>
</tr>
<tr>
<td>0.9</td>
<td>0.0</td>
<td>78.7</td>
<td>0.0</td>
</tr>
</tbody>
</table>

**Sample means.**

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
</tr>
<tr>
<td>1.2</td>
</tr>
<tr>
<td>33.8</td>
</tr>
</tbody>
</table>

**Standard deviation.**

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5.8</td>
</tr>
<tr>
<td>2.0</td>
</tr>
</tbody>
</table>
Correlation matrix.

\[
\begin{pmatrix}
1.0000 & 0.9999 & -0.4781 \\
0.9999 & 1.0000 & -0.4881 \\
-0.4781 & -0.4881 & 1.0000
\end{pmatrix}
\]

Variance matrix.

\[
\begin{pmatrix}
33.951 & 11.567 & -108.343 \\
11.567 & 3.941 & -37.687 \\
-108.343 & -37.687 & 1512.750
\end{pmatrix}
\]

Sum of weights 3.0

Case 6 --- Using weights

Input data

\[
\begin{pmatrix}
1.1 & 0.1 & 9.7 & 0.7 \\
17.1 & 93.5 & 13.1 & 19.4 \\
30.9 & 0.0 & 78.7 & 0.9
\end{pmatrix}
\]

Sample means.

17.2
86.3
15.9

Standard deviation.

4.2
25.6
13.7

Correlation matrix.

\[
\begin{pmatrix}
1.0000 & -0.0461 & 0.7426 \\
-0.0461 & 1.0000 & -0.7033 \\
0.7426 & -0.7033 & 1.0000
\end{pmatrix}
\]

Variance matrix.

\[
\begin{pmatrix}
17.846 & -4.989 & 43.123 \\
-4.989 & 656.407 & -247.692 \\
43.123 & -247.692 & 188.970
\end{pmatrix}
\]

Sum of weights 21.0