1 Purpose

nag_regsn_quant_linear_iid (g02qfc) performs a multiple linear quantile regression, returning the parameter estimates and associated confidence limits based on an assumption of Normal, independent, identically distributed errors. nag_regsn_quant_linear_iid (g02qfc) is a simplified version of nag_regsn_quant_linear (g02qgc).

2 Specification

```c
#include <nag.h>
#include <nagg02.h>
void nag_regsn_quant_linear_iid (Integer n, Integer m, const double x[],
                           const double y[], Integer ntau, const double tau[],
                           double *df,
                           double b[], double bl[], double bu[], Integer info[],
                           NagError *fail)
```

3 Description

Given a vector of \( n \) observed values, \( y = \{y_i : i = 1, 2, \ldots, n\} \), an \( n \times p \) design matrix \( X \), a column vector, \( x \), of length \( p \) holding the \( i \)th row of \( X \) and a quantile \( \tau \in (0, 1) \), nag_regsn_quant_linear_iid (g02qfc) estimates the \( p \)-element vector \( \beta \) as the solution to

\[
\underset{\beta \in \mathbb{R}^p}{\text{minimize}} \sum_{i=1}^{n} \rho_{\tau}(y_i - x_i^T \beta)
\]

where \( \rho_{\tau} \) is the piecewise linear loss function \( \rho_{\tau}(z) = z(\tau - I(z < 0)) \), and \( I(z < 0) \) is an indicator function taking the value 1 if \( z < 0 \) and 0 otherwise.

nag_regsn_quant_linear_iid (g02qfc) assumes Normal, independent, identically distributed (IID) errors and calculates the asymptotic covariance matrix from

\[
\Sigma = \frac{\tau(1 - \tau)}{n} (s(\tau))^2 (X^T X)^{-1}
\]

where \( s \) is the sparsity function, which is estimated from the residuals, \( r_i = y_i - x_i^T \hat{\beta} \) (see Koenker (2005)).

Given an estimate of the covariance matrix, \( \hat{\Sigma} \), lower, \( \hat{\beta}_L \), and upper, \( \hat{\beta}_U \), limits for a 95\% confidence interval are calculated for each of the \( p \) parameters, via

\[
\hat{\beta}_L = \hat{\beta}_i - t_{n-p,0.975} \sqrt{\hat{\Sigma}_{ii}}, \quad \hat{\beta}_U = \hat{\beta}_i + t_{n-p,0.975} \sqrt{\hat{\Sigma}_{ii}}
\]

where \( t_{n-p,0.975} \) is the 97.5 percentile of the Student’s \( t \) distribution with \( n - k \) degrees of freedom, where \( k \) is the rank of the cross-product matrix \( X^T X \).

Further details of the algorithms used by nag_regsn_quant_linear_iid (g02qfc) can be found in the documentation for nag_regsn_quant_linear (g02qgc).

4 References

5 Arguments

1: \( \text{n} \) – Integer

\textit{Input}

\textit{On entry:} \( n \), the number of observations in the dataset.

\textit{Constraint:} \( n \geq 2 \).

2: \( \text{m} \) – Integer

\textit{Input}

\textit{On entry:} \( p \), the number of variates in the model.

\textit{Constraint:} \( 1 \leq m < n \).

3: \( \text{x}[\text{n} \times \text{m}] \) – const double

\textit{Input}

\textit{Note:} where \( X(i,j) \) appears in this document, it refers to the array element \( x[(i - 1) \times m + j - 1] \).

\textit{On entry:} \( X \), the design matrix, with the \( i \)th value for the \( j \)th variate supplied in \( X(i,j) \), for \( i = 1,2,\ldots,n \) and \( j = 1,2,\ldots,m \).

4: \( \text{y}[\text{n}] \) – const double

\textit{Input}

\textit{On entry:} \( y \), the observations on the dependent variable.

5: \( \text{ntau} \) – Integer

\textit{Input}

\textit{On entry:} the number of quantiles of interest.

\textit{Constraint:} \( ntau \geq 1 \).

6: \( \text{tau}[ntau] \) – const double

\textit{Input}

\textit{On entry:} the vector of quantiles of interest. A separate model is fitted to each quantile.

\textit{Constraint:} \( \sqrt{\epsilon} < \tau[l - 1] < 1 - \sqrt{\epsilon} \) where \( \epsilon \) is the \textit{machine precision} returned by \texttt{nag_machine_precision (X02AJC)}, for \( l = 1,2,\ldots,ntau \).

7: \( \text{df} \) – double *

\textit{Output}

\textit{On exit:} the degrees of freedom given by \( n - k \), where \( n \) is the number of observations and \( k \) is the rank of the cross-product matrix \( X^T X \).

8: \( \text{b}[m \times ntau] \) – double

\textit{Output}

\textit{Note:} where \( B(j,l) \) appears in this document, it refers to the array element \( b[(l - 1) \times m + j - 1] \).

\textit{On exit:} \( \hat{\beta} \), the estimates of the parameters of the regression model, with \( B(j,l) \) containing the coefficient for the variable in column \( j \) of \( X \), estimated for \( \tau = \tau[l - 1] \).

9: \( \text{b}[m \times ntau] \) – double

\textit{Output}

\textit{Note:} where \( BL(j,l) \) appears in this document, it refers to the array element \( bl[(l - 1) \times m + j - 1] \).

\textit{On exit:} \( \hat{\beta}_l \), the lower limit of a 95\% confidence interval for \( \hat{\beta} \), with \( BL(j,l) \) holding the lower limit associated with \( B(j,l) \).

10: \( \text{bu}[m \times ntau] \) – double

\textit{Output}

\textit{Note:} where \( BU(j,l) \) appears in this document, it refers to the array element \( bu[(l - 1) \times m + j - 1] \).

\textit{On exit:} \( \hat{\beta}_U \), the upper limit of a 95\% confidence interval for \( \hat{\beta} \), with \( BU(j,l) \) holding the upper limit associated with \( B(j,l) \).
On exit: info[l] holds additional information concerning the model fitting and confidence limit calculations when \( \tau = \text{tau}[l] \).

**Code** | **Warning** |
--- | --- |
0 | Model fitted and confidence limits calculated successfully. |
1 | The function did not converge whilst calculating the parameter estimates. The returned values are based on the estimate at the last iteration. |
2 | A singular matrix was encountered during the optimization. The model was not fitted for this value of \( \tau \). |
8 | The function did not converge whilst calculating the confidence limits. The returned limits are based on the estimate at the last iteration. |
16 | Confidence limits for this value of \( \tau \) could not be calculated. The returned upper and lower limits are set to a large positive and large negative value respectively. |

It is possible for multiple warnings to be applicable to a single model. In these cases the value returned in info is the sum of the corresponding individual nonzero warning codes.

**fail** – NagError *

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

**6 Error Indicators and Warnings**

**NE_ALLOC_FAIL**

Dynamic memory allocation failed.

See Section 3.2.1.2 in the Essential Introduction for further information.

**NE_BAD_PARAM**

On entry, argument \(<\text{value}\>) had an illegal value.

**NE_INT**

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

Constraint: \( n \geq 2 \).

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

Constraint: \( ntau \geq 1 \).

**NE_INT_2**

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

Constraint: \( 1 \leq m < n \).

**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 contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.

See Section 3.6.6 in the Essential Introduction for further information.

**NE_NO_LICENCE**

Your licence key may have expired or may not have been installed correctly.

See Section 3.6.5 in the Essential Introduction for further information.

**NE_REAL_ARRAY**

On entry, \( \text{tau}[\langle\text{value}\rangle] = \langle\text{value}\rangle \) is invalid.
NW_POTENTIAL_PROBLEM
A potential problem occurred whilst fitting the model(s).
Additional information has been returned in info.

7 Accuracy
Not applicable.

8 Parallelism and Performance
nag_regsn_quant_linear_iid (g02qfc) is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

nag_regsn_quant_linear_iid (g02qfc) makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this function. Please also consult the Users’ Note for your implementation for any additional implementation-specific information.

9 Further Comments
Calling nag_regsn_quant_linear_iid (g02qfc) is equivalent to calling nag_regsn_quant_linear (g02qgc) with

order = Nag_RowMajor, intcpt = Nag_NoIntercept,
no weights supplied, i.e., wt set to NULL,
pddat = m,
setting each element of isx to 1,
    ip = m,
    Interval Method = IID, and
    Significance Level = 0.95.

10 Example
A quantile regression model is fitted to Engels 1857 study of household expenditure on food. The model regresses the dependent variable, household food expenditure, against household income. An intercept is included in the model by augmenting the dataset with a column of ones.

10.1 Program Text
/* nag_regsn_quant_linear_iid (g02qfc) Example Program. */
/* Copyright 2014 Numerical Algorithms Group. */
/* Mark 23, 2011. */
/* Pre-processor includes */
#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg02.h>
#define B(i,j) b[(j) * m + i]
#define BU(i,j) bu[(j) * m + i]
#define BL(i,j) bl[(j) * m + i]
#define X(i,j) x[(i) * m + j]
int main(void) {
    /* Integer scalar and array declarations */
    Integer i, j, l, m, n, ntau;
    Integer *info = 0;
    Integer exit_status = 0;
    /* NAG structures */
    NagError fail;
    /* Double scalar and array declarations */
    double df;
    double *b = 0, *bl = 0, *bu = 0, *tau = 0, *x = 0, *y = 0;
    /* Initialise the error structure */
    INIT_FAIL(fail);
    printf("nag_regsn_quant_linear_iid (g02qfc) Example Program Results\n\n");
    /* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n "");
#else
    scanf("%*[\n "");
#endif
    /* Read in the problem size */
#ifdef _WIN32
    scanf_s("%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%*[\n ",&n,&m,&ntau);
#else
    scanf("%"NAG_IFMT"%"NAG_IFMT"%"NAG_IFMT"%*[\n ",&n,&m,&ntau);
#endif
    /* Allocate memory for input arrays */
    if (!(y = NAG_ALLOC(n, double)) ||
        !(tau = NAG_ALLOC(ntau, double)) ||
        !(x = NAG_ALLOC(n*m, double)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }
    /* Read in the data */
    for (i = 0; i < n; i++) {
        for (j = 0; j < m; j++)
#ifdef _WIN32
        scanf_s("%lf", &X(i,j));
#else
        scanf("%lf", &X(i,j));
#endif
#ifdef _WIN32
        scanf_s("%lf", &y[i]);
#else
        scanf("%lf", &y[i]);
#endif
#ifdef _WIN32
        scanf_s("%*[\n ");
#else
        scanf("%*[\n ");
#endif
    }
    /* Read in the quantiles required */
    for (l = 0; l < ntau; l++)
#ifdef _WIN32
        scanf_s("%lf", &tau[l]);
#else
        scanf("%lf", &tau[l]);
#endif
}
/* Allocate memory for output arrays */
if (!(b = NAG_ALLOC(m*ntau, double)) ||
    !(info = NAG_ALLOC(ntau, Integer)) ||
    !(bl = NAG_ALLOC(m*ntau, double)) ||
    !(bu = NAG_ALLOC(m*ntau, double)))
{
    printf("Allocation failure\n");
    exit_status = -1;
    goto END;
}

/* nag_regsn_quant_linear_iid (g02qfc). Quantile linear regression, simple
interface, independent, identically distributed (IID) errors */
if (fail.code != NE_NOERROR) {
    printf("Error from nag_regsn_quant_linear_iid (g02qfc).\n%"NAG_IFMT, fail.message);
    if (fail.code == NW_POTENTIAL_PROBLEM) {
        printf("Additional error information: ");
        for (i = 0; i < ntau; i++)
            printf("%"NAG_IFMT, info[i]);
        printf("\n");
    } else {
        exit_status = 1;
        goto END;
    }
}

/* Display the parameter estimates */
for (l = 0; l < ntau; l++) {
    printf(" Quantile: %6.3f\n\n", tau[l]);
    printf(" Lower Parameter Upper\n");
    printf(" Limit Estimate Limit\n");
    for (j = 0; j < m; j++) {
        printf(" %"NAG_IFMT, BL(j,l), B(j,l), BU(j,l));
    }
    printf("\n\n");
}

END:
NAG_FREE(info);
NAG_FREE(b);
NAG_FREE(bl);
NAG_FREE(bu);
NAG_FREE(tau);
NAG_FREE(x);
NAG_FREE(y);
return(exit_status);
}

10.2 Program Data

nag_regsn_quant_linear_iid (g02qfc) Example Program Data

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>235</td>
<td>2</td>
<td>5</td>
<td>n, m, ntau</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>420.1577</td>
<td>255.8394</td>
<td>1.0</td>
<td>800.7990</td>
<td>572.0807</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>541.117</td>
<td>310.9587</td>
<td>1.0</td>
<td>1245.6964</td>
<td>907.3969</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>901.1575</td>
<td>485.6800</td>
<td>1.0</td>
<td>1201.0002</td>
<td>811.5776</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>639.0802</td>
<td>402.9974</td>
<td>1.0</td>
<td>634.0002</td>
<td>427.7975</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>750.8756</td>
<td>495.5608</td>
<td>1.0</td>
<td>956.2315</td>
<td>649.9985</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>945.7989</td>
<td>633.7978</td>
<td>1.0</td>
<td>1148.6010</td>
<td>860.6002</td>
<td>1.0</td>
</tr>
<tr>
<td>1.0</td>
<td>829.3979</td>
<td>630.7566</td>
<td>1.0</td>
<td>1768.8236</td>
<td>1143.4211</td>
<td>1.0</td>
</tr>
<tr>
<td>ID</td>
<td>g02 - Correlation and Regression Analysis</td>
<td>g02qfc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------------</td>
<td>-------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>979.1640</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1309.8789</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1458.8789</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>502.8390</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>616.7168</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>790.9225</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>555.8786</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>713.4412</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>838.7561</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>535.0766</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>596.4408</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>924.5619</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>487.7583</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>692.6397</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>997.8770</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>506.9995</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>654.1587</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>933.9193</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>433.6813</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>587.5962</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>896.4746</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>454.4782</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>584.9989</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>800.7990</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>502.4369</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>713.5197</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>900.5056</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>880.5969</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>796.8289</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>854.8791</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1167.3761</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>523.8000</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>670.7792</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>377.0584</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>851.5430</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1121.0937</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>625.5179</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>805.5377</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>558.5812</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>884.4005</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1257.4989</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>2051.1789</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1300.9989</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>2432.3910</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>940.9218</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1177.8547</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1222.5939</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1519.5811</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>687.6638</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>953.1192</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>939.0418</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1283.4025</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1511.5789</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1342.5821</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>511.7980</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>689.7988</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>952.3074</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1056.0080</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>387.3195</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>410.9987</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>499.7510</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>832.7554</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>614.9886</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>887.4658</td>
<td>1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mark 25
g02qfc.7
### 10.3 Program Results

**nag_regsn_quant_linear_iid (g02qfc) Example Program Results**

Quantile: 0.100

<table>
<thead>
<tr>
<th>Lower</th>
<th>Parameter Limit</th>
<th>Upper Limit</th>
<th>Estimate Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>74.946</td>
<td>110.142</td>
<td>145.337</td>
</tr>
<tr>
<td>2</td>
<td>0.370</td>
<td>0.402</td>
<td>0.433</td>
</tr>
</tbody>
</table>

Quantile: 0.250

<table>
<thead>
<tr>
<th>Lower</th>
<th>Parameter Limit</th>
<th>Upper Limit</th>
<th>Estimate Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>64.232</td>
<td>95.483</td>
<td>126.735</td>
</tr>
<tr>
<td>2</td>
<td>0.446</td>
<td>0.474</td>
<td>0.502</td>
</tr>
</tbody>
</table>

Quantile: 0.500

<table>
<thead>
<tr>
<th>Lower</th>
<th>Parameter Limit</th>
<th>Upper Limit</th>
<th>Estimate Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>55.399</td>
<td>81.482</td>
<td>107.566</td>
</tr>
<tr>
<td>2</td>
<td>0.537</td>
<td>0.560</td>
<td>0.584</td>
</tr>
</tbody>
</table>

Quantile: 0.750

<table>
<thead>
<tr>
<th>Lower</th>
<th>Parameter Limit</th>
<th>Upper Limit</th>
<th>Estimate Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>41.372</td>
<td>62.396</td>
<td>83.421</td>
</tr>
<tr>
<td>2</td>
<td>0.625</td>
<td>0.644</td>
<td>0.663</td>
</tr>
</tbody>
</table>

Quantile: 0.900

<table>
<thead>
<tr>
<th>Lower</th>
<th>Parameter Limit</th>
<th>Upper Limit</th>
<th>Estimate Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.829</td>
<td>67.351</td>
<td>107.873</td>
</tr>
<tr>
<td>2</td>
<td>0.650</td>
<td>0.686</td>
<td>0.723</td>
</tr>
</tbody>
</table>
Example Program
Quantile Regression - Simple Interface
Engels 1857 Study of Household Expenditure on Food

\[ \tau = 0.10 \]
\[ \tau = 0.25 \]
\[ \tau = 0.50 \]
\[ \tau = 0.75 \]
\[ \tau = 0.90 \]