# NAG C Library Function Document

# nag_partial_corr (g02byc)

## 1 Purpose

nag_partial_corr (g02byc) computes a partial correlation/variance-covariance matrix from a correlation or variance-covariance matrix computed by nag_corr_cov (g02bxc).

## 2 Specification

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

```
void nag_partial_corr(Integer m, Integer ny, Integer nx, const Integer sz[],
          const double r[], Integer tdr, double p[], Integer tdp,
          NagError *fail)
```

## 3 Description

Partial correlation can be used to explore the association between pairs of random variables in the presence of other variables. For three variables, $y_1$, $y_2$ and $x_3$ the partial correlation coefficient between $y_1$ and $y_2$ given $x_3$ is computed as:

$$\frac{r_{12} - r_{13}r_{23}}{\sqrt{(1 - r_{13}^2)(1 - r_{23}^2)}},$$

where $r_{ij}$ is the product-moment correlation coefficient between variables with subscripts $i$ and $j$. The partial correlation coefficient is a measure of the linear association between $y_1$ and $y_2$ having eliminated the effect due to both $y_1$ and $y_2$ being linearly associated with $x_3$. That is, it is a measure of association between $y_1$ and $y_2$ conditional upon fixed values of $x_3$. Like the full correlation coefficients the partial correlation coefficient takes a value in the range $(-1,1)$ with the value 0 indicating no association.

In general, let a set of variables be partitioned into two groups $Y$ and $X$ with $n_y$ variables in $Y$ and $n_x$ variables in $X$ and let the variance-covariance matrix of all $n_y + n_x$ variables be partitioned into,

$$\begin{bmatrix} \Sigma_{xx} & \Sigma_{xy} \\ \Sigma_{yx} & \Sigma_{yy} \end{bmatrix}$$

The variance-covariance of $Y$ conditional on fixed values of the $X$ variables is given by:

$$\Sigma_{y|x} = \Sigma_{yy} - \Sigma_{yx}\Sigma_{xx}^{-1}\Sigma_{xy}$$

The partial correlation matrix is then computed by standardising $\Sigma_{y|x}$,

$$\mathrm{diag}(\Sigma_{y|x})^{-\frac{1}{2}}\Sigma_{y|x}\,\mathrm{diag}(\Sigma_{y|x})^{-\frac{1}{2}}.$$

To test the hypothesis that a partial correlation is zero under the assumption that the data has an approximately Normal distribution a test similar to the test for the full correlation coefficient can be used. If $r$ is the computed partial correlation coefficient then the appropriate $t$ statistic is

$$r\sqrt{\frac{n - n_x - 2}{1 - r^2}}$$

which has approximately a Student's $t$-distribution with $n - n_x - 2$ degrees of freedom, where $n$ is the number of observations from which the full correlation coefficents were computed.

## 4 Parameters

1: **m** – Integer *Input*

*On entry:* the number of variables in the variance-covariance/correlation matrix given in **r**.

*Constraint:* **m** $\geq 3$.

2: **ny** – Integer *Input*

*On entry:* the number of $Y$ variables, $n_y$, for which partial correlation coefficents are to be computed.

*Constraint:* **ny** $\geq 2$.

3: **nx** – Integer *Input*

*On entry:* the number of $X$ variables, $n_x$, which are to be considered as fixed.

*Constraints:*
**nx** $\geq 1$,
**ny** $+$ **nx** $\leq$ **m**.

4: **sz[m]** – const Integer *Input*

*On entry:* indicates which variables belong to set $X$ and $Y$.
If **sz**$(i) < 0$, then the $i$th variable is a $Y$ variable, for $i = 1, 2, \ldots,$**m**.
If **sz**$(i) > 0$, then the $i$th variable is a $X$ variable.
If **sz**$(i) = 0$, then the $i$th variable is not included in the computations.

*Constraints:*
exactly **ny** elements of **sz** must be $< 0$,
exactly **nx** elements of **sz** must be $> 0$.

5: **r[m][tdr]** – const double *Input*

*On entry:* the variance-covariance or correlation matrix for the **m** variables as given by nag_corr_cov (g02bxc). Only the upper triangle need be given.

**Note:** the matrix must be a full rank variance-covariance or correlation matrix and so be positive-definite. This condition is not directly checked by the function.

6: **tdr** – Integer *Input*

*On entry:* the second dimension of the array **r** as declared in the function from which nag_partial_corr is called.

*Constraint:* **tdr** $\geq$ **m**.

7: **p[ny][tdp]** – double *Output*

*On exit:* the strict upper triangle of **p** contains the strict upper triangular part of the $n_y$ by $n_y$ partial correlation matrix. The lower triangle contains the lower triangle of the $n_y$ by $n_y$ partial variance-covariance matrix if the matrix given in **r** is a variance-covariance matrix. If the matrix given in **r** is a correlation matrix then the variance-covariance matrix is for standardised variables.

8: **tdp** – Integer *Input*

*On entry:* the second dimension of the array **p** as declared in the function from which nag_partial_corr is called.

*Constraint:* **tdp** $\geq$ **ny**.

9: **fail** – NagError * *Input/Output*

The NAG error parameter (see the Essential Introduction).

## 5 Error Indicators and Warnings

**NE_INT_ARG_LT**

On entry, **m** must not be less than 3: **m** = <*value*>.
On entry, **ny** must not be less than 2: **ny** = <*value*>.
On entry, **nx** must not be less than 1: **nx** = <*value*>.

**NE_3_INT_ARG_CONS**

On entry, **ny** = <*value*>, **nx** = <*value*> and **m** = <*value*>.
These parameters must satisfy **ny** + **nx** $\leq$ **m**.

**NE_2_INT_ARG_LT**

On entry, **tdr** = <*value*> while **m** = <*value*>.
These parameters must satisfy **tdr** $\geq$ **m**.

On entry, **tdp** = <*value*> while **ny** = <*value*>.
These parameters must satisfy **tdp** $\geq$ **ny**.

**NE_BAD_NY_SET**

On entry, **ny** = <*value*> and there are not exactly **ny** values of **isz** $<$ 0.
Number of values of **isz** $<$ 0 = <*value*>.

**NE_BAD_NX_SET**

On entry, **nx** = <*value*> and there are not exactly **nx** values of **isz** $<$ 0.

**NE_ALLOC_FAIL**

Memory allocation failed.

**NE_INTERNAL_ERROR**

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please consult NAG for assistance.

**NE_COR_MAT_RANK**

On entry, either the variance-covariance matrix or the correlation matrix is not of full rank. Try removing some of the **x** variables by setting the appropriate elements of **isz** to zero.

**NE_COR_MAT_POSDEF**

Either a diagonal element of the partial variance-covariance matrix is zero and/or a computed partial correlation coefficient is greater than one. Both indicate that the matrix input in **r** was not positive-definite.

## 6 Further Comments

Models that represent the linear associations given by partial correlations can be fitted using the multiple regression function nag_regsn_mult_linear (g02dac).

### 6.1 Accuracy

nag_partial_corr computes the partial variance-covariance matrix, $\Sigma_{y|x}$, by computing the Cholesky factorization of $\Sigma_{xx}$. If $\Sigma_{xx}$ is not of full rank the computation will fail.

### 6.2 References

Krzanowski W J (1990) *Principles of Multivariate Analysis* Oxford University Press

Morrison D F (1967) *Multivariate Statistical Methods* McGraw-Hill

Snedecor G W and Cochran W G (1967) *Statistical Methods* Iowa State University Press

Osborn J F (1979) *Statistical Exercises in Medical Research* Blackwell

## 7 See Also

nag_corr_cov (g02bxc)
nag_regsn_mult_linear (g02dac)

## 8 Example

Data, given by Osborn (1979), on the number of deaths, smoke ($mg/m^3$) and sulphur dioxide (parts/million) during an intense period of fog is input. The correlations are computed using nag_corr_cov (g02bxc) and the partial correlation between deaths and smoke given sulphur dioxide is computed using nag_partial_corr.

### 8.1 Program Text

```
/* nag_partial_corr (g02byc) Example Program.
 *
 * Copyright 2000 Numerical Algorithms Group.
 *
 * Mark 6, 2000.
 */


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

int main(void)
{

  double *r=0, *std=0, *v, *x=0, *xbar=0, sw;
  Integer *sz=0, j, k, m, n, nx, ny;
  Integer exit_status=0;
  NagError fail;

#define X(I,J) x[((I)-1)*m + ((J)-1)]
#define R(I,J) r[((I)-1)*m + ((J)-1)]

  INIT_FAIL(fail);
  Vprintf("g02byc Example Program Results\n");

  /*  Skip heading in data file */
  Vscanf("%*[^\n]");

  Vscanf("%ld %ld", &n, &m);
  if (!(r=NAG_ALLOC(m*m, double))
      || !(std=NAG_ALLOC(m, double))
      || !(v=NAG_ALLOC(m*m, double))
      || !(x=NAG_ALLOC(n*m, double))
      || !(xbar=NAG_ALLOC(m, double))
      || !(sz=NAG_ALLOC(m, Integer)))
    {
      Vprintf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }
  for (j = 1; j <= n; ++j)
    for (k = 1; k <= m; ++k)
      Vscanf("%lf", &X(j,k));

  /*  Calculate correlation matrix */
  g02bxc(n, m, x, m, 0, 0, &sw, xbar, std, r, m, v, m,
         &fail);
  if (fail.code == NE_NOERROR)
    {
      /*  Print the correlation matrix */
      Vprintf("\nCorrelation Matrix\n\n");
      for (j=1; j<=m; j++)
        {
          for(k=1; k<=m; k++)
            if (j>k)
              Vprintf("%11s", "");
            else
              Vprintf("%7.4f%4s",  R(j,k),"");
          Vprintf("\n");
        }

      Vscanf("%ld %ld", &ny, &nx);
      for (j = 1; j <= m; ++j)
        Vscanf("%ld", &sz[j - 1]);

      /*  Calculate partial correlation matrix */
      g02byc(m, ny, nx, sz, v, m, r, m, &fail);
      if (fail.code != NE_NOERROR)
        {
          Vprintf("Error from g02byc.\n%s\n", fail.message);
          exit_status = 1;
          goto END;
        }
      /*  Print partial correlation matrix */
      Vprintf("\n");
      Vprintf("\nPartial Correlation Matrix\n\n");
      for (j=1; j<=ny; j++)
        {
          for(k=1; k<=ny; k++)
            {
              if (j>k)
                Vprintf("%11s", "");
              else if (j==k)
                Vprintf("%7.4f%4s",  1.0, "");
              else
                Vprintf("%7.4f%4s",  R(j,k), "");
            }
          Vprintf("\n");
        }

    }
  else
    {
      Vprintf("Error from g02bxc.\n%s\n", fail.message);
      exit_status = 1;
      goto END;
    }
 END:
  if (r) NAG_FREE(r);
  if (std) NAG_FREE(std);
  if (v) NAG_FREE(v);
  if (x) NAG_FREE(x);
  if (xbar) NAG_FREE(xbar);
  if (sz) NAG_FREE(sz);
  return exit_status;
}
```

## 8.2 Program Data

```
g02byc Example Program Data
15 3
112 0.30 0.09
140 0.49 0.16
143 0.61 0.22
120 0.49 0.14
196 2.64 0.75
294 3.45 0.86
513 4.46 1.34
518 4.46 1.34
430 1.22 0.47
274 1.22 0.47
255 0.32 0.22
236 0.29 0.23
256 0.50 0.26
222 0.32 0.16
213 0.32 0.16

 2  1
-1 -1  1
```

## 8.3 Program Results

```
g02byc Example Program Results

Correlation Matrix

 1.0000      0.7560      0.8309
             1.0000      0.9876
                         1.0000


Partial Correlation Matrix

 1.0000     -0.7381
             1.0000
```