NAG Library Function Document

nag_simple_linear_regression (g02cac)

1 Purpose

nag_simple_linear_regression (g02cac) performs a simple linear regression with or without a constant term. The data is optionally weighted.

2 Specification

```c
#include <nag.h>
#include <nagg02.h>
void nag_simple_linear_regression (Nag_SumSquare mean, Integer n,
                     const double x[], const double y[], const double wt[],
                     double *a, double *b, double *a_serr, double *b_serr,
                     double *rsq, double *rss, double *df, NagError *fail)
```

3 Description

nag_simple_linear_regression (g02cac) fits a straight line model of the form,

\[ E(y) = a + bx, \]

where \( E(y) \) is the expected value of the variable \( y \), to the data points

\( (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n), \)

such that

\[ y_i = a + bx_i + e_i, \quad i = 1, 2, \ldots, n(n > 2). \]

where the \( e_i \) values are independent random errors. The \( i \)th data point may have an associated weight \( w_i \), these may be used either in the situation when \( \text{var}(e_i) = \sigma^2/w_i \) or if observations have to be removed from the regression by having zero weight or have been observed with frequency \( w_i \).

The regression coefficient, \( b \), and the regression constant, \( a \) are estimated by minimizing

\[ \sum_{i=1}^{n} w_i e_i^2, \]

if the weights option is not selected then \( w_i = 1.0 \).

The following statistics are computed:

- the estimate of regression constant \( \hat{a} = \bar{y} - \hat{b}\bar{x}, \)
- the estimate of regression coefficient \( \hat{b} = \sum w_i (x_i - \bar{x})(y_i - \bar{y}) \sum w_i (x_i - \bar{x})^2, \)
- the residual sum of squares \( rss = \sum w_i (y_i - \hat{y}_i)^2, \)

where the weighted means \( \bar{x} \) and \( \bar{y} \) are

\[ \bar{x} = \frac{\sum w_i x_i}{\sum w_i} \quad \text{and} \quad \bar{y} = \frac{\sum w_i y_i}{\sum w_i} \]

The number of degrees of freedom associated with \( rss \) is

\[ df = \sum w_i - 2 \quad \text{where} \quad \text{mean} = \text{Nag_AboutMean} \]

\[ df = \sum w_i - 1 \quad \text{where} \quad \text{mean} = \text{Nag_AboutZero} \]
Note: the weights should be scaled to give the correct degrees of freedom in the case \( \var{\epsilon}_i = \sigma^2/w_i \).

The \( R^2 \) value or coefficient of determination

\[
R^2 = \frac{\sum w_i(y_i - \bar{y})^2 - \text{rss}}{\sum w_i(y_i - \bar{y})^2}.
\]

This measures the proportion of the total variation about the mean \( \bar{y} \) that can be explained by the regression.

The standard error for the regression constant \( \hat{a} \)

\[
a_{\text{serr}} = \sqrt{\frac{\text{rss}}{df} \left( \frac{1}{\sum w_i} + \frac{(\bar{x})^2}{\sum w_i(x_i - \bar{x})^2} \right)} = \sqrt{\frac{\text{rss}}{df} \frac{1}{\sum w_i \sum w_i(x_i - \bar{x})^2}}.
\]

The standard error for the regression coefficient \( \hat{b} \)

\[
b_{\text{serr}} = \sqrt{\frac{\text{rss}}{df \sum w_i(x_i - \bar{x})^2}}.
\]

Similar formulae can be derived for the case when the line goes through the origin, that is \( a = 0 \).

4 References


5 Arguments

1: \textbf{mean} – Nag_SumSquare

\textit{Input}

\textit{On entry:} indicates whether \texttt{nag_simple_linear_regression (g02cac)} is to include a constant term in the regression.

\texttt{mean} = Nag_AboutMean

The regression constant \( a \) is included.

\texttt{mean} = Nag_AboutZero

The regression constant \( a \) is not included, i.e., \( a = 0 \).

\textit{Constraint:} \texttt{mean} = Nag_AboutMean or Nag_AboutZero.

2: \texttt{n} – Integer

\textit{Input}

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

\textit{Constraints:}

\begin{align*}
\text{if } & \texttt{mean} = \text{Nag_AboutMean}, \ n \geq 2; \\
\text{if } & \texttt{mean} = \text{Nag_AboutZero}, \ n \geq 1.
\end{align*}

3: \texttt{x[n]} – const double

\textit{Input}

\textit{On entry:} the values of the independent variable with the \( i \)th value stored in \( x[i-1] \), for \( i = 1, 2, \ldots, n \).

\textit{Constraint:} all the values of \( x \) must not be identical.

4: \texttt{y[n]} – const double

\textit{Input}

\textit{On entry:} the values of the dependent variable with the \( i \)th value stored in \( y[i-1] \), for \( i = 1, 2, \ldots, n \).

\textit{Constraint:} all the values of \( y \) must not be identical.
5:  wt[n] – const double           Input
On entry: if weighted estimates are required then wt must contain the weights to be used in the
weighted regression. Usually wt[i - 1] will be an integral value corresponding to the number of
observations associated with the i-th data point, or zero if the i-th data point is to be ignored. The
sum of the weights therefore represents the effective total number of observations used to create
the regression line.

If weights are not provided then wt must be set to NULL and the effective number of
observations is n.
Constraint: if wt is not NULL, wt[i - 1] = 0.0, for i = 1, 2, ..., n.

6:  a – double *                   Output
On exit: if mean = Nag_AboutMean then a is the regression constant a, otherwise a is set to zero.

7:  b – double *                   Output
On exit: the regression coefficient b.

8:  a_serr – double *              Output
On exit: the standard error of the regression constant a.

9:  b_serr – double *              Output
On exit: the standard error of the regression coefficient b.

10: rsq – double *                 Output
On exit: the coefficient of determination, $R^2$.

11: rss – double *                 Output
On exit: the sum of squares of the residuals about the regression.

12: df – double *                  Output
On exit: the degrees of freedom associated with the residual sum of squares.

13: fail – NagError*              Input/Output
The NAG error argument (see Section 3.6 in the Essential Introduction).

6   Error Indicators and Warnings

NE_BAD_PARAM
On entry, argument mean had an illegal value.

NE_INT_ARG_LT
On entry, n = <value>.
Constraint: n ≥ 1
if mean = Nag_AboutZero.
On entry, n = <value>.
Constraint: n ≥ 2
if mean = Nag_AboutMean.

NE_NEG_WEIGHT
On entry, at least one of the weights is negative.
NE_SW_LOW
On entry, the sum of elements of \( wt \) must be greater than 1.0 if \( mean = Nag\_AboutZero \) or greater than 2.0 if \( mean = Nag\_AboutMean \).

NE_WT_LOW
On entry, \( wt \) must contain at least 1 positive element if \( mean = Nag\_AboutZero \) or at least 2 positive elements if \( mean = Nag\_AboutMean \).

NE_X_OR_Y_IDEN
On entry, all elements of \( x \) and/or \( y \) are equal.

NE_ZERO_DOF_RESID
On entry, the degrees of freedom for the residual are zero, i.e., the designated number of arguments = the effective number of observations.

NW_RSS_EQ_ZERO
Residual sum of squares is zero, i.e., a perfect fit was obtained.

7 Accuracy
The computations are believed to be stable.

8 Parallelism and Performance
Not applicable.

9 Further Comments
The time taken by the function depends on \( n \). The function uses a two-pass algorithm.

10 Example
A program to calculate regression constants, \( \hat{a} \) and \( \hat{b} \), the standard error of the regression constants, the regression coefficient of determination and the degrees of freedom about the regression.

10.1 Program Text
/* nag_simple_linear_regression (g02cac) Example Program. *
* Copyright 2014 Numerical Algorithms Group.
*/
#include <nag.h>
#include <stdio.h>
#include <nag_stdlib.h>
#include <nagg02.h>
int main(void)
{
    Integer exit_status = 0, i, n;
    Nag_SumSquare mean;
    Nag_Boolean weight;
    char nag_enum_arg[40];
    double a, b, df, err_a, err_b, rsq, rss;
    double *wt = 0, *wtptr, *x = 0, *y = 0;
    NagError fail;
INIT_FAIL(fail);

printf("nag_simple_linear_regression (g02cac) Example Program Results\n");
/* Skip heading in data file */
#ifdef _WIN32
    scanf_s("%*[\n]");
#else
    scanf("%*[\n]");
#endif
#ifdef _WIN32
    scanf_s(" %39s", nag_enum_arg, _countof(nag_enum_arg));
#else
    scanf(" %39s", nag_enum_arg);
#endif
/* nag_enum_name_to_value (x04nac).
 * Converts NAG enum member name to value */
    mean = (Nag_SumSquare) nag_enum_name_to_value(nag_enum_arg);
#ifdef _WIN32
    scanf_s(" %39s", nag_enum_arg, _countof(nag_enum_arg));
#else
    scanf(" %39s", nag_enum_arg);
#endif
    weight = (Nag_Boolean) nag_enum_name_to_value(nag_enum_arg);
#ifdef _WIN32
    scanf_s("%"NAG_IFMT", &n);
#else
    scanf("%"NAG_IFMT", &n);
#endif
if (n >= (mean == Nag_AboutMean?2:1))
{
    if (!(x = NAG_ALLOC(n, double)) ||
        !(y = NAG_ALLOC(n, double)) ||
        !(wt = NAG_ALLOC(n, double)))
    {
        printf("Allocation failure\n");
        exit_status = -1;
        goto END;
    }
} else
{
    printf("Invalid n.\n");
    exit_status = 1;
    return exit_status;
}
if (weight)
{
    wtptr = wt;
    for (i = 0; i < n; ++i)
#ifdef _WIN32
        scanf_s("%lf%lf%lf", &x[i], &y[i], &wt[i]);
#else
        scanf("%lf%lf%lf", &x[i], &y[i], &wt[i]);
#endif
} else
{
    wtptr = (double *) 0;
    for (i = 0; i < n; ++i)
#ifdef _WIN32
        scanf_s("%lf%lf", &x[i], &y[i]);
#else
        scanf("%lf%lf", &x[i], &y[i]);
#endif
    /* nag_simple_linear_regression (g02cac)."
* Simple linear regression with or without a constant term,
* data may be weighted
*/

call nag_simple_linear_regression(mean, n, x, y, wtptr, &a, &b, &err_a, &err_b,
&rsq, &rss, &df, &fail);
if (fail.code != NE_NOERROR)
{
    printf("Error from nag_simple_linear_regression (g02cac).\n%s\n",
fail.message);
    exit_status = 1;
    goto END;
}

if (mean == Nag_AboutMean)
{
    printf("\nRegression constant a = %6.4f\n\n", a);
    printf("Standard error of the regression constant a = %6.4f\n\n",
    err_a);
}

printf("Regression coefficient b = %6.4f\n\n", b);
printf("Standard error of the regression coefficient b = %6.4f\n\n",
err_b);

printf("The regression coefficient of determination = %6.4f\n\n", rsq);
printf("The sum of squares of the residuals about the "
"regression = %6.4f\n\n", rss);
printf("Number of degrees of freedom about the "
"regression = %6.4f\n\n", df);

END:
NAG_FREE(x);
NAG_FREE(y);
NAG_FREE(wt);

return exit_status;

10.2 Program Data
nag_simple_linear_regression (g02cac) Example Program Data
Nag_AboutMean Nag_TRUE
B
1.0 20.0 1.0
0.0 15.5 1.0
4.0 28.3 1.0
7.5 45.0 1.0
2.5 24.5 1.0
0.0 10.0 1.0
10.0 99.0 1.0
5.0 31.2 1.0

10.3 Program Results
nag_simple_linear_regression (g02cac) Example Program Results

Regression constant a = 7.5982
Standard error of the regression constant a = 6.6858
Regression coefficient b = 7.0905
Standard error of the regression coefficient b = 1.3224
The regression coefficient of determination = 0.8273
The sum of squares of the residuals about the regression = 965.2454
Number of degrees of freedom about the regression = 6.0000