**nag_regsn_mult_linear_add_var (g02dec)**

### 1. Purpose

*nag_regsn_mult_linear_add_var (g02dec)* adds a new independent variable to a general linear regression model.

### 2. Specification

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

void nag_regsn_mult_linear_add_var(Integer n, Integer ip, double q[],
                Integer tdq, double p[], double wt[], double x[], double *rss,
                double tol, NagError *fail)
```

### 3. Description

A linear regression model may be built up by adding new independent variables to an existing model. *nag_regsn_mult_linear_add_var* updates the QR decomposition used in the computation of the linear regression model. The QR decomposition may come from *nag_regsn_mult_linear (g02dac)* or a previous call to *nag_regsn_mult_linear_add_var*. The general linear regression model is defined by:

\[
y = X\beta + \varepsilon
\]

where \( y \) is a vector of \( n \) observations on the dependent variable, \( X \) is an \( n \) by \( p \) matrix of the independent variables of column rank \( k \), \( \beta \) is a vector of length \( p \) of unknown parameters, and \( \varepsilon \) is a vector of length \( n \) of unknown random errors such that \( \text{var} \varepsilon = V\sigma^2 \), where \( V \) is a known diagonal matrix.

If \( V = I \), the identity matrix, then least-squares estimation is used. If \( V \neq I \), then for a given weight matrix \( W \propto V^{-1} \), weighted least-squares estimation is used.

The least-squares estimates, \( \hat{\beta} \) of the parameters \( \beta \) minimize \( (y - X\beta)^T(y - X\beta) \) while the weighted least-squares estimates minimize \( (y - X\beta)^TW(y - X\beta) \).

The parameter estimates may be found by computing a QR decomposition of \( X \) (or \( W^{1/2}X \) in the weighted case), i.e.,

\[
X = QR^* \quad \text{(or } W^{1/2}X = QR^*)
\]

where \( R^* = \begin{pmatrix} R \\ 0 \end{pmatrix} \) and \( R \) is a \( p \) by \( p \) upper triangular matrix and \( Q \) is an \( n \) by \( n \) orthogonal matrix.

If \( R \) is of full rank, then \( \hat{\beta} \) is the solution to:

\[
R\hat{\beta} = c_1
\]

where \( c = QTy \) (or \( QTW^{1/2}y \)) and \( c_1 \) is the first \( p \) elements of \( c \).

If \( R \) is not of full rank a solution is obtained by means of a singular value decomposition (SVD) of \( R \).

To add a new independent variable, \( x_{p+1} \), \( R \) and \( c \) have to be updated. The matrix \( Q_{p+1} \) is found such that \( Q_{p+1}^T[R : QT_{p+1}x_{p+1}] \) (or \( Q_{p+1}^T[R : QT^{1/2}W^{1/2}x_{p+1}] \)) is upper triangular. The vector \( c \) is then updated by multiplying by \( Q_{p+1}^T \).

The new independent variable is tested to see if it is linearly related to the existing independent variables by checking that at least one of the values \( (Q^T_{p+1}x_{p+1})_i \), for \( i = p+2, p+3, \ldots, n \) is non-zero.

The new parameter estimates, \( \hat{\beta} \), can then be obtained by a call to *nag_regsn_mult_linear_updmd_model (g02ddc)*.

The function can be used with \( p = 0 \), in which case \( R \) and \( c \) are initialized.
4. Parameters

\( n \)

Input: the number of observations, \( n \).
Constraint: \( n \geq 1 \).

\( ip \)

Input: the number of independent variables already in the model, \( p \).
Constraint: \( ip \geq 0 \) and \( ip < n \).

\( q[n][tdq] \)

Input: if \( ip \neq 0 \), then \( q \) must contain the results of the \( QR \) decomposition for the model with \( p \) parameters as returned by \texttt{nag_regn_mult_linear} (g02dac) or a previous call to \texttt{nag_regn_mult_linear_add_var}.
If \( ip = 0 \), then the first column of \( q \) should contain the \( n \) values of the dependent variable, \( y \).
Output: the results of the \( QR \) decomposition for the model with \( p + 1 \) parameters:
the first column of \( q \) contains the updated value of \( c \),
the columns 2 to \( ip + 1 \) are unchanged,
the first \( ip + 1 \) elements of column \( ip + 2 \) contain the new column of \( R \), while the remaining \( n - ip - 1 \) elements contain details of the matrix \( Q_{p+1} \).

\( tdq \)

Input: \( tdq \) the last dimension of the array \( q \) as declared in the function from which \texttt{nag_regn_mult_linear_add_var} is called.
Constraint: \( tdq \geq ip + 2 \).

\( p[ip+1] \)

Input: \( p \) contains further details of the \( QR \) decomposition used. The first \( ip \) elements of \( p \) must contain the zeta values for the \( QR \) decomposition (see \texttt{nag_real_qr} (f01qcc) for details).
The first \( ip \) elements of array \( p \) are provided by \texttt{nag_regn_mult_linear} (g02dac) or by previous calls to \texttt{nag_regn_mult_linear_add_var}.
Output: the first \( ip \) elements of \( p \) are unchanged and the \((ip+1)\)th element contains the zeta value for \( Q_{p+1} \).

\( wt[n] \)

Input: if weighted estimates are required, then \( wt \) must contain the weights to be used in the weighted regression. Otherwise \( wt \) need not be defined and may be set to the null pointer \texttt{NULL}, i.e., \((\text{double } *)0\).
If \( wt[i] = 0.0 \), then the \( i \)th observation is not included in the model, in which case the effective number of observations is the number of observations with non-zero weights.
If \( wt = \text{NULL} \), then the effective number of observations is \( n \).
Constraint: \( wt = \text{NULL} \) or \( wt[i] \geq 0.0 \), for \( i = 0, 1, \ldots, n - 1 \).

\( x[n] \)

Input: the new independent variable, \( x \).

\( rss \)

Output: the residual sum of squares for the new fitted model.

Note: this will only be valid if the model is of full rank, see Section 6.

\( tol \)

Input: the value of \( tol \) is used to decide if the new independent variable is linearly related to independent variables already included in the model. If the new variable is linearly related then \( c \) is not updated. The smaller the value of \( tol \) the stricter the criterion for deciding if there is a linear relationship.
Suggested value: \( tol = 0.000001 \).
Constraint: \( tol > 0.0 \).

\( fail \)

The NAG error parameter, see the Essential Introduction to the NAG C Library.
5. Error Indications and Warnings

**NE_INT_ARG_LT**
- On entry, \( n \) must not be less than 1: \( n = \langle \text{value} \rangle \).
- On entry, \( ip \) must not be less than 0: \( ip = \langle \text{value} \rangle \).

**NE_2_INT_ARG_GE**
- On entry \( ip = \langle \text{value} \rangle \) while \( n = \langle \text{value} \rangle \). These parameters must satisfy \( ip < n \).

**NE_2_INT_ARG_LT**
- On entry \( tdq = \langle \text{value} \rangle \) while \( ip+2 = \langle \text{value} \rangle \). These parameters must satisfy \( tdq \geq ip+2 \).

**NE_REAL_ARG_LT**
- On entry, \( wt[\langle \text{value} \rangle] \) must not be less than 0.0: \( wt[\langle \text{value} \rangle] = \langle \text{value} \rangle \).

**NE_REAL_ARG_LE**
- On entry, \( tol = \langle \text{value} \rangle \).

**NE_VAR_NOT_IND**
- The new independent variable is a linear combination of existing variables. The \((ip+1)\)th column of \( q \) is, therefore, null.

6. Further Comments

It should be noted that the residual sum of squares produced by \texttt{nag_regnu_mult_linear_add_var} may not be correct if the model to which the new independent variable is added is not of full rank. In such a case \texttt{nag_regnu_mult_linear_upd_model (g02ddc)} should be used to calculate the residual sum of squares.

6.1. Accuracy

The accuracy is closely related to the accuracy of \texttt{nag_real_apply_q (f01qdc)} which should be consulted for further details.

6.2. References


Searle S R (1971) \textit{Linear Models} Wiley.

7. See Also

\texttt{nag_real_qr (f01qcc)}

\texttt{nag_real_apply_q (f01qdc)}

\texttt{nag_regnu_linear (g02dac)}

\texttt{nag_regnu_mult_linear_upd_model (g02ddc)}

\texttt{nag_regnu_linear_delete_var (g02dfc)}

8. Example

A data set consisting of 12 observations is read in. The four independent variables are stored in the array \( x \) while the dependent variable is read into the first column of \( q \). If the character variable \texttt{meanc} indicates that a mean should be included in the model, a variable taking the value 1.0 for all observations is set up and fitted. Subsequently, one variable at a time is selected to enter the model as indicated by the input value of \texttt{indx}. After the variable has been added the parameter estimates are calculated by \texttt{nag_regnu_mult_linear_upd_model (g02ddc)} and the results printed. This is repeated until the input value of \texttt{indx} is 0.
8.1. Program Text

/* nag_resgn_mult_linear_add_var(g02dec) Example Program */
*/
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg02.h>

#define NMAX 12
#define MMAX 5
#define TDX MMAX
#define TDQ MMAX+1

main()
{
    double rss, rsst, tol;
    Integer i, indx, ip, rank, j, m, n;
    double df;
    Boolean svd;
    char meanc, weight;
    Nag_IncludeMean mean;
    double b[MMAX], cov[MMAX*(MMAX+1)/2], p[MMAX*(MMAX+2)],
    q[NMAX][MMAX+1], se[MMAX], wt[NMAX], x[NMAX][MMAX], xe[NMAX];
    double *wtptr;
    static NagError fail;

    Vprintf("g02dec Example Program Results\n");
    /* Skip heading in data file */
    Vscanf("%*[\n");
    Vscanf("%ld %ld %c %c", &n, &m, &weight, &meanc);
    if (meanc=='m')
        mean = Nag_MeanInclude;
    else
        mean = Nag_MeanZero;
    if (weight=='w')
        wtptr = wt;
    else
        wtptr = (double *)0;
    /* Set tolerance */
    tol = 0.000001e0;
    ip = 0;
    if (mean==Nag_MeanInclude)
    {
        for (i=0; i<n; i++)
        {
            for (j=0; j<m; j++)
                Vscanf("%lf", &x[i][j]);
            Vscanf("%lf%lf", &q[i][0], &wt[i]);
        }
        else
        {
            for (i=0; i<n; i++)
            {
                for (j=0; j<m; j++)
                    Vscanf("%lf", &x[i][j]);
                Vscanf("%lf", &q[i][0]);
            }
        }
    } /* Set tolerance */
    tol = 0.000001e0;
    ip = 0;
    if (mean==Nag_MeanInclude)
for (i = 0; i<n; ++i)
x[i] = 1.0;
g02dec(n, ip, (double *)q, (Integer)(TDQ), p, wtptr, xe, &rss,
tol, NAGERR_DEFAULT);

ip = 1;
while(scanf("%ld", &indx)!=EOF)
{
    if (indx>0)
    {
        for (i=0; i<n; i++)
            xe[i] = x[i][indx-1];
g02dec(n, ip, (double *)q, (Integer)(TDQ), p, wtptr, xe, &rss,
tol, &fail);
        if (fail.code==NE_NOERROR)
            {
            ip += 1;
            Vprintf("Variable %4ld added
", indx);
            rssst = 0.0;
g02ddc(n, ip, (double *)q, (Integer)(TDQ), &rssst, &df, b, se,
cov, &svd, &rank, p, tol, NAGERR_DEFAULT);
            if (svd)
                Vprintf("Model not of full rank
);
            Vprintf("Residual sum of squares = %13.4e
", rssst);
            Vprintf("Degrees of freedom = %3.1f
", df);
            Vprintf("Variable Parameter estimate Standard error
")
            for (j=0; j<ip; j++)
                Vprintf("%6ld%20.4e%20.4e
", j+1, b[j], se[j]);
            Vprintf("\n");
        }
        else if (fail.code==NE_NVAR_NOT_IND)
            Vprintf(" * New variable not added *
");
        else
            {
                Vprintf("%s
", fail.message);
                exit(EXIT_FAILURE);
            }
    }
else
    {
        Vfprintf(stderr, "One or both of m and n are out of range:\
m = %-3ld while n = %-3ld
", m, n);
        exit(EXIT_FAILURE);
    }
exit(EXIT_SUCCESS);

8.2. Program Data

g02dec Example Program Data
12 4 u m
1.0 1.4 0.0 0.0 4.32
1.5 2.2 0.0 0.0 5.21
2.0 4.5 0.0 0.0 6.49
2.5 6.1 0.0 0.0 7.10
3.0 7.1 0.0 0.0 7.94
3.5 7.7 0.0 0.0 8.53
4.0 8.3 1.0 4.0 8.84
4.5 8.6 1.0 4.5 9.02
5.0 8.8 1.0 5.0 9.27
5.5 9.0 1.0 5.5 9.43
6.0 9.3 1.0 6.0 9.68
6.5 9.2 1.0 6.5 9.83
1
3
### 8.3. Program Results

**g02dec Example Program Results**

**Variable 1 added**

Residual sum of squares = 4.0164e+00  
Degrees of freedom = 10.0

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter estimate</th>
<th>Standard error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.4100e+00</td>
<td>4.3756e-01</td>
</tr>
<tr>
<td>2</td>
<td>9.4979e-01</td>
<td>1.0599e-01</td>
</tr>
</tbody>
</table>

**Variable 3 added**

Residual sum of squares = 3.8872e+00  
Degrees of freedom = 9.0

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter estimate</th>
<th>Standard error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.2236e+00</td>
<td>5.6734e-01</td>
</tr>
<tr>
<td>2</td>
<td>1.0554e+00</td>
<td>2.2217e-01</td>
</tr>
<tr>
<td>3</td>
<td>-4.1962e-01</td>
<td>7.6695e-01</td>
</tr>
</tbody>
</table>

**Variable 4 added**

Residual sum of squares = 1.8702e-01  
Degrees of freedom = 8.0

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter estimate</th>
<th>Standard error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.7605e+00</td>
<td>1.7592e-01</td>
</tr>
<tr>
<td>2</td>
<td>1.7057e+00</td>
<td>7.3100e-02</td>
</tr>
<tr>
<td>3</td>
<td>4.4575e+00</td>
<td>4.2676e-01</td>
</tr>
<tr>
<td>4</td>
<td>-1.3006e+00</td>
<td>1.0338e-01</td>
</tr>
</tbody>
</table>

**Variable 2 added**

Residual sum of squares = 8.4066e-02  
Degrees of freedom = 7.0

<table>
<thead>
<tr>
<th>Variable</th>
<th>Parameter estimate</th>
<th>Standard error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.1440e+00</td>
<td>1.8181e-01</td>
</tr>
<tr>
<td>2</td>
<td>9.0748e-01</td>
<td>2.7761e-01</td>
</tr>
<tr>
<td>3</td>
<td>2.0790e+00</td>
<td>8.6804e-01</td>
</tr>
<tr>
<td>4</td>
<td>-6.1589e-01</td>
<td>2.4530e-01</td>
</tr>
<tr>
<td>5</td>
<td>2.9224e-01</td>
<td>9.9810e-02</td>
</tr>
</tbody>
</table>