AN IMPROVED MULTIPLE LINEAR REGRESSION AND DATA ANALYSIS COMPUTER PROGRAM PACKAGE

by Steven M. Sidik

Lewis Research Center
Cleveland, Ohio 44135

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION - WASHINGTON, D. C. - APRIL 1972
**Title and Subtitle**
AN IMPROVED MULTIPLE LINEAR REGRESSION AND DATA ANALYSIS COMPUTER PROGRAM PACKAGE

**Author(s)**
Steven M. Sidik

**Performing Organization Name and Address**
Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio 44135

**Sponsoring Agency Name and Address**
National Aeronautics and Space Administration
Washington, D.C. 20546

**Abstract**
NEWRAP, an improved version of a previous multiple linear regression program called RAPIER, CREDUC, and CRSPLT, allows for a complete regression analysis including cross plots of the independent and dependent variables, correlation coefficients, regression coefficients, analysis of variance tables, t-statistics and their probability levels, rejection of independent variables, plots of residuals against the independent and dependent variables, and a canonical reduction of quadratic response functions useful in optimum seeking experimentation. A major improvement over RAPIER is that all regression calculations are done in double precision arithmetic.

**Key Words (Suggested by Author(s))**
Regression analysis; Multiple linear regression; Least squares; Curve fitting; Response surface analysis

**Distribution Statement**
Unclassified - unlimited

**For sale by the National Technical Information Service, Springfield, Virginia 22151**
## CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>2</td>
</tr>
<tr>
<td>SYMBOLS</td>
<td>3</td>
</tr>
<tr>
<td>ESTIMATION OF BASIC LINEAR MODEL</td>
<td>5</td>
</tr>
<tr>
<td>BASIC LINEAR MODEL</td>
<td>5</td>
</tr>
<tr>
<td>ESTIMATING ( b )</td>
<td>8</td>
</tr>
<tr>
<td>CORRELATION MATRIX</td>
<td>11</td>
</tr>
<tr>
<td>ESTIMATING ( \sigma^2 )</td>
<td>11</td>
</tr>
<tr>
<td>HYPOTHESIS TESTING</td>
<td>13</td>
</tr>
<tr>
<td>NORMALITY OF ( \epsilon )</td>
<td>13</td>
</tr>
<tr>
<td>ANALYSIS OF VARIANCE TABLE</td>
<td>14</td>
</tr>
<tr>
<td>CHOICE OF ESTIMATOR FOR ( \sigma^2 )</td>
<td>15</td>
</tr>
<tr>
<td>TEST OF OVERALL REGRESSION</td>
<td>16</td>
</tr>
<tr>
<td>( t )-TESTS</td>
<td>17</td>
</tr>
<tr>
<td>PREDICTING VALUES FROM ESTIMATED REGRESSION EQUATION</td>
<td>18</td>
</tr>
<tr>
<td>NEWRAP PROGRAM</td>
<td>19</td>
</tr>
<tr>
<td>USERS GUIDE TO NEWRAP INPUT</td>
<td>19</td>
</tr>
<tr>
<td>Illustrative Regression Problem Requiring No Transformations</td>
<td>19</td>
</tr>
<tr>
<td>Detailed Description of Input Cards</td>
<td>21</td>
</tr>
<tr>
<td>Illustrative Problem Requiring Transformations</td>
<td>31</td>
</tr>
<tr>
<td>NEWRAP DOCUMENTATION AND LISTINGS</td>
<td>40</td>
</tr>
<tr>
<td>General Mathematical and Logical Flow of Program</td>
<td>40</td>
</tr>
<tr>
<td>Routines and Their Major Functions</td>
<td>42</td>
</tr>
<tr>
<td>Call Structure of Program</td>
<td>42</td>
</tr>
<tr>
<td>Dictionary of Program</td>
<td>43</td>
</tr>
<tr>
<td>Program Listing</td>
<td>46</td>
</tr>
<tr>
<td>CRSPLT PROGRAM</td>
<td>76</td>
</tr>
<tr>
<td>CREDUC PROGRAM</td>
<td>85</td>
</tr>
<tr>
<td>APPENDIX - BORROWED ROUTINES</td>
<td>91</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>94</td>
</tr>
</tbody>
</table>
AN IMPROVED MULTIPLE LINEAR REGRESSION AND DATA
ANALYSIS COMPUTER PROGRAM PACKAGE

by Steven M. Sidik
Lewis Research Center

SUMMARY

NEWRAP is a digital computer program which can be used with ease to perform extensive regression analyses or a simple least-squares curve fit. The program is written in FORTRAN IV, version 13, for the IBM 7094/7044 DCS. The major value of the program is the comprehensiveness of its calculations and options.

NEWRAP computes the variance-covariance matrix of the independent variables, regression coefficients, t-statistics for individual tests, and analysis of variance tables for overall testing of regression. There is a provision for a choice of three strategies for the variance estimate to be used in computing t-statistics.

Also, more than one set of responses of dependent variables can be analyzed for the same set of independent variables.

A backward rejection option method based on the first dependent variable may be used to delete nonsignificant terms from the model. In this case, a critical significance level is supplied as input. The least significant independent variable is deleted and the regression recomputed. This process is repeated until all remaining variables have significantly nonzero coefficients.

The NEWRAP program uses the triangular form of symmetric matrices throughout. It also allows for the use of weighted regression, computation of predicted values at any combination of independent variables, a table of residuals, and plots of residuals.

By use of CRSPLT, a preregression analysis may be performed which may aid in the choice of model to use in NEWRAP. This program accepts the same raw data in the same format and computes the variance-covariance matrix and correlation matrix of all the variables and an eigenvector decomposition of the variance-covariance matrix corresponding to the independent variables. Microfilm plots are then printed of specified pairs of variables. Punched output of residuals and predicted values from NEWRAP can also be used for more complicated residual plots than the direct use of the plotting option NEWRAP permits.

When a quadratic response function has been estimated (as for example in optimum-seeking experimentation) CREDUC may be used to obtain all information necessary for a canonical analysis of the function.

The three programs together provide a useful data analysis package that can be applied to a large variety of common research and development situations.
INTRODUCTION

RAPIER (ref. 1) is a very flexible multiple linear regression analysis computer program which has been in frequent use at the NASA Lewis Research Center. It was tested with the data presented in Wampler (ref. 2) and performed quite poorly. This alone was not very disturbing since real data are seldom even nearly as ill-conditioned as that set of data. A second factor, however, is that Wampler's data leads to a 5 by 5 matrix to be inverted whereas RAPIER is designed to handle matrices of up to 60 by 60. With real data it is not uncommon for the matrix to become more ill-conditioned as the dimension increases. Often the user increases the size of the model by adding terms which are functions of the original independent variables (as for example in polynomial models) and this often leads to increased correlations and ill-conditioning. For this reason, RAPIER was modified primarily by rearranging the storage of variables in COMMON blocks and performing all the regression calculations in double precision. This was done without losing any of the capabilities of the original program (in fact adding new options). The resulting version is called NEWRAP.

It may be of interest to some RAPIER users that in a number of sample calculations the major numerical inaccuracies arising in the regression calculations were not involved in the actual inversion of the $X'X$ matrix but in the calculation of the inner products which give

$$\hat{b} = (X'X)^{-1}(X'y)$$

Thus a major improvement might be made by computing inner products in double precision arithmetic and truncating to single precision answers without going to complete double precision arithmetic although the latter alternative would further increase the accuracy. As a matter of fact, the double precision inner product calculation is used in a different least-squares method proposed by Golub (ref. 3) which is reference 19 of Wampler's paper.

It should be pointed out that in estimation problems an alternative to the obvious step of more accurate routines is provided by Hoerl and Kennard (ref. 4). They present a technique called "ridge regression" which uses the method of minimum mean squared error estimation in place of minimum variance unbiased estimation. The ridge regression technique should have some definite appeal to statisticians, because it recognizes the fact that existence of ill-conditioned data indicates a problem which should be accounted for statistically as well as computationally. They do consider the problem of rejecting terms but their methods are not amenable to incorporation in NEWRAP in its present form.

A second reason for modifying the program was the desire to provide plots of the residuals as was strongly recommended in chapter 3 of Draper and Smith (ref. 5). With
the microfilm plotting capabilities provided by CINEMATIC (ref. 6) available at the Lewis Research Center computer facility, this feature was also added to NEWRAP without significantly increasing printed output. CINEMATIC is a very specialized set of routines for the 7094/7044 DCS and 360/67 systems. If microfilm plotting is not available at other computer installations, the subroutines used in plotting may readily be changed to routines which produce line printer plots or CALCOMP plots however.

The RAPIER program used an algorithm for the coefficient calculations that inverted the correlation matrix and then converted this to the \((X'X)^{-1}\) matrix to calculate \(\hat{b}\). After inspection of several test cases, it seemed that this method did not improve the accuracy of the calculation of \((X'X)^{-1}\). Thus it was dropped and NEWRAP inverts \(X'X\) directly.

As with the RAPIER report, only the statistics and mathematics necessary to explain the program capabilities will be presented along with illustrative input and output listings and listings of the programs.

SYMBOLS

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>matrix</td>
</tr>
<tr>
<td>b</td>
<td>vector (column)</td>
</tr>
<tr>
<td>(b_i)</td>
<td>true regression coefficient</td>
</tr>
<tr>
<td>(\hat{b}_i)</td>
<td>estimated regression coefficient</td>
</tr>
<tr>
<td>(b_0)</td>
<td>constant term</td>
</tr>
<tr>
<td>(b_1, \ldots, b_J)</td>
<td>unknown parameters</td>
</tr>
<tr>
<td>C</td>
<td>correlation matrix</td>
</tr>
<tr>
<td>(C_{ij})</td>
<td>elements of (C)</td>
</tr>
<tr>
<td>D</td>
<td>indicator variable, equal to 0 if no (b_0) coefficient is estimated and equal to 1 if (b_0) is estimated</td>
</tr>
<tr>
<td>(E(x))</td>
<td>expected value of (x) (i.e., mean of (x) over all possible values of (x))</td>
</tr>
<tr>
<td>e</td>
<td>vector of observed values minus predicted values</td>
</tr>
<tr>
<td>(F_{a,d})</td>
<td>statistic distributed as variance ratio with (a) and (d) degrees of freedom</td>
</tr>
<tr>
<td>(f_j(z_1, \ldots, z_K))</td>
<td>term of regression equation</td>
</tr>
<tr>
<td>(H_0)</td>
<td>statistical hypothesis to be tested</td>
</tr>
</tbody>
</table>
$H_1$ alternate hypothesis to be accepted if $H_0$ is judged to be false

$J$ number of coefficients estimated, excluding $b_0$

$K$ number of independent variables observed

$k$ number of segments or cells in range of possible studentized residuals

$LOF$ lack of fit

$M$ total number of independent and dependent variables

$MS(source)$ mean square due to source, where source is REG, RES, etc.

$N$ number of observations

$NPDEG$ pooled degrees of freedom for replication error

$N(\mu, \sigma^2)$ normal distribution with mean $\mu$ and variance $\sigma^2$

$R$ number of sets of replicates

$REG$ regression

$REP$ replication

$RES$ residual

$r_i$ number of replicates in set $i$

$S$ diagonal matrix

$S_c$ sum of squares correction if $D = 1$, and 0 if $D = 0$

$SSQ(source)$ sum of squares due to source, where source is REG, RES, etc.

$s_j$ elements of diagonal matrix

$TOT$ total

$t_n$ statistic distributed as Student's $t$ with $n$ degrees of freedom

$V(x)$ variance of $x$, expected value of $(x - E(x))^2$

$W, X$ matrices

$w, x$ vectors (column)

$X_s$ stationary point of estimated quadratic surface

$x(J)$ $x^J$

$\bar{x}_{.j} = \frac{1}{N} \sum_{i=1}^{N} x_{ij}$

$y$ vector (column)

$Z_i$ studentized residual

4
\( z_1, \ldots, z_K \) \hspace{1cm} \text{variables}
\( \epsilon \) \hspace{1cm} \text{vector of observation errors}
\( \mu_x \) \hspace{1cm} \text{mean of } x \text{ defined as } E(x)
\( \hat{\mu} \) \hspace{1cm} \text{estimate of } \mu \text{ based on observation of random sample}
\( \sigma^2_x \) \hspace{1cm} \text{variance of } x \text{ defined as } V(x)
\( \hat{\sigma}^2 \) \hspace{1cm} \text{estimate of } \sigma^2 \text{ based on observation of random sample}

Superscript:
\( \top \) \hspace{1cm} \text{transpose}

**ESTIMATION OF BASIC LINEAR MODEL**

**BASIC LINEAR MODEL**

In multiple linear regression, a dependent or response variable \( Y \) (such as temperature or pressure) measured on an object or experiment is assumed to be correlated with a function of one or more other variables \( (z_1, \ldots, z_K) \) measured on the same object or experiment. This function includes a number of unknown parameters \( (b_1, \ldots, b_J) \) and can be represented as

\[
y = h(b_1, \ldots, b_J, z_1, \ldots, z_K) + \epsilon
\]

The only restriction imposed on this function is that it be linear in the parameters; that is, the function is of the form

\[
y = \sum_{j=1}^{J} b_j f_j(z_1, \ldots, z_K) + \epsilon
\]

where \( f_j(z_1, \ldots, z_K) \) is a TERM of the regression equation. (A TERM is a quantity which may be a variable or a function of a variable, e.g., \( T \) is a TERM and \( Z \), after it is defined as \( Z = \log T \), is also a TERM.)

Suppose that there are \( N \) observations of the dependent variable. Let the subscript \( i \) indicate that the values are associated with the \( i^{th} \) observation; in particular, the value of the response variable \( y_i \) would depend on the observed values of the variables \( (z_{i1}, \ldots, z_{iK}) \). Also, let the subscript \( j \) denote the \( j^{th} \) term in the regression
model so that \( x_{ij} = f_j(z_{i1}, \ldots, z_{iK}) \) describes the transformations of the \( z_{i1}, \ldots, z_{iK} \) to produce the value of \( x_{ij} \) for the \( j \)th term at the \( i \)th observation.

The regression model can now be rewritten as

\[
y_i = b_1 x_{i1} + b_2 x_{i2} + \ldots + b_J x_{iJ} + \epsilon_i \quad i = 1, \ldots, N
\]

where \( \epsilon_i \) denotes the difference between the observed value and the expected value of \( y_i \). For the \( N \) observations, it is convenient to write this regression model in matrix notation as \( y = Xb + \epsilon \) where

\[
y = \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix}, \\
X = \begin{pmatrix} x_{11} & \cdots & x_{1J} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{NJ} \end{pmatrix}, \\
b = \begin{pmatrix} b_1 \\ \vdots \\ b_J \end{pmatrix}, \\
\epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_N \end{pmatrix}
\]
More often than not, the analyst feels the following model is more appropriate:

\[ y_i = b_0 + b_1 x_{i1} + \ldots + b_J x_{iJ} + \epsilon_i \quad i = 1, \ldots, N \]  

(5)

Let \( a_0 = b_0 + b_1 \bar{x}_{1,1} + \ldots + b_J \bar{x}_{J,1} \). Then, the result of adding this equation to and subtracting it from equation (5) and rearranging the terms is

\[ y_i = (b_0 + b_1 \bar{x}_{1,1} + \ldots + b_J \bar{x}_{J,1}) \]

\[ + b_1 (x_{i1} - \bar{x}_{1,1}) + \ldots + b_J (x_{iJ} - \bar{x}_{J,1}) + \epsilon_i \quad i = 1, \ldots, N \]  

(6)

If, then, a dummy variable \( x_0 \) is introduced such that, for all values of \( i \), \( x_{i0} = 1.0 \), equation (6) may be written as

\[ y_i = a_0 x_{i0} + b_1 (x_{i1} - \bar{x}_{1,1}) + \ldots + b_J (x_{iJ} - \bar{x}_{J,1}) + \epsilon_i \quad i = 1, \ldots, N \]  

(6a)

Equation (6a) now resembles equation (3) and may be written in matrix notation, similar to equation (4), as \( y = Xb + \epsilon \) where now

\[ y = \begin{pmatrix} \hat{y}_1 \\ \vdots \\ \hat{y}_N \end{pmatrix} \]

\[ x = \begin{pmatrix} 1 & x_{11} - \bar{x}_{1,1} & \ldots & x_{1J} - \bar{x}_{J,1} \\ 1 & x_{21} - \bar{x}_{1,1} & \ldots & x_{2J} - \bar{x}_{J,1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{N1} - \bar{x}_{1,1} & \ldots & x_{NJ} - \bar{x}_{J,1} \end{pmatrix} \]

\[ b = \begin{pmatrix} a_0 \\ b_1 \\ \vdots \\ b_J \end{pmatrix} \]

\[ \epsilon = \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_N \end{pmatrix} \]  

(7)
ESTIMATING $b$

Equations (4) and (7) are similar in form and for $N > J$ are an overdetermined set of linear equations. There will be some vector $\hat{b}$ which is a "best" vector to use. If the vector $\epsilon$ is composed of random variables $\epsilon_i$ such that $E(\epsilon_i) = 0$, $V(\epsilon_i) = \sigma^2 < +\infty$, and the $\epsilon_i$ are uncorrelated, then as is well known, the method of least squares gives the linear minimum variance unbiased estimators $\hat{b}$ for $b$. And $\hat{b}$ is given by

$$\hat{b} = (X'X)^{-1}X'y$$  (8)

The matrix $X'X$ divided by $N - 1$ is called the moment matrix of the experiment. The variance-covariance matrix of $\hat{b}$ is given by

$$V(\hat{b}) = \sigma^2(X'X)^{-1}$$  (9)

It is important to note that when the form of equation (7) is used, $X'X$ is given by

$$X'X = \begin{pmatrix}
N & 0 & \ldots & 0 \\
0 & \sum_{i} (x_{i1} - \bar{x}_1)^2 & \ldots & \sum_{i} (x_{i1} - \bar{x}_1)(x_{iJ} - \bar{x}_J) \\
\vdots & \vdots & \ddots & \vdots \\
0 & \sum_{i} (x_{i1} - \bar{x}_1)(x_{iJ} - \bar{x}_J) & \ldots & \sum_{i} (x_{iJ} - \bar{x}_J)^2
\end{pmatrix}$$  (10)

This is seen to be symmetric and of the form

$$X'X = \begin{pmatrix}
A & 0 \\
0 & B
\end{pmatrix}$$

Hence,

$$(X'X)^{-1} = \begin{pmatrix}
A^{-1} & 0 \\
0 & B^{-1}
\end{pmatrix}$$
NEWRAP uses this relation to advantage by storing only the lower triangular part of $B$ and computing only the coefficients $b_1, \ldots, b_J$ by matrix manipulations. Then $b_0$ is given by the simple equation

$$b_0 = \bar{y} - \hat{b}_1 \bar{x}_1 - \hat{b}_2 \bar{x}_2 - \cdots - \hat{b}_J \bar{x}_J$$

where $\bar{y} = \sum y_i / N = \hat{a}_0$. It can also be shown that

$$V(\hat{b}_0) = V(\bar{y}) + V(\hat{b}'\bar{x}) = \left[ \frac{1}{N} + \bar{x}'(X'X)^{-1} \bar{x} \right] \sigma^2$$

$$\text{COV}(\hat{b}_0, \hat{b}) = -(X'X)^{-1} \bar{x} \sigma^2$$

When there is no $b_0$ term in the regression model,

$$X'X = \begin{bmatrix} \Sigma_{i=1}^{N} x_{i1}^2 & \Sigma_{i=1}^{N} x_{i1} x_{i2} & \cdots & \Sigma_{i=1}^{N} x_{i1} x_{iJ} \\ \\
\vdots & \vdots & & \vdots \\ \\
\Sigma_{i=1}^{N} x_{i2} x_{iJ} & \Sigma_{i=1}^{N} x_{i2} x_{iJ} & \cdots & \Sigma_{i=1}^{N} x_{i2}^2 \end{bmatrix}$$

Comparing this to equation (10) shows this form of $X'X$ to be similar to the lower right submatrix in equation (10). This similarity is used to simplify notation by assuming that $X'X$ represents either the form of equation (12) or the lower right portion of equation (10) and considering the calculation of $b_0$ as a special case. Thus, further reference to $b$ implies

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_J \end{pmatrix}$$

There are two different methods of computing the regression coefficients which may
be used in NEWRAP. The first method uses bordering (ref. 7) on the full \( X'X \) matrix. If \( X'X \) is a nearly singular matrix, there may be problems with accuracy resulting in overflows or underflows causing execution to terminate without any results being printed. The second method uses a method of bordering which enters one term at a time into the model equation. After each term is entered, a full regression analysis is printed. Typically, if \( X'X \) is nearly singular, a number of terms will have been added to the model before the results become unreliable or cause execution to be terminated. Thus, at least a partial analysis of the full model is available to aid in selection of further models to submit. After all the terms have been entered, the program then switches to the procedure which inverts the appropriate full \( X'X \) matrix at each stage for further analyses.

The use of the bordering method leads to a large volume of printed output and is not recommended as a standard procedure. Through use of CRSPLT as a preregression analysis program it may be easier to determine if bordering should be used. CRSPLT can also help indicate the order of arrangement of the terms of the model so that those thought to be most important can be entered into the model first.

Also note that the individual observations may be weighted to perform a weighted regression analysis. NEWRAP permits the use of weights (ref. 5). In this case, the \( X'X \) and \( X'y \) matrices take the following form:

\[
X'X = \begin{pmatrix}
\sum_{i=1}^{N} [(x_{i1} - \bar{x}_{1})^2 w_i] & \cdots & \sum_{i=1}^{N} [w_i(x_{i1} - \bar{x}_{1})(x_{iJ} - \bar{x}_{J})] \\
\vdots & \ddots & \vdots \\
\sum_{i=1}^{N} [(x_{iJ} - \bar{x}_{J})(x_{i1} - \bar{x}_{1})w_i] & \cdots & \sum_{i=1}^{N} [(x_{iJ} - \bar{x}_{J})^2 w_i] \\
\end{pmatrix}
\]

\[
X'y = \begin{pmatrix}
\sum_{i=1}^{N} x_{i1}y_i w_i \\
\vdots \\
\sum_{i=1}^{N} x_{ij}y_i w_i \\
\end{pmatrix}
\]
CORRELATION MATRIX

Another matrix of interest both computationally and statistically in the correlation matrix $C$. The elements of $C$, which are denoted $C_{ij}$, are the sample correlation coefficients between the terms $X_i$ and $X_j$. These are

$$C_{ij} = \sum_{l=1}^{N} \frac{(x_{l1} - \bar{x}_1)(x_{lj} - \bar{x}_j)}{\sqrt{\sum_{k=1}^{N} (x_{ki} - \bar{x}_i)^2 \sum_{k=1}^{N} (x_{kj} - \bar{x}_j)^2}}$$

(13)

and all these numbers are between 1.0 and -1.0.

The calculation of $C$ can be expressed in matrix notation conveniently by defining a diagonal matrix $S = \text{diag}(s_1, s_2, \ldots, s_J)$ with elements

$$s_j = \frac{1.0}{\sqrt{(X'X)_{jj}}} \quad j = 1, \ldots, J$$

(14)

Then

$$C = S(X'X)S$$

(15)

It may also be that the independent variables are random variables. Then $X'X$ divided by $N - 1$ represents the sample variance-covariance matrix and $C$ the sample correlation matrix. If the independent variables are considered to be from a multivariate distribution, it is useful in some cases to consider the eigenvalues and eigenvectors of $X'X$. For these reasons, NEWRAP includes options to compute and print these quantities. These may also be computed and printed through use of the CRSPLT program.

ESTIMATING $\sigma^2$

For any regression model $y = Xb + \epsilon$, there are possibly two methods of estimating $\sigma^2$. First, if the assumed regression model is, in reality, the true model, it is well known that an unbiased estimator is given by
\[ \hat{\sigma}^2_{\text{RES}(j)} = \frac{y'y - \hat{b}'x'y}{N - J - D} \]
\[ = \frac{\text{SSQ(RES)}}{N - J - D} \]
\[ = \text{MS(RES(j))} \] (16)

Second, where there are replicated data points, another estimator of \( \sigma^2 \), depending only on \( V(\epsilon_i) = \sigma^2 \) for all \( i \) and not on the validity of the assumed model, is the pooled mean squares computed from the replicated data points.

Assume the observations are grouped into replicate sets in sequence. Let \( R \) be the number of sets of replicates and \( r_i \) be the number of replicates in the \( i^{\text{th}} \) replicate set. Let

\[ \text{SSQ}(i) = \sum_{k=r^*+1}^{r^*+r_i} (y_k - \bar{y}_i)^2 \] (17)

where

\[ r^* = \sum_{j=1}^{i-1} r_j \]

It is assumed \( y_n \) is from the \( i^{\text{th}} \) replicate set and \( \bar{y}_i \) is calculated only from those \( y_n \) in the \( i^{\text{th}} \) replicate set. Then define the pooled sum of squares due to replication as

\[ \text{SSQ(REP)} = \sum_{i=1}^{R} \text{SSQ}(i) \]

and the pooled degrees of freedom as \( \text{NPDEG} = \sum_{i=1}^{R} (r_i - 1) \). The second estimator of \( \sigma^2 \) becomes

\[ \sigma^2_{\text{REP}} = \frac{\text{SSQ(REP)}}{\text{NPDEG}} \]
\[ = \text{MS(REP)} \] (18)

It can be shown (ref. 5, p. 26) that the sums of squares due to residuals can be partitioned into a component due to replication and a component due to lack of fit; that is,

\[ \text{SSQ(RES)} = \text{SSQ(LOF)} + \text{SSQ(REP)} \] (19)
This partitioning is used later to determine the estimate of $\sigma^2$ to use in tests of hypotheses.

**HYPOTHESIS TESTING**

**NORMALITY OF $\epsilon$**

As stated before, the only assumption necessary for $\hat{\beta}$ to be a linear minimum variance unbiased estimator is that $E(\epsilon_i) = 0.0$, $V(\epsilon_i) = \sigma^2 < +\infty$, and $\epsilon_i$ be uncorrelated. If it can further be assumed that $\epsilon_i \sim N(0, \sigma^2)$, a number of standard tests become available. NEWRAP computes a chi-squared statistic which can be used as an approximate test.

Under the hypothesis $\epsilon_i \sim N(0, \sigma^2)$, the studentized residuals defined by

$$Z_i = \frac{y_i - \hat{y}_i}{\hat{\sigma}} = \frac{e_i}{\hat{\sigma}}$$

will be distributed as Student's $t$ with the degrees of freedom associated with the estimate $\sigma$. If the degree of freedom is 30 or more, the $t$ distribution is very close to the normal.

The range of possible studentized residuals is $(-\infty, +\infty)$ and may be divided into $k$ segments or cells each with probability $p_i$, so that each segment will have $Np_i$ as the expected number of observations falling into it. Let $n_i$ denote the number of studentized residuals in the $i^{th}$ cell. Then a chi-squared goodness-of-fit statistic may be calculated as

$$\chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - Np_i)^2}{Np_i}$$

NEWRAP computes this statistic by using an even number of cells greater than or equal to four and less than or equal to 20, such that the expected numbers of observations per cell is five or more. This statistic is not computed when there are less than 20 observations. The bounding values for the $i^{th}$ cell are $Z_{i-1}, Z_i$ where $F(Z_i) = (i \cdot k)/N$ and $F(Z)$ is the cumulative normal distribution function. Then each cell has the same expected number of observations, say $f = N/k$. Then
There is a point to be made concerning the chi-squared calculations. The validity of the use of the chi-squared statistic in a test depends upon the residuals forming a sample of independent and identically distributed random variables. This is not usually the case for regression residuals. Although the tail probabilities of the chi-squared tests might be in error, they should still be able to tell the statistician whether one intended normalizing transformation was more successful than another.

**ANALYSIS OF VARIANCE TABLE**

For most hypothesis testing of the regression model, it is convenient to summarize the available information in an Analysis of Variance (ANOVA) table, as follows:

<table>
<thead>
<tr>
<th>Source</th>
<th>Sums of squares</th>
<th>Degrees of freedom</th>
<th>Mean squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regression (REG)</td>
<td>SSQ(REG) = b'X'y - S_c^a</td>
<td>J</td>
<td>MS(REG) = SSQ(REG) J</td>
</tr>
<tr>
<td>Residual (RES)</td>
<td>SSQ(RES) = y'y - b'X'y</td>
<td>N - J - D^b</td>
<td>MS(RES) = SSQ(RES) (N - J - D)</td>
</tr>
<tr>
<td>Total</td>
<td>SSQ(TOT) = y'y - S_c</td>
<td>N - D</td>
<td></td>
</tr>
</tbody>
</table>

\[ S_c = \begin{cases} 
0 & \text{if no } b_0 \text{ coefficient is estimated.} \\
Ny^2 & \text{if a } b_0 \text{ coefficient is estimated.} 
\end{cases} \]

\[ b_D = \begin{cases} 
0 & \text{if no } b_0 \text{ coefficient is estimated.} \\
1 & \text{if } b_0 \text{ is estimated.} 
\end{cases} \]

If there are replicated data points, another ANOVA table can be constructed to show the separation of the residual sums of squares into components from lack of fit and replication, as in the following table:

\[ \chi^2_{k-1} = \sum_{i=1}^{k} \frac{(n_i - f)^2}{f} = \frac{k}{N} \sum_{i=1}^{k} n_i^2 - N \]
Sums of squares

<table>
<thead>
<tr>
<th>Source</th>
<th>Sums of squares</th>
<th>Degrees of freedom</th>
<th>Mean squares</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of fit</td>
<td>SSQ(LOF) = SSQ(RES) - SSQ(REP)</td>
<td>N - J - D - NPDEG</td>
<td>MS(LOF) = SSQ(LOF) / (N - J - D - NPDEG)</td>
</tr>
<tr>
<td>(LOF)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replication</td>
<td>SSQ(REP)</td>
<td>NPDEG</td>
<td>MS(REP) = SSQ(REP) / NPDEG</td>
</tr>
<tr>
<td>(REP)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Residual</td>
<td>y'y - b'X'y</td>
<td>N - J - D</td>
<td></td>
</tr>
<tr>
<td>(RES)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CHOOSE OF ESTIMATOR FOR $\sigma^2$

As mentioned previously, there are two possible methods of estimating $\sigma^2$ depending on whether there are replicated data points. This is true for any given model equation. When the backward rejection option of NEWRAP is used, there is no longer one hypothetical model but a series of models. Thus, there is the choice of estimator for $\sigma^2$ to be made after each rejection of a term in the previous model.

As an example, consider the model

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \epsilon$$  \hspace{1cm} (20)

with replicated data points. The first step is to estimate $b_0$, $b_1$, $b_2$, and $b_3$. There will then be the estimators $\hat{\sigma}^2_{\text{RES}(J)}$ and $\hat{\sigma}^2_{\text{REP}}$. If the model in equation (20) has not left out any important terms, $\hat{\sigma}^2_{\text{RES}(J)}$ as well as $\hat{\sigma}^2_{\text{REP}}$ is a valid estimator.

The ratio $F = MS(LOF)/MS(REP)$ can be used to test the hypothesis that there is no lack of fit, where $F \sim F_{a, d}$ with $a = N - J - D - NPDEG$ and $d = NPDEG$ degrees of freedom. If the test accepts the hypothesis of no lack of fit, $MS(RES)$ is a pooled estimate of $\sigma^2$ with more degrees of freedom. But there is the possibility that the hypothesis was accepted as a result of random fluctuation when there really is some lack of fit; that is, there is the possibility that $\hat{\sigma}^2_{\text{RES}(J)}$ is a biased estimator. If lack of fit is not concluded to be significant, the decision to pool or not is usually made on the basis of the number of degrees of freedom for replication. If this is "large" (no definition of large is given herein), $\hat{\sigma}^2_{\text{REP}}$ is used. If "small," the pooled estimate $\hat{\sigma}^2_{\text{RES}(J)}$ is used.

In testing equation (20), should it be decided that $b_3$ is not significantly different from zero (see section t-TESTS), the coefficients of the following model would be estimated:

$$y = b_0 + b_1 x_1 + b_2 x_2 + \epsilon$$
From this model there is an estimate \( \hat{\sigma}^2_{\text{RES}(J-1)} \). This estimate could also be biased since \( b_3 \) may be small but nonzero and the decision of \( b_3 = 0 \) may have been due to the low power of the test.

At the first step, the lack of fit can be considered a random sample of an infinite possibility of biases. But the biases due to pooling mean squares after rejecting terms can be considered to be systematic biasing. In such a case the use of Cochran's test for "the largest of a set of estimated variances as a fraction of their total" might be appropriate.

NEWRAP provides three strategies of pooling estimates for use in the decision procedure:

1. Never pool. This is usable only when there are replicated data points. The estimator used in all t-tests is \( \hat{\sigma}^2_{\text{REP}} \).
2. Pool initial residual. This will pool the lack of fit and replication (if any) from the first model only. Additional mean squares due to rejected terms will be ignored.
3. Always pool. This strategy will always use \( \hat{\sigma}^2_{\text{RES}(J-i)} \) for the model with \( i \) rejected terms.

Wherever a \( \hat{\sigma} \) or \( \hat{\sigma}^2 \) is indicated, NEWRAP always uses the value calculated according to the strategy chosen by the user.

**TEST OF OVERALL REGRESSION**

One of the first tests usually applied to a regression model is the test of the overall significance of the model. In the notation of hypothesis testing this is stated \( H_0: b = 0; \ H_1: b \neq 0 \) where

\[
b = \begin{pmatrix} b_1 \\ \vdots \\ b_J \end{pmatrix}
\]

The statistic for this test is \( F = \text{MS(REG)}/\hat{\sigma}^2 \). The \( F \sim F_{a, d} \) with \( a = J - D \), and \( d \) equals the degrees of freedom associated with \( \hat{\sigma}^2 \).

Another useful statistic for judging the significance of overall regression is \( R^2 = \text{SSQ(REG)}/\text{SSQ(TOT)} \). The sampling distribution of \( R \) does not lend itself to very simple tests except in the case of \( H_0: b = 0 \). The main value of \( R^2 \) is that it is a
number in the range 0 to 1 and 100 $R^2$ is a measure of the percentage of variation in the $y$ values that is accounted for by the regression model.

**t-TESTS**

In many cases, the regression model contains terms whose estimated coefficients are "small." This may be an indication that the term does not have a real effect on the dependent variable and that the estimate is nonzero due to random sampling variation. If this is true, it is desirable to delete the term from the regression model. A test statistic for deciding this is

$$t = \frac{\hat{b}_i}{\sqrt{\hat{\sigma}^2 (X'X)^{-1}}_{ii}}$$

(21)

where $(X'X)^{-1}_{ii}$ denotes the $i^{th}$ diagonal element of the $(X'X)^{-1}$ matrix. The statistic $t \sim t_{N-J-D}$. An equivalent test statistic is

$$F = t^2 = \frac{\hat{b}_i^2}{\hat{\sigma}^2 (X'X)^{-1}}_{ii}$$

(22)

where $F \sim F_{1, N-J-D}$. This is often referred to (ref. 5) as the partial F-test. The quantity $\hat{b}_i^2/[(X'X)^{-1}]_{ii}$ is called the additional sum of squares due to $b_i$, if $x_i$ were last to enter the equation. NEWRAP computes and prints the $t$-statistics, the probability associated with the interval $(-t, t)$, and the additional sums of squares for each term.

This particular test is the basis for the rejection option of NEWRAP. The analyst initially chooses which $\hat{\sigma}^2$ estimator to use by the choice of strategy. Then the analyst may choose a confidence level which all coefficients must meet. For example, suppose a confidence level of 0.900 is chosen. The $t$-statistic is then computed for each coefficient, and the coefficient with minimum $|t|$ is identified. If $\min |t| > t_{N-J-D, 0.950}$, all terms are concluded to be significant at the 0.1 level (or 90.0 percent level of confidence). If $\min |t| < t_{N-J-D, 0.950}$, the term corresponding to the minimum $|t|$ is dropped from the hypothetical model, and the regression is recomputed. This process is repeated until all remaining coefficients are significant at the specified level of probability. This procedure can be overridden by an option which allows certain specified terms of the model to be retained regardless of the significance of the coefficient. Ken-
nedy and Bancroft (ref. 8) present a study indicating the backward deletion method is slightly more efficient than forward selection in special situations.

**PREDICTING VALUES FROM ESTIMATED REGRESSION EQUATION**

Regression equations are often used to predict an estimated response at some condition of the independent variables. Useful estimates of parameters to know are the variance of the regression equation and the variance of a single further observation at the desired combination of the independent variables.

Let $x^t = (x_1, \ldots, x_J)$ denote the vector of independent variables at which a prediction is desired. Let $x^* = x - \bar{x}$. Let $\hat{\sigma}^2_{\mu \cdot x}$ denote the estimated variance of the regression equation at $x$. Let $\hat{\sigma}^2_{y \cdot x}$ denote the estimated variance of a single further observation at $x$. Then,

$$\hat{\sigma}^2_{\mu \cdot x} = \hat{\sigma}^2 \left[ \frac{D}{N} + x^* (X'X)^{-1} x^* \right]$$

(23)

$$\hat{\sigma}^2_{y \cdot x} = \hat{\sigma}^2 \left[ 1.0 + \frac{D}{N} + x^* (X'X)^{-1} x^* \right]$$

(24)

where, as before, $D = 1$ if a $b_0$ coefficient is estimated and $D = 0$ if a $b_0$ coefficient is not estimated. The quantity $s = \hat{\sigma}_{\text{RES}(J)}$ is called the standard error of estimate and often is used as a simple approximation to $\hat{\sigma}_{y \cdot x}$. This approximation is close if $N$ is very large and $x = \bar{x}$, in which case,

$$\hat{\sigma}^2_{y \cdot \bar{x}} = s^2 \left( 1.0 + \frac{D}{N} \right) \approx s^2$$

When $x \neq \bar{x}$, this may be a poor approximation. NEWRAP accepts input vectors $x$ and computes $\hat{y} = \hat{b}_0 + \hat{b}_1 x_1 + \ldots + \hat{b}_J x_J$, as well as $\hat{\sigma}^2_{\mu \cdot x}$, $\hat{\sigma}_{\mu \cdot x}$, $\hat{\sigma}^2_{y \cdot x}$, $\hat{\sigma}_{y \cdot x}$ and the standard error of estimate.
NEWRAP PROGRAM

USERS GUIDE TO NEWRAP INPUT

Illustrative Regression Problem Requiring No Transformations

The illustrative example is described in chapter 7 of reference 5. The data is reproduced in table I. Figure 1 presents this data in a sample input form. The basic model to be fitted is

\[ y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 \]  

(25)

TABLE I. - OBSERVED VALUES
FOR EXAMPLE PROBLEM

<table>
<thead>
<tr>
<th>Unit number</th>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( x_3 )</th>
<th>( x_4 )</th>
<th>( y )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-75</td>
<td>0</td>
<td>0</td>
<td>-65</td>
<td>1.4</td>
</tr>
<tr>
<td>2</td>
<td>175</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>26.3</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>-65</td>
<td>...</td>
<td>29.4</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>165</td>
<td>-6</td>
<td>9.7</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>150</td>
<td>32.9</td>
</tr>
<tr>
<td>10</td>
<td>-75</td>
<td>-75</td>
<td>0</td>
<td>150</td>
<td>26.4</td>
</tr>
<tr>
<td>11</td>
<td>175</td>
<td>175</td>
<td>0</td>
<td>-65</td>
<td>8.4</td>
</tr>
<tr>
<td>14</td>
<td>-75</td>
<td>-75</td>
<td>-65</td>
<td>150</td>
<td>28.4</td>
</tr>
<tr>
<td>15</td>
<td>175</td>
<td>175</td>
<td>165</td>
<td>-65</td>
<td>11.5</td>
</tr>
<tr>
<td>18</td>
<td>0</td>
<td>0</td>
<td>-65</td>
<td>-65</td>
<td>1.3</td>
</tr>
<tr>
<td>19</td>
<td>0</td>
<td>0</td>
<td>165</td>
<td>150</td>
<td>21.4</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>-75</td>
<td>-65</td>
<td>-65</td>
<td>.4</td>
</tr>
<tr>
<td>21</td>
<td>0</td>
<td>175</td>
<td>165</td>
<td>150</td>
<td>22.9</td>
</tr>
<tr>
<td>24</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-65</td>
<td>3.7</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>-75</td>
<td>0</td>
<td>150</td>
<td>26.5</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>-75</td>
<td>0</td>
<td>150</td>
<td>23.4</td>
</tr>
<tr>
<td>16</td>
<td>0</td>
<td>-75</td>
<td>0</td>
<td>150</td>
<td>26.5</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>175</td>
<td>0</td>
<td>-65</td>
<td>5.8</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>175</td>
<td>0</td>
<td>-65</td>
<td>7.4</td>
</tr>
<tr>
<td>17</td>
<td>0</td>
<td>175</td>
<td>0</td>
<td>-65</td>
<td>5.8</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>-75</td>
<td>-65</td>
<td>150</td>
<td>28.8</td>
</tr>
<tr>
<td>22</td>
<td>0</td>
<td>-75</td>
<td>-65</td>
<td>150</td>
<td>26.4</td>
</tr>
<tr>
<td>13</td>
<td>0</td>
<td>175</td>
<td>165</td>
<td>-65</td>
<td>11.8</td>
</tr>
<tr>
<td>23</td>
<td>0</td>
<td>175</td>
<td>165</td>
<td>-65</td>
<td>11.4</td>
</tr>
</tbody>
</table>
15 SAMPLE NEWRAP PROBLEM 2
DATA IS FROM DRAPER AND SMITH APPLIED REGRESSION, ANALYSIS (REFERENCE 4 OF NEWRAP REPORT)
CHAPTER 7
INITIAL MODEL EQUATION IS

\[ Y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 + \text{ERR} \]

- \( Y \): CHAMBER PRESSURE
- \( x_1 \): TEMPERATURE OF CYCLE
- \( x_2 \): VIBRATION LEVEL
- \( x_3 \): DROP (SHOCK)
- \( x_4 \): STATIC PRESSURE

RESIDUALS ARE BEING REQUESTED TO BE PUNCH (FOR COMPIL PROGRAM).
FOR RESIDUAL PLOTTING, ANALYSIS.

\[ \text{UNIT NO. } \begin{array}{cccccc}
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array} \]

\[ \begin{array}{cccccc}
0.1 & -7.5 & 0 & 0 & -65 & 3.4 \\
0.2 & -1.75 & 0 & 1.5 & 10 & 4.6 \\
0.7 & 0 & 0.65 & 0 & 150 & 2.4 \\
8 & 0 & 0 & 1.65 & -65 & 9.7 \\
\end{array} \]

**SAMPLE INPUT (Concluded)**

**FORTRAN STATEMENT**

Figure 1 - Sample input form.
The preceding model requires no transformations of the tabulated data for the dependent or independent variables. Suitable input statements are also given in figure 1.

A subsequent example will illustrate the requirements on the input cards when transformations are involved.

**Detailed Description of Input Cards**

This section of the report describes the input cards as classified into nine sets according to table II.

<table>
<thead>
<tr>
<th>Set number</th>
<th>Name of set</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IDENTIFICATION</td>
<td>Identify and describe problem</td>
</tr>
<tr>
<td>2</td>
<td>PROBLEM SIZE</td>
<td>Define problem size</td>
</tr>
<tr>
<td>3</td>
<td>LOGIC</td>
<td>Specify general logical controls</td>
</tr>
<tr>
<td>4</td>
<td>MODEL</td>
<td>Define model equation</td>
</tr>
<tr>
<td></td>
<td>(a) MODEL SIZE</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(b) TERMS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(c) TRANSFORMATIONS</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(d) CONSTANTS</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>REJECTION</td>
<td>Backward rejection controls</td>
</tr>
<tr>
<td>6</td>
<td>REPLICAetes</td>
<td>Identify replicated data</td>
</tr>
<tr>
<td>7</td>
<td>FORMAT</td>
<td>Give data format</td>
</tr>
<tr>
<td>8</td>
<td>DATA</td>
<td>Input observed data</td>
</tr>
<tr>
<td>9</td>
<td>PREDICTIONS</td>
<td>Predicted values data</td>
</tr>
</tbody>
</table>

The model equation is defined by set 4 of table II. An example illustrating the use of one blank card for input set 4 which can be used for simple linear regressions is presented by figure 1 and table I. A second example illustrating the use of the set of MODEL cards in the presence of prior constants and transformations will be given at the end of this section of the report. A pictorial representation of an input deck is given by figure 2.
Figure 2. - Sample input deck. (Asterisk denotes the card is optional and its use depends upon data input on previous cards.)
Figure 2. - Concluded.
Details of the input cards are as follows:

(1) **IDENTIFICATION** (I, IDENT)(II, 13A6): IDENT is Hollerith data used to identify the problem. I indicates the number of additional cards to read for further identification or description (columns 1 to 78).

(2) **PROBLEM SIZE** (NOVAR, NODEP, NOTERM, NOOB, NTKEEP)(3I4, I5, I4)

- **NOVAR** Number of input independent variables (number of z's in eq. (2))
- **NODEP** Number of input dependent variables
- **NOTERM** Number of terms in model equation (number of x's in eq. (3)). Note that b_0 is not counted as a term.
- **NOOB** Number of observations
- **NTKEEP** First NTKEEP independent terms of model equation will be retained in model regardless of significance level

(3) **LOGIC:** One card with nine one-column fields

- **BZERO** b_0 term appears in model equation (T or F)
- **IFTT** t-statistics and their descriptive confidence levels are to be computed (T or F)
- **IFWT** Weight of 1.0 is applied to all observations (T or F). If this is F, see input sets 7 and 8 for further information.
- **IFCHI** Compute and plot residuals (T or F)
- **STORYX** Calculate eigenvalues and eigenvectors of X'X (T or F)
- **IFSSR** Model shall be increased by one term at a time using bordering method for matrix inversion (T or F)
- **ECONMY** Use economy version of output (T or F). NEWRAP does not print X'y, (X'X)^{-1}, or C when set to F.
- **ISTRAT** Pooling strategy is 1, 2, or 3:
  1. Never pool. Use replication error as estimate of error. If 1 is selected and no replication is found, strategy 3 is used.
  2. Pool initial residual only.
  3. Pool all residuals.
Punch residuals and predicted values (T or F). If T, observation number is punched and then residuals and predicted values are punched in \((16, 4E16.8/(6X, 4E16.8))\) format in pairs (observation number, \(e_1, \hat{y}_1, e_2, \hat{y}_2\), etc.).

(4) MODEL: The MODEL cards are used to manipulate the observed input data, supplied by input set 8, into the form of the desired model equation. There are four subsets of this input set 4, namely, MODEL SIZE, TERMS, TRANSFORMATIONS, and CONSTANTS, of which the latter three are used only in the development of complex models.

If a simple linear model is being analyzed, the MODEL SIZE card is left blank, indicating that the number of transformations is zero and the number of constants to be read in is zero. In this case, the TERMS, TRANSFORMATIONS, and CONSTANTS cards of this input set are not expected by the program, and the program assumes the independent and dependent variables are arranged on the input cards of input set 8 as

\[
x_1, x_2, \ldots, x_j, y_1, \ldots, y_{\text{NODEP}}
\]

where NODEP is the number of dependent variables.

If a weighting factor other than 1.0 is to be used (i.e., if item 3 of the LOGIC card contains an F), the value of the weighting factor for each observation must appear as the last item in the list, so that in this case the data for each observation is entered on the cards as

\[
x_1, x_2, \ldots, x_j, y_1, \ldots, y_{\text{NODEP}}, WT
\]

If the weighting factor is identically 1.0, NEWRAP reads a total of \(M\) numerical values for each observation, where \(M\) is the sum of the number of independent and dependent variables. The variables are stored consecutively in an array called \(X\), beginning with location 01 and ending with location \(M\). If the weighting factor is not identically 1.0, then \(M + 1\) numerical values are read for each observation, but the last value, being the weighting factor, is treated and stored separately. The data in \(X\) are used with their appropriate weighting factors to cumulatively create \(X'X\) and \(X'y\) as shown in equations (12a).

The remaining discussion of this set explains the use of transformations and/or constants to build more complex models. Therefore, the reader who does not immediately need a complex model may skip this material and proceed directly to the description of input set 5.

As mentioned previously, there are up to four subsets of the MODEL cards. Their purpose is to give the structure of the model equation and thereby specify the initial
operations to be performed on the input data. As used here, CONSTANTS means any numerical value specified to be in the model equation in advance of parameter estimation. These numerical values are read from the CONSTANTS cards.

Also, the word TRANSFORMATIONS is to be interpreted as the operations performed on the input data (read from data cards) to compute the \( f_j \) values (eq. (2)) of the model equation. The structure of these functions (and of any transformations of the dependent variables) is read from the TRANSFORMATION cards. Finally, the word TERMS is to be interpreted as the computed results of the operations specified by the transformation (including any operations that leave the input data unchanged). The results of the TRANSFORMATIONS are stored in an array CON, and the TERMS cards designate the order of the relative locations in CON where the final values for the terms of the model equations are to be found.

The four subsets, MODEL SIZE, TERMS, TRANSFORMATIONS, and CONSTANTS, will be described in detail now. Also, at the end of the description of this input set, a summary of these cards, with the formats used, is given for convenience.

The MODEL SIZE card specifies NTRANS and KONNO(214) where

\[
\text{NTRANS} \quad \text{Number of transformations that will be performed}
\]

\[
\text{KONNO} \quad \text{Number of constants that will be read in which are required to specify model equation}
\]

If the number of transformations is zero, and therefore, the number of constants is zero, the TERMS, TRANSFORMATIONS, and CONSTANTS cards are not expected by the program. This being a simple linear model case, only the MODEL SIZE card, which can be blank, is necessary in this subset, but the values for the observations which are provided in input set 8 must conform to the order as specified in the first three paragraphs describing this input set.

When, however, a more complex model is desired, information must be supplied instructing the program as to (1) where to find the values for the TERMS of the equation, (2) how to create the terms from the variables and the constants, and (3) what the values of the constants are. This information is supplied on the TERMS, TRANSFORMATIONS, and CONSTANTS cards.

The numerical values to be used in the transformations are stored in two arrays called X and CON. The transformations always require that an operator (some value from CON) performs an operation (see table III) on an operand (some specified value from X) to produce a result which will be stored in CON. Thus CON serves two purposes. First, if the number of constants (KONNO) specified on the MODEL SIZE card is nonzero, that many constants will be read from the CONSTANTS cards and stored in CON beginning with location 01. If the number of constants is zero, a CONSTANTS card is not expected by the program. Second, all intermediate and final results of
TABLE III. - OPERATIONS AND CODE NUMBERS

[X indicates a value from X and C a value from CON.]

<table>
<thead>
<tr>
<th>Operation code (OP)</th>
<th>Resulting operation</th>
<th>Operation code (OP)</th>
<th>Resulting operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>No operation</td>
<td>16</td>
<td>1.0. SQRT(X)</td>
</tr>
<tr>
<td>01</td>
<td>N + C</td>
<td>17</td>
<td>C**X</td>
</tr>
<tr>
<td>02</td>
<td>X*C</td>
<td>18</td>
<td>10.0**X</td>
</tr>
<tr>
<td>03</td>
<td>C/X</td>
<td>19</td>
<td>SINH(X)</td>
</tr>
<tr>
<td>04</td>
<td>EXP(X)</td>
<td>20</td>
<td>COSH(X)</td>
</tr>
<tr>
<td>05</td>
<td>X**C</td>
<td>21</td>
<td>(1.0-COS(X)).2.0</td>
</tr>
<tr>
<td>06</td>
<td>ALOG(X)</td>
<td>22</td>
<td>ATAN(X)</td>
</tr>
<tr>
<td>07</td>
<td>ALOG10(X)</td>
<td>23</td>
<td>ATAN2(X, C)</td>
</tr>
<tr>
<td>08</td>
<td>SIN(X)</td>
<td>24</td>
<td>X**2</td>
</tr>
<tr>
<td>09</td>
<td>COS(X)</td>
<td>25</td>
<td>X**3</td>
</tr>
<tr>
<td>10</td>
<td>SIN(π<em>C</em>X)</td>
<td>26</td>
<td>ARCSIN(SQRT(X))</td>
</tr>
<tr>
<td>11</td>
<td>COS(π<em>C</em>X)</td>
<td>27</td>
<td>2.0<em>π</em>X</td>
</tr>
<tr>
<td>12</td>
<td>1.0/X</td>
<td>28</td>
<td>1.0.(2.0.<em>π</em>X)</td>
</tr>
<tr>
<td>13</td>
<td>EXP(C./X)</td>
<td>29</td>
<td>ERF(X)</td>
</tr>
<tr>
<td>14</td>
<td>EXP(C;X**2)</td>
<td>30</td>
<td>GAMMA(X)</td>
</tr>
<tr>
<td>15</td>
<td>SQRT(X)</td>
<td>31</td>
<td>X C</td>
</tr>
</tbody>
</table>

aAll function names and operations are consistent with FORTRAN IV mathematical subroutines.

Transformations are also stored in CON as specified on the TRANSFORMATION cards. The TERMS card then specifies which of the locations in CON finally contain the values needed to construct the X'X and X'y matrices. After all the transformations have been performed on an observation, the contents of the relative locations of the CON array specified on the TERMS card are moved back to X in consecutive locations beginning with location 01.

Note especially that CONSTANTS data are stored in CON from location 01 through KONNO. Thus, if a transformation specifies that a result is to be placed in any of these locations, the result will replace the constant, so that further operations on subsequent transformations would use the new value stored instead of the constant value to which it was initialized. Care should be taken, therefore, that the results of the transformations be stored in relative locations greater than KONNO.

Each transformation code is made up of four subfields of two card columns each, with the following interpretation:
Thus, subfield 1 always references the X array, and subfields 3 and 4 reference the CON array. The result of every transformation is a term which is stored in the designated location of the CON array, with the added feature that, if the term is stored in relative location 61 or beyond, it is also stored in the parallel location in the X array. This is illustrated by the arrows in figure 3. This feature allows successive transformations to be performed more easily.

The OP (operation codes) are tabulated in table III. The transformation with \( \text{OP} = 00 \) is simply an identity transformation. This transfers data from X to CON so that when terms are selected there is a value available in CON that can be moved back to X.

When there are no transformations, NEWRAP assumes the first NOTERM values on a data card are the independent variables and the last NODEP values are the depend-

<table>
<thead>
<tr>
<th>Subfield</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Operand</td>
</tr>
<tr>
<td>2</td>
<td>Operation (OP)</td>
</tr>
<tr>
<td>3</td>
<td>Operator</td>
</tr>
<tr>
<td>4</td>
<td>Result</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location</th>
<th>X</th>
<th>Location</th>
<th>CON</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>Input array</td>
<td>01</td>
<td>Input constants</td>
</tr>
<tr>
<td>02</td>
<td></td>
<td>02</td>
<td></td>
</tr>
<tr>
<td>03</td>
<td></td>
<td>03</td>
<td></td>
</tr>
<tr>
<td>04</td>
<td></td>
<td>04</td>
<td></td>
</tr>
<tr>
<td>05</td>
<td></td>
<td>05</td>
<td></td>
</tr>
<tr>
<td>06</td>
<td></td>
<td>06</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>56</td>
<td></td>
<td>56</td>
<td></td>
</tr>
<tr>
<td>57</td>
<td></td>
<td>57</td>
<td></td>
</tr>
<tr>
<td>58</td>
<td></td>
<td>58</td>
<td></td>
</tr>
<tr>
<td>59</td>
<td></td>
<td>59</td>
<td></td>
</tr>
<tr>
<td>60</td>
<td></td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>61</td>
<td></td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>62</td>
<td></td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>63</td>
<td></td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td></td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>65</td>
<td></td>
<td>65</td>
<td></td>
</tr>
<tr>
<td>66</td>
<td></td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>67</td>
<td></td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>95</td>
<td></td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>96</td>
<td></td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>97</td>
<td></td>
<td>97</td>
<td></td>
</tr>
<tr>
<td>98</td>
<td></td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>99</td>
<td></td>
<td>99</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3. Map of X and CON arrays. Data transferred into any location of CON array beyond location 60 are immediately duplicated in same relative location in X array.
ent variables. When transformations are used, this convention need not hold for the raw input data but instead holds for the terms on the TERMS card. Thus, the first NOTERM values input on the TERMS card indicate the locations of the CON array which correspond to the independent variables and the last NODEP values indicate the locations of the CON array which correspond to the dependent variables. If the analyst desires to force certain terms to remain in the model regardless of their significance, these terms must be the first terms of the model. Then if the input for NTKEEP of the PROBLEM SIZE card is not zero, the first NTKEEP terms of the model will be retained.

The complete sequence of MODEL cards and the formats used are summarized as follows:

(a) **MODEL SIZE (NTRANS, KONNO)(214):** NTRANS specifies the number of transformations required and KONNO the number of CONSTANTS involved. NTRANS may not be greater than 100 and KONNO not greater than 60. If NTRANS = 0, the following three sets are skipped and the program goes directly to input set 5.

(b) **TERMS (4012):** One or more cards as necessary, using two-column fields to denote the relative locations of the CON array containing the final values for the terms to be used and the order in which they enter into the model equation. The number of terms used is specified on the PROBLEM SIZE card.

(c) **TRANSFORMATIONS (4012):** As many cards as necessary containing up to 10 transformation instructions per card. Each transformation instruction is composed of four two-column fields. See table III for the list of available transformations.

(d) **CONSTANTS (5E15.7):** As many cards as necessary containing the number of CONSTANTS as specified by KONNO. Up to 60 CONSTANTS may be specified. If KONNO = 0, these cards are not expected by the program.

(5) **REJECTION (DELETE, P)(L1, F3.3):** If DELETE is set to T, the backward rejection option is used and the desired level of confidence is given by P. The P value is written without a decimal point so that a 95 percent confidence level is indicated by a 950, a 99.9 percent level as 999, and so forth.

(6) **REPLICATION (REPS)(L1):** If REPS is F, the program skips to set 7 and assumes there is no replicated data. If REPS is set to T, then more cards are read in 2014 format specifying:

IREP in the first field of the first card indicates the number of replicate sets.

NARAY in the second field of the first card and the remaining fields of this and succeeding cards consists of an array containing the number of observations in each of the replicate sets.

Note that it is not safe for the program to assume that all the data points for an experiment with the same levels of the independent variables are true replicates. Thus the user must explicitly specify the truly replicated sets. NEWRAP does check that all
independent terms within a replicate set are the same. If not, the program stops. A nonreplicated data point is considered to be a group of size 1. Note that the data in table I are grouped to clearly indicate the replicated data points.

(7) FORMAT (INPUT, FMT)(I2, 13A6): INPUT specifies the unit number on which the input data is stored; and FMT supplies the format for reading it.

Note that, if a weighting factor other than 1.0 is to be used, its value will be read with each data point, and the format must allow for this.

The example from Draper and Smith (ref. 5) uses a weighting value of 1.0 for all data. The format is (12X, 5F6.0) since there are four independent and one dependent variable to be read. If a weighting value other than 1.0 is used, it must appear with every data point as the last value on the card. In such a case, the format could, for example, be (12X, 5F6.0, F10.3).

(8) DATA: Each observation consisting of the given z's and y's read by the execution of one READ statement. Thus, there will be at least one card for each observation. As mentioned previously, if the transformation option is not used, the program expects the first variables to be the independent variables, in the order in which they enter the model, followed by the dependent variables and then the weighting value if IFWT = .F. Otherwise, if transformations are used, the independent and dependent variables may be entered in any convenient order, because the TERMS card(s) will be needed to specify the order in which the values will enter the model equation. However, if IFWT = .F., the weighting value is still the last value supplied with each observation.

(9) PREDICTIONS (PREDCT)(L1):

If, with the program LOGIC (input set 3) card, a computation of residuals is requested by a T in card column 4, then predicted values of the dependent variables are computed for all the input values of the independent variables. In addition to these predicted values, predictions at other values of the independent variables might be desired. In PREDICTIONS input, one card with one column is used to indicate if these other predictions are desired (T or F). If this is F, a new case is started and the new case should start with input set 1 cards. If it is T, the following cards are read: One card with one four-column field specifying the number of predictions desired. This is followed by cards with the values of the independent variables at which predictions are desired. Only the final regression model is used, but the number of independent and dependent variables originally supplied on the PROBLEM SIZE data cards are read. All transformations indicated on the MODEL cards are performed. Then the proper terms are chosen by the program to correspond to the final model. Since the dependent variables are not needed in this part of the program, the numerical values for the dependent variables are dummy values and should be in the appropriate range so that when subroutines required for the transformations (e.g., ALOG, SQRT) use these values, abnormal exits will not occur.
Illustrative Problem Requiring Transformations

As an example of the MODEL cards usage consider the following. Suppose the model we are required to construct is

\[ \log_{10}(y + 273.15) = b_0 + b_1z_1 + b_2z_2 + b_3z_3 + b_4z_1z_2 + b_5z_1z_3 + b_6z_2z_3 + b_7z_1^2 + b_8z_2^2 + b_9z_3^2 \]

Thus, in terms of equation (2) we have

\[
\begin{align*}
  x_1 &= z_1 \\
  x_2 &= z_2 \\
  x_3 &= z_3 \\
  x_4 &= z_1z_2 \\
  x_5 &= z_1z_3 \\
  x_6 &= z_2z_3 \\
  x_7 &= z_1^2 \\
  x_8 &= z_2^2 \\
  x_9 &= z_3^2
\end{align*}
\]

\[ y = \log_{10}(y + 273.15) \]

Table IV shows a sequence of transformations which could be used to construct this model equation. Figure 4 shows how the MODEL cards describing this equation would appear on a FORTRAN data sheet. Figure 5 shows the X and CON array contents both before and after the transformations are performed upon one observation and the X array after the appropriate terms have been selected according to the TERMS card data.
### TABLE IV. - SEQUENCY OF TRANSFORMATIONS FOR EXAMPLE

<table>
<thead>
<tr>
<th>Transformation number</th>
<th>Operand</th>
<th>Operation</th>
<th>Operator</th>
<th>Result</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>11</td>
<td>$x_1 - \text{CON}(11)$</td>
</tr>
<tr>
<td>2</td>
<td>01</td>
<td>00</td>
<td>00</td>
<td>61</td>
<td>$x_1 - X(61), \text{CON}(61)$</td>
</tr>
<tr>
<td>3</td>
<td>02</td>
<td>00</td>
<td>00</td>
<td>12</td>
<td>$x_2 - \text{CON}(12)$</td>
</tr>
<tr>
<td>4</td>
<td>02</td>
<td>00</td>
<td>00</td>
<td>62</td>
<td>$x_2 - X(62), \text{CON}(62)$</td>
</tr>
<tr>
<td>5</td>
<td>03</td>
<td>00</td>
<td>00</td>
<td>13</td>
<td>$x_3 - \text{CON}(13)$</td>
</tr>
<tr>
<td>6</td>
<td>03</td>
<td>00</td>
<td>00</td>
<td>63</td>
<td>$x_3 - X(63), \text{CON}(63)$</td>
</tr>
<tr>
<td>7</td>
<td>61</td>
<td>02</td>
<td>00</td>
<td>17</td>
<td>$x_1^2 - \text{CON}(17)$</td>
</tr>
<tr>
<td>8</td>
<td>62</td>
<td>02</td>
<td>00</td>
<td>18</td>
<td>$x_2^2 - \text{CON}(18)$</td>
</tr>
<tr>
<td>9</td>
<td>63</td>
<td>02</td>
<td>00</td>
<td>19</td>
<td>$x_3^2 - \text{CON}(19)$</td>
</tr>
<tr>
<td>10</td>
<td>61</td>
<td>02</td>
<td>00</td>
<td>20</td>
<td>$x_1x_2 - \text{CON}(14)$</td>
</tr>
<tr>
<td>11</td>
<td>61</td>
<td>02</td>
<td>00</td>
<td>15</td>
<td>$x_1x_3 - \text{CON}(15)$</td>
</tr>
<tr>
<td>12</td>
<td>62</td>
<td>02</td>
<td>00</td>
<td>16</td>
<td>$x_2x_3 - \text{CON}(16)$</td>
</tr>
<tr>
<td>13</td>
<td>04</td>
<td>01</td>
<td>00</td>
<td>98</td>
<td>$y + 273.15 - X(98), \text{CON}(98)$</td>
</tr>
<tr>
<td>14</td>
<td>98</td>
<td>07</td>
<td>00</td>
<td>20</td>
<td>$\log_{10}(y + 273.15) - \text{CON}(20)$</td>
</tr>
</tbody>
</table>

---

**Figure 4.** - An example of MODEL cards.
Figure 5. - Arrays $X$ and $CON$ before and after transformations and terms selection for the example.
SAMPLE MCREP PROBLEM
DATA IS FROM GRAPHER AND SMITH
APPLIED REGRESSION ANALYSIS (REFERENCE 4 OF MCREP REPORT)
CHAPTER 7
INITIAL MODEL EQUATION IS
Y= CHAMBER PRESSURE
X1= TEMPERATURE OF CYCLE
X2= VIBRATION LEVEL
X3= DROPSHOCK
Y= SB + X1X2 + X3X2 + E
RESIDUALS ARE BEING REQUESTED TO BE PUNCHED (FOR CARSPT PROGRAM)
FOR RESIDUAL PLOTTING ANALYSES
***********************************************************************
1 2 24
TTLITIPIT
THERE IS A BD TO ESTIMATE
THERE ARE 10 REPPLICATE SETS
1 1 1 1 1 1 1 1 1 1 3 3 2 2

123,567.01
SAMPLE MCREP PROBLEM
TECH OF THE EQUATION, OBSERVATION =
-75.00000 1.400000
** REPPLICATE SET
1
TERMS OF THE EQUATION, OBSERVATION =
175.00000 150.00000 26.300000
** REPPLICATE SET
2
TERMS OF THE EQUATION, OBSERVATION =
0 0 -65.00000 150.00000 29.400000
** REPPLICATE SET
3
TERMS OF THE EQUATION, OBSERVATION =
0 0 165.00000 -65.00000 9.700000
** REPPLICATE SET
4
TERMS OF THE EQUATION, OBSERVATION =
0 0 150.00000 32.900000
** REPPLICATE SET
5
TERMS OF THE EQUATION, OBSERVATION =
-75.00000 -75.00000 150.00000 26.400000
** REPPLICATE SET
6
TERMS OF THE EQUATION, OBSERVATION =
175.00000 175.00000 0 -65.00000 8.400000
** REPPLICATE SET
7
TERMS OF THE EQUATION, OBSERVATION =
-75.00000 -75.00000 -65.00000 28.400000
** REPPLICATE SET
8
TERMS OF THE EQUATION, OBSERVATION =
175.00000 175.00000 185.00000 21.400000
** REPPLICATE SET
9
TERMS OF THE EQUATION, OBSERVATION =
0 -65.00000 -65.00000 1.300000
** REPPLICATE SET
10

34
**REPLICATE SET 16**  
**DEP. VAR.** 1  
\[ SSQ = 6.006836 \]  
\[ SUM = 76.400000 \]  
\[ MEAN = 25.466666 \]

**REPLICATE SET 17**  
**DEP. VAR.** 1  
\[ SSQ = 0.8000137 \times 10^{-01} \]  
\[ SUM = 23.200000 \]  
\[ MEAN = 11.600000 \]

**MEANS OF INJEP AND DEP VARIABLES**  
12.50000  
33.33333  
25.00000  
42.500000  
16.579167

\[ X \] \[ \text{TRANSPOSE} \] \[ X \] \[ \text{MATRIX} \]  
\[ \text{WHERE} \] \[ X \] \[ \text{IS} \] \[ \text{DEVIATION} \] \[ \text{FROM} \] \[ \text{MEAN} \]

\begin{array}{c}
\text{ROW 1} & 105000.0 \\
\text{ROW 2} & 62500.00 \\
\text{ROW 3} & 26250.00 \\
\text{ROW 4} & -26875.00 \\
\end{array}

\begin{array}{c}
\text{ROW 1} & 1.000000 \\
\text{ROW 2} & 0.375865 \\
\text{ROW 3} & 0.194372 \\
\text{ROW 4} & -0.157485 \\
\end{array}

\[ X \] \[ \text{TRANSPOSE} \] \[ Y \] \[ \text{MATRIX} \]  
\[ \text{WHERE} \] \[ X \] \[ \text{AND} \] \[ Y \] \[ \text{ARE} \] \[ \text{DEVIATIONS} \] \[ \text{FROM} \] \[ \text{MEAN} \]

\begin{array}{c}
\text{ROW 1} & -1103.750 \\
\text{ROW 2} & -12393.33 \\
\text{ROW 3} & -2767.500 \\
\text{ROW 4} & 25875.25 \\
\end{array}

\[ \text{CORRELATION COEFFICIENTS} \]

\begin{array}{c}
\text{ROW 1} & 1.000000 \\
\text{ROW 2} & 0.357665 \\
\text{ROW 3} & 0.194372 \\
\text{ROW 4} & -0.157485 \\
\end{array}

\[ \text{THE FOLLOWING ARE EIGENVALUES OF} \] \[ X \] \[ \text{TRANSPOSE} \] \[ X \] \[ \text{MATRIX} \]

\begin{array}{c}
4890.3932 \\
1090.3844 \\
9938.353 \\
61947.348 \\
\end{array}

\[ \text{THIS IS THE MODAL} \] \[ \text{MATRIX} \] \[ \text{OF EIGENVECTORS. EIGENVECTORS ARE} \] \[ \text{WRITTEN} \] \[ \text{IN} \] \[ \text{COLUMNS} \] \[ \text{LEFT} \] \[ \text{TO} \] \[ \text{RIGHT} \] \[ \text{IN} \] \[ \text{SAME} \] \[ \text{ORDER AS} \] \[ \text{EIGENVALUES} \]

\begin{array}{c}
\text{ROW 1} & 0.117713 \\
\text{ROW 2} & 0.674835 \\
\text{ROW 3} & 0.358164 \\
\text{ROW 4} & -0.060269 \\
\end{array}

\begin{array}{c}
\text{ROW 1} & 0.231943 \\
\text{ROW 2} & 0.224345 \\
\text{ROW 3} & 0.194372 \\
\text{ROW 4} & 0.231943 \\
\end{array}

\begin{array}{c}
\text{ROW 1} & 0.820665 \\
\text{ROW 2} & 0.211339 \\
\text{ROW 3} & 0.300994 \\
\text{ROW 4} & 0.393240 \\
\end{array}

\begin{array}{c}
\text{ROW 1} & 0.682691 \\
\text{ROW 2} & 0.171590 \\
\text{ROW 3} & -0.343240 \\
\text{ROW 4} & 0.900091 \\
\end{array}

\begin{array}{c}
\text{ROW 1} & 0.504097 \\
\text{ROW 2} & -0.225295 \\
\text{ROW 3} & 0.420793 \\
\text{ROW 4} & 0.282691 \\
\end{array}
SAMPLE DESIGN PROBLEM
EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDENT TERM

CONSTANT TERM (BX)
1 1 1 1 1
REGRESSION COEFFICIENTS (B1,...,BK)
1 0.731127E-02
2 0.111125E-01
3 0.424556E-02
4 0.101374

ANOVA OF REGRESSION ON DEPENDENT VARIABLE Y

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>SUMS OF SQUARES</th>
<th>DEGREES OF FREEDOM</th>
<th>MEAN SQUARES</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGRESSION</td>
<td>2462.36246</td>
<td>4</td>
<td>615.64739</td>
</tr>
<tr>
<td>RESIDUAL</td>
<td>244.03688</td>
<td>19</td>
<td>13.10740</td>
</tr>
<tr>
<td>TOTAL</td>
<td>2711.59978</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

R SQUARED = 35566/35566 = 0.9980299
STANDARD ERROR OF ESTIMATE = 3.620E-05

USING DUMMY STRATEGY 1 THE REGRESSION MEAN SQUARE = 1.8455954 WITH DEGREES OF FREEDOM = 6
F = 333.58 COMPARABLE TO FIT 4.

ANOVA OF LACK OF FIT

<table>
<thead>
<tr>
<th>SOURCE</th>
<th>SUMS OF SQUARES</th>
<th>DEGREES OF FREEDOM</th>
<th>MEAN SQUARES</th>
</tr>
</thead>
<tbody>
<tr>
<td>LACK OF FIT</td>
<td>237.46724</td>
<td>13</td>
<td>18.304649</td>
</tr>
<tr>
<td>RESIDUAL</td>
<td>244.03688</td>
<td>19</td>
<td>13.10740</td>
</tr>
<tr>
<td>TOTAL</td>
<td>2711.59978</td>
<td>23</td>
<td></td>
</tr>
</tbody>
</table>

F = 6.03
SAMPLE WITH RESIDUES

SUMS OF SQUARES DUE TO EACH VARIABLE IF IT WERE LAST TO ENTER REGRESSION
1 5.20792
2 14.27918
3 2.21954
4 1783.654

STANDARD DEVIATION OF REGRESSION COEFFICIENTS (DERIVED FROM DIAGONAL ELEMENTS OF (X TRANSPOSE X) INVERSE MATRIX)
0 0.354598
1 0.496375E-02
2 0.420452E-02
3 0.391621E-02
4 0.324088E-02

(X TRANSPOSE X) INVERSE MATRIX

| HOW | 1 | 0.111626E-04 |
| BOW | 2 | -0.360249E-05 |
| ROW | 3 | 0.220169E-05 |
| ROW | 4 | -0.73908E-06 |

SAMPLE Y-LAP PROBLEM

CALCULATE T STATISTICS
THE T STATISTICS CAN BE USED TO TEST THE REGRESSION COEFFICIENTS B1,...,BK.

T = 3.856
T = 2.892
T = 3.687
T = 2.049

UNDER NULL HYPOTHESIS THE INTERVAL (-T,T) WHERE T IS GIVEN ABOVE, HAS APPROX PROBABILITY LISTED BELOW.
MINUS SIGN INDICATES PROB EXCEEDS .005.
1 0.05
2 0.025
3 0.01
4 0.005

THE DESIRABLE VALUE OF PROBABILITY IS 0.05 PERCENT.
THE TERM AT 33 IS BEING DELETED.

36
SAMPLE: I:45AP 24URLY
EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDENT TERM
CONSTANT TERM (100)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.751722</td>
<td>-0.2</td>
<td>-0.1</td>
</tr>
<tr>
<td>2</td>
<td>0.11102</td>
<td>0.834</td>
<td>0.675</td>
</tr>
</tbody>
</table>

CALCULATE THE STATISTICS
THE T STATISTICS CAN BE USED TO TEST THE REGRESSION COEFFICIENTS

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3.85</td>
<td>3.99</td>
</tr>
<tr>
<td>2</td>
<td>3.85</td>
<td>3.99</td>
</tr>
<tr>
<td>3</td>
<td>3.85</td>
<td>3.99</td>
</tr>
<tr>
<td>4</td>
<td>3.85</td>
<td>3.99</td>
</tr>
</tbody>
</table>

SAMPLE SIZE: 4

UNDER NULL HYPOTHESIS THE INTERVAL [-T, T] WHERE T IS GIVEN ABOVE, HAS APPROXIMATE PROBABILITY LISTED BELOW.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.99</td>
<td>5.99</td>
</tr>
<tr>
<td>2</td>
<td>5.99</td>
<td>5.99</td>
</tr>
<tr>
<td>3</td>
<td>5.99</td>
<td>5.99</td>
</tr>
<tr>
<td>4</td>
<td>5.99</td>
<td>5.99</td>
</tr>
</tbody>
</table>

THE REJECTED VALUE OF PROBABILITY IS .05, .5 PERCENT
THE TERM AT .05 IS BEING CONSIDERED.
SAMPLE A RAP PROBLEM
EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDENT TERM

11.7 691
REGRESSION COEFFICIENTS [B1,...,BK]
2 0.195925E+01
4 0.107936

ANNOA OF REGRESSION ON DEPENDENT VARIABLE:

SOURCE SUMS OF SQUARES DEGREES OF FREEDOM MEAN SQUARES
REGRESSION 2455.28703 2 1227.64381
RESIDUAL 256.311962 21 12.2053314
TOTAL 2711.5988 23

R SQUARED = SS(REG)/SS(TOT) = 0.905676 R = 0.951565
STANDARD ERROR OF ESTIMATE = 3.493613

USING POOLED STRATEGY 1 THE ERROR MEAN SQUARE = 1.8455595 WITH DEGREES OF FREEDOM = 6
F = MS(RES)/MS(R) = 665.19

ANNOA OF LACK OF FIT:

SOURCE SUMS OF SQUARES DEGREES OF FREEDOM MEAN SQUARES
LACK OF FIT 245.238604 15 16.3692401
REPLICATION 11.073571 6 1.84955951
RESIDUAL 256.311962 21 12.2053314

F = MS(LACK)/MS(REPS) = 8.859

SUMS OF SQUARES DUE TO EACH VARIABLE IF IT WERE LAST TO ENTER REGRESSION
2 1.26720
4 1871.545

STANDARD DEVIATION OF REGRESSION COEFFICIENTS (DERIVED FROM DIAGONAL ELEMENTS OF (X TRANSPOSE X) INVERSE MATRIX)
0 0.539465
2 0.3249836E+02
4 0.3241486E+02

(X TRANSPOSE X) INVERSE MATRIX
ROW 1 0.589698E-05
ROW 2 0.324838E+05 0.559879E-05
SAMPLE NEWRAP PROBLEM

CALCULATED T STATISTICS
THE T STATISTICS CAN BE USED TO TEST THE REGRESSION COEFFICIENTS bilateral,
4.428584
3.884463
UNDER NULL HYPOTHESIS THE INTERVAL (-T,T) WHERE T IS GIVEN ABOVE, HAS APPROX PROBABILITY LISTED BELOW.
MINUS SIGN INDICATES T EXCEEDS .95%
2 0.997
4 0.999

THE DESIRED VALUE OF PROBABILITY IS .950 PERCENT
SAMPLE NEWRAP PROBLEM
FOR EACH DEPENDENT TERM AND OBSERVATION IS PRINTED
OBSERVED (Y OBSERVED)
CALCULATED RESPONSE (Y CALC)
RESIDUAL (Y OIF - Y CALC)
STUDENTIZED RESIDUAL (T)

Y OBSERVED 1.0000
Y CALC 1.0000
Y DIF 0.0000
STUDENTIZED 0.0000

Y OBSERVED 26.300
Y CALC 27.063
Y DIF -0.763
STUDENTIZED 0.3619

Y OBSERVED 29.400
Y CALC 27.063
Y DIF 2.3367
STUDENTIZED 1.7200
<table>
<thead>
<tr>
<th>Sample Newrap Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y O'SERVED</td>
</tr>
<tr>
<td>Y CALC</td>
</tr>
<tr>
<td>Y DIF</td>
</tr>
<tr>
<td>STUDENTIZED</td>
</tr>
<tr>
<td>Y OBSERVED</td>
</tr>
<tr>
<td>Y CALC</td>
</tr>
<tr>
<td>Y DIF</td>
</tr>
<tr>
<td>STUDENTIZED</td>
</tr>
</tbody>
</table>

| Y OBSERVED | 26.4000 |
| Y CALC | 25.2933 |
| Y DIF | 0.5166 |
| STUDENTIZED | 0.3729 |

| Y OBSERVED | 8.4000 |
| Y CALC | 7.7650 |
| Y DIF | 0.6150 |
| STUDENTIZED | 0.5704 |

| Y OBSERVED | 28.4000 |
| Y CALC | 25.8933 |
| Y DIF | 2.5066 |
| STUDENTIZED | 1.8451 |

| Y OBSERVED | 11.5000 |
| Y CALC | 7.7650 |
| Y DIF | 3.7150 |
| STUDENTIZED | 2.7344 |

| Y OBSERVED | 1.5000 |
| Y CALC | 5.1550 |
| Y DIF | -3.6550 |
| STUDENTIZED | -2.7641 |

<table>
<thead>
<tr>
<th>Sample Ylwrap Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y OBSERVED</td>
</tr>
<tr>
<td>Y CALC</td>
</tr>
<tr>
<td>Y DIF</td>
</tr>
<tr>
<td>STUDENTIZED</td>
</tr>
</tbody>
</table>

| Y OBSERVED | 0.4000 |
| Y CALC | 3.8951 |
| Y DIF | -3.4951 |
| STUDENTIZED | -2.9854 |

| Y OBSERVED | 22.9000 |
| Y CALC | 29.7933 |
| Y DIF | -6.8932 |
| STUDENTIZED | -5.0794 |

| Y OBSERVED | 3.7000 |
| Y CALC | 5.0550 |
| Y DIF | -1.3550 |
| STUDENTIZED | -0.9974 |

| Y OBSERVED | 26.5000 |
| Y CALC | 25.0933 |
| Y DIF | 1.4066 |
| STUDENTIZED | 1.0466 |

| Y OBSERVED | 23.4000 |
| Y CALC | 25.8933 |
| Y DIF | -2.4934 |
| STUDENTIZED | -1.8394 |

| Y OBSERVED | 26.5000 |
| Y CALC | 25.0933 |
| Y DIF | 1.4066 |
| STUDENTIZED | 1.0466 |

| Y OBSERVED | 5.8000 |
| Y CALC | 7.7050 |
| Y DIF | -1.9050 |
| STUDENTIZED | -1.4611 |

| Y OBSERVED | 7.4000 |
| Y CALC | 7.7050 |
| Y DIF | -0.3050 |
| STUDENTIZED | -0.2034 |

| Y OBSERVED | 5.8000 |
| Y CALC | 7.7050 |
| Y DIF | -1.9050 |
| STUDENTIZED | -1.4611 |

<table>
<thead>
<tr>
<th>Sample Ylwrap Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y OBSERVED</td>
</tr>
<tr>
<td>Y CALC</td>
</tr>
<tr>
<td>Y DIF</td>
</tr>
<tr>
<td>STUDENTIZED</td>
</tr>
</tbody>
</table>
The contents of this section include a flow chart of the program, a listing of the routines used in NEWRAP and their major functions, the call structure of the program, a dictionary of the program, and the listing.

**General Mathematical and Logical Flow of Program**

The flow of operation in NEWRAP is illustrated in figure 6.
Figure 6. - Flow chart for NEWRAP.
Routines and Their Major Functions

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Function of routine</th>
</tr>
</thead>
<tbody>
<tr>
<td>BORD</td>
<td>Inverts symmetric matrix of order ( n ) by adding bordering column to already inverted matrix of order ( n - 1 )</td>
</tr>
<tr>
<td>EIGEN</td>
<td>Computes eigenvalues and eigenvectors of input symmetric matrix</td>
</tr>
<tr>
<td>HIST</td>
<td>Prints histogram of residuals</td>
</tr>
<tr>
<td>INVXTX</td>
<td>Inverts symmetric matrix</td>
</tr>
<tr>
<td>LOC</td>
<td>When given row and column indices of symmetric matrix element, it computes location this element would have if only lower triangular part were stored as vector.</td>
</tr>
<tr>
<td>MATINV</td>
<td>Controls inversion process; computes regression coefficients; computes eigenvalues and eigenvectors of ( X'X ) if requested</td>
</tr>
<tr>
<td>MFIX</td>
<td>Prints ( X'X ) and computes and prints ( C )</td>
</tr>
<tr>
<td>NEWRAP</td>
<td>Executes overall problem control; computes replication error; controls deletion of variables when given results of ( t )-test</td>
</tr>
<tr>
<td>OUTPLT</td>
<td>Computes residuals at observed points and plots them. Compute chi-squared statistic</td>
</tr>
<tr>
<td>PREDCT</td>
<td>Computes predicted values, variances, and standard deviations of regression line and further observations at specified points</td>
</tr>
<tr>
<td>RECT</td>
<td>Writes rectangular matrix</td>
</tr>
<tr>
<td>RSTATS</td>
<td>Computes regression statistics and writes regression and lack-of-fit analysis of variance tables</td>
</tr>
<tr>
<td>SUMUPS</td>
<td>Constructs ( X'X ) and ( X'y ) matrices one observation at a time, in double precision</td>
</tr>
<tr>
<td>TRAN</td>
<td>Performs transformations</td>
</tr>
<tr>
<td>TRIANG</td>
<td>Writes lower triangular part of symmetric matrix</td>
</tr>
<tr>
<td>TTEST</td>
<td>Computes ( t )-statistics and their significance levels; determines which variable should be deleted</td>
</tr>
</tbody>
</table>

Call Structure of Program

The call structure of the program is illustrated in figure 7.
Dictionary of Program

<table>
<thead>
<tr>
<th>FORTRAN name</th>
<th>Mathematical symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTDEV</td>
<td>$e_i$</td>
<td>Error in observation $i$; difference between observed and predicted response</td>
</tr>
<tr>
<td>B</td>
<td>$b$</td>
<td>Regression coefficients other than the constant</td>
</tr>
<tr>
<td>BO</td>
<td>$b_0$</td>
<td>Constant regression coefficient</td>
</tr>
<tr>
<td>BZERO</td>
<td></td>
<td>Logical variable set to T if constant $b_0$ coefficient should be in regression model</td>
</tr>
<tr>
<td>CHISQ</td>
<td>$\chi^2$</td>
<td>Chi-squared statistic</td>
</tr>
<tr>
<td>CON</td>
<td></td>
<td>Constants used in transformations, and results of transformations</td>
</tr>
<tr>
<td>DELETE</td>
<td></td>
<td>Logical variable set to T when deletion of terms is desired</td>
</tr>
<tr>
<td>DUMMY</td>
<td></td>
<td>Extra array used for plotting data</td>
</tr>
<tr>
<td>ECONMY</td>
<td></td>
<td>Logical variable indicating suppress printout of $X'X$, $X'X$ deviations, and $C$ if $T$</td>
</tr>
<tr>
<td>ERRMS</td>
<td>$\sigma^2$</td>
<td>Estimate of $\sigma^2$ used in hypothesis tests</td>
</tr>
<tr>
<td>FMT</td>
<td></td>
<td>Variable input format</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>FMTTRI</td>
<td>Format for printing matrix</td>
<td></td>
</tr>
<tr>
<td>IDENT</td>
<td>First identification printed at top of each page</td>
<td></td>
</tr>
<tr>
<td>IDOUT</td>
<td>Original sequence number of each term relating reduced models to original</td>
<td></td>
</tr>
<tr>
<td></td>
<td>model</td>
<td></td>
</tr>
<tr>
<td>IFCHI</td>
<td>Logical variable set to T if residual computations and plots are desired</td>
<td></td>
</tr>
<tr>
<td>IFSSR</td>
<td>Logical variable set to T if sequential regressions are desired</td>
<td></td>
</tr>
<tr>
<td>IFTT</td>
<td>Logical variable set to T if t-statistics are desired</td>
<td></td>
</tr>
<tr>
<td>IFWT</td>
<td>Logical variable set to T if all weights of observations are 1.0</td>
<td></td>
</tr>
<tr>
<td>INPUT</td>
<td>Input logical tape unit number for data</td>
<td></td>
</tr>
<tr>
<td>INPUT5</td>
<td>Set equal to 5 to denote input device is card reader</td>
<td></td>
</tr>
<tr>
<td>INTER</td>
<td>Tape unit where input data is stored for use in OUTPLT</td>
<td></td>
</tr>
<tr>
<td>IOUT</td>
<td>Sequence number of term among those remaining which is to be deleted</td>
<td></td>
</tr>
<tr>
<td>JCOL</td>
<td>Total number of independent and dependent terms in regression model</td>
<td></td>
</tr>
<tr>
<td>KONNO</td>
<td>Number of constants originally supplied for transformations</td>
<td></td>
</tr>
<tr>
<td>LENGTH</td>
<td>Number of locations in matrix storage area currently needed</td>
<td></td>
</tr>
<tr>
<td>LIST</td>
<td>Set equal to 6 to denote output device is printer</td>
<td></td>
</tr>
<tr>
<td>NARAY</td>
<td>Number of replications per replicate set</td>
<td></td>
</tr>
<tr>
<td>NCON</td>
<td>Array containing addresses in CON array for use in transformations</td>
<td></td>
</tr>
<tr>
<td>NERROR</td>
<td>Degrees of freedom for error mean square estimate</td>
<td></td>
</tr>
<tr>
<td>NLOF</td>
<td>Degrees of freedom for estimating variance due to lack of fit</td>
<td></td>
</tr>
<tr>
<td>NODEP</td>
<td>Number of dependent variables</td>
<td></td>
</tr>
<tr>
<td>NOOB</td>
<td>Number of observations</td>
<td></td>
</tr>
</tbody>
</table>

44
<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOTERM</td>
<td>J</td>
</tr>
<tr>
<td>NOVAR</td>
<td>K</td>
</tr>
<tr>
<td>NPDEG</td>
<td>NPDEG</td>
</tr>
<tr>
<td>NREG</td>
<td>J</td>
</tr>
<tr>
<td>NRES</td>
<td>N - J - D</td>
</tr>
<tr>
<td>NTERM</td>
<td>Array containing locations of terms in CON array that should be in regression model</td>
</tr>
<tr>
<td>NTOT</td>
<td>N - D</td>
</tr>
<tr>
<td>NTRAN</td>
<td>Array containing transformation codes for use in performing transformations</td>
</tr>
<tr>
<td>NTRANS</td>
<td>Number of transformations to perform</td>
</tr>
<tr>
<td>NWHERE</td>
<td>Location in X array of first dependent variable; used in prediction routine to adjust for deleted terms</td>
</tr>
<tr>
<td>NXCOD</td>
<td>Array containing addresses of variables (or terms with address &gt;60) for use in transformations</td>
</tr>
<tr>
<td>P</td>
<td>Probability that interval (-t, t) must have before a term is considered to be significant</td>
</tr>
<tr>
<td>PNCH</td>
<td>Logical variable set to T if residuals are to be punched</td>
</tr>
<tr>
<td>POOLED</td>
<td>SSQ(REP)</td>
</tr>
<tr>
<td>PREDCT</td>
<td>Logical variable set to T if prediction option is desired</td>
</tr>
<tr>
<td>REPS</td>
<td>Logical variable set to T if there are replicate sets in data</td>
</tr>
<tr>
<td>REPVAR</td>
<td>Array containing replication variance of each dependent term</td>
</tr>
<tr>
<td>RESMS</td>
<td>Array containing residual mean square or variance of each dependent term</td>
</tr>
<tr>
<td>RNLOF</td>
<td>Reciprocal of degrees of freedom for lack of fit</td>
</tr>
<tr>
<td>RNREG</td>
<td>Reciprocal of degrees of freedom for regression</td>
</tr>
</tbody>
</table>
RNRES
Reciprocal of degrees of freedom for residual

RWT
Reciprocal of total weight

SATRTD
Logical variable indicating that there are no degrees of freedom for residual if T

STORYX
Logical variable set to T if eigenvectors and eigenvalues of X'X are to be computed and printed

SUMX
\[ \sum x, \sum y \]
Array containing sums of independent and dependent terms

SUMX2
\[ \sum x^2, \sum y^2 \]
Array containing sum of squared independent and dependent terms

SUMXX
X'X
Sums of squares and crossproducts matrix, and variance-covariance matrix of independent terms

SUMXXI
\( (X'X)^{-1} \)
Inverse of variance-covariance matrix of independent variables

SUMXY
X'y
Array containing sums of crossproducts of independent terms with dependent terms

TOTWT
\[ w_i \]
Sum of weight of observations

X
Before transformations are performed, this contains the variables as read in. After transformations are performed, appropriate data from CON array are placed here according to information on TERMS cards.

XCHK
Array used in checking if all values of independent terms are the same within a replicate set

ZEAN
E(X), \( \mu \)
Expected or mean value (or X)

Program Listing

SIBFTC BLDV

BLOCK DATA
COMMON /FRMTS/ FMT(13),FMTTRI(14)
COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
X,DUMMY(2300)
DOUBLE PRECISION B,SUMXY,SUMXX,SUMXXI
COMMON/MED/B0(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
X CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
C THIS IS NEWRAP, MAIN PROGRAM FOR REGRESSION ANALYSIS PROVIDING
C INTERNAL EVALUATION OF RESULTS.

COMMON /FRMTS/ FMT(13), FMTTRI(14)
COMMON /BIG/ B(60, 9), SUMXY(60, 9), SUMXX(1830), SUMXI(1830)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON /MED/ B(60, 9), SUMXY(60, 9), SUMXX(1830), SUMXI(1830),
X CON(99), ERRMS(9), IDENT(13), IDOUT(60), NCON(100), NTERM(69),
X NTRAN(100), NXCOD(100), POOLED(9), REPVAR(9), RESMS(9), X(99)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON /SMALL/ BYPASS, BZERO, DELETE, FIRST, IFCHI, IFSSR,
X IFFT, IFWT, INPUT, INPUT5, INTER,
X ISTRAT, JCDS, KONNO, LENGTH, LIST,
X NERROR, NODEP, NODEB, NOTER,
X NOVAR, NPDEG, NKES, NTRANS, NWHERE,
X P, PREDCT, REPS, RWT,
X STORY, STORY1, STORYX, TOTWT, WEIGHT,
X ERRFXD, ECONMY, IOU, ICOL
LOGICAL ECONMY
DOUBLE PRECISION RWT, TOTWT, WEIGHT
DATA INTER/3/, INPUT5/5/, LIST/6/
DATA (FMTTRI(I), I=1, 4)/6H(SHRO, 6HW 15129 6HX*(8Gl, 6H5.6)) /
C MAXPLT SHOULD BE THE NUMBER OF SINGLE LENGTH WORDS IN COMMON/BIG/
C BEGINNING AT THE FIRST LOCATION OF SUMXY
DATA MAXPLT/10700/
END

$IBFTC NEWRAP

C THIS IS NEWRAP, MAIN PROGRAM FOR REGRESSION ANALYSIS PROVIDING
C INTERNAL EVALUATION OF RESULTS.

COMMON /FRMTS/ FMT(13), FMTTRI(14)
COMMON /BIG/ B(60, 9), SUMXY(60, 9), SUMXX(1830), SUMXI(1830)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON /MED/ B(60, 9), SUMXY(60, 9), SUMXX(1830), SUMXI(1830),
X CON(99), ERRMS(9), IDENT(13), IDOUT(60), NCON(100), NTERM(69),
X NTRAN(100), NXCOD(100), POOLED(9), REPVAR(9), RESMS(9), X(99)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON /SMALL/ BYPASS, BZERO, DELETE, FIRST, IFCHI, IFSSR,
X IFFT, IFWT, INPUT, INPUT5, INTER,
X ISTRAT, JCDS, KONNO, LENGTH, LIST,
X NERROR, NODEP, NODEB, NOTER,
X NOVAR, NPDEG, NKES, NTRANS, NWHERE,
X P, PREDCT, REPS, RWT,
X STORY, STORY1, STORYX, TOTWT, WEIGHT,
X ERRFXD, ECONMY, IOU, ICOL
LOGICAL ECONMY
DOUBLE PRECISION RWT, TOTWT, WEIGHT
DATA INTER/3/, INPUT5/5/, LIST/6/
DATA (FMTTRI(I), I=1, 4)/6H(SHRO, 6HW 15129 6HX*(8Gl, 6H5.6)) /
C MAXPLT SHOULD BE THE NUMBER OF SINGLE LENGTH WORDS IN COMMON/BIG/
C BEGINNING AT THE FIRST LOCATION OF SUMXY
DATA MAXPLT/10700/
END

C THIS IS NEWRAP, MAIN PROGRAM FOR REGRESSION ANALYSIS PROVIDING
C INTERNAL EVALUATION OF RESULTS.

COMMON /FRMTS/ FMT(13), FMTTRI(14)
COMMON /BIG/ B(60, 9), SUMXY(60, 9), SUMXX(1830), SUMXI(1830)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON /MED/ B(60, 9), SUMXY(60, 9), SUMXX(1830), SUMXI(1830),
X CON(99), ERRMS(9), IDENT(13), IDOUT(60), NCON(100), NTERM(69),
X NTRAN(100), NXCOD(100), POOLED(9), REPVAR(9), RESMS(9), X(99)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON /SMALL/ BYPASS, BZERO, DELETE, FIRST, IFCHI, IFSSR,
X IFFT, IFWT, INPUT, INPUT5, INTER,
X ISTRAT, JCDS, KONNO, LENGTH, LIST,
X NERROR, NODEP, NODEB, NOTER,
X NOVAR, NPDEG, NKES, NTRANS, NWHERE,
X P, PREDCT, REPS, RWT,
X STORY, STORY1, STORYX, TOTWT, WEIGHT,
X ERRFXD, ECONMY, IOU, ICOL
LOGICAL ECONMY
DOUBLE PRECISION RWT, TOTWT, WEIGHT
DATA INTER/3/, INPUT5/5/, LIST/6/
DATA (FMTTRI(I), I=1, 4)/6H(SHRO, 6HW 15129 6HX*(8Gl, 6H5.6)) /
C MAXPLT SHOULD BE THE NUMBER OF SINGLE LENGTH WORDS IN COMMON/BIG/
C BEGINNING AT THE FIRST LOCATION OF SUMXY
DATA MAXPLT/10700/
END
C READ IDENTIFICATION CARD AND OPTIONS CARD

READ(INPUT5,110) I,IDENT
WRITE(List,111) IDENT
FIRST=.TRUE.
ERRFX=.FALSE.
113 IF(I) 120,120,115
115 READ(INPUT5,300) FMT
WRITE(List,301) FMT
I=I-1
GO TO 113
120 READ(INPUT5,1282) NDVAR,NDDEP,NTERM,NODE,NTKEEP
WRITE(List,1283) NDVAR,NDDEP,NTERM,NODE
IF(NTKEEP.NE.0) WRITE(List,1307) NTKEEP
READ(INPUT5,117) BZERO, IFTT, IFWT, IFCHI, STORYX, IFSSR,
X ECONMY, ISTRAT, PNCH
WRITE(List,118) BZERO, IFTT, IFWT, IFCHI, STORYX, IFSSR, ECONMY,
XISTRAT, PNCH
C THESE ARE INITIALIZATIONS MADE BEFORE EACH SET OF DATA
C ICOL DETERMINES THE NUMBER OF VARIABLES READ PER OBSERVATION
C JCOL IS THE NUMBER OF TERMS IN THE TOTAL REGRESSION EQUATION
C LENGTH IS THE NUMBER OF STORES NEEDED FOR THE MATRICES
LENGTH= NOTERM*(NOTERM+1)/2
ICOL=VOVAR + NODEP
JCOL = NOTERM + NODEP
NWHERE= NOTERM
REWIND INTER
DO 140 J=1,60
IDOUT(J) = J
140 NTERM(J)=J
DO 145 J=1,100
NXCOD(J)=J
NTRAN(J)=0
145 NCON(2*J)=J
C
IF(BZERO) WRITE(List,190)
IF(.NOT.BZERO) WRITE(List,170)
C READ(INPUT5,282) NTRANS,KONNO
IF(NTRANS.EQ.0) GO TO 255
READ (INPUT5,260) (TRK(K),K=1,JCOL)
WRITE(List,235) (TRK(K),K=1,JCOL)
READ (INPUT5,230)(NXCOD(I),NTRAN(I),NCON(2*I-1),NCON(2*I),I=1,NTR
AANS )
WRITE(List,240) (NXCOD(I),NTRAN(I),NCON(2*I-1),NCON(2*I),I=1,
X NTRANS)
IF(KONNO) 255,255,250
250 READ (INPUT5,260)(CON(I),I=1,KONNO)
WRITE(List,262) (CON(I),I=1,KONNO)
C
255 READ(INPUT5,257) DELETE,P
IF(DELETE) IFTT=.TRUE.
C IF THERE ARE REPLICATED POINTS READ IN THE NUMBER OF POINTS AND
C THE NUMBER OF REPLICATIONS. SINGLE DATA POINTS ARE DATA POINTS
C REPLICATED ONCE. IMPLIED HERE.
C READ(INPUT5,257) REPS
XSAVE=.FALSE.
265 IF(.NOT.REPS) GO TO 290
READ(INPUT5,282) IREP,(NARAY(I),I=1,IREP)
WRITE(LIST,284) IREP
WRITE(LIST,283) (NARAY(I),I=1,IREP)
NPDCG=0
IREP=1
IC=NARAY(1)
XSAVE=.TRUE.
DO 315 I=1,NODEP
POOLED(I)=0.0
S(I)=0.0
315 SSQ(I)=0.0
C READ VARIABLE FORMAT FOR DATA
290 READ(INPUT5,110) INPUT,FMT
WRITE(LIST,111) FMT
310 TOTWT=0.000
WEIGHT=1.000
WRITE(LIST,301) IDENT
C READ IN INPUT VARIABLES
DO 490 J=1,NOOR
330 IF(.NOT.IFHTI GO TO 350
340 REAL) (INPUTSFMT) (X(I),I=1,ICOL)
GO TO 360
350 READ (INPUT,FMT)(X(I),I=1,ICOL), WEIGHT
360 CONTINUE
IF(ECONMY) WRITE(LIST,381) J,(X(I),I=1,ICOL)
381 FORMAT(1H14,9G14.6/(5X,9G14.6))
IF(INTRANS.EQ.0) GO TO 450
IF(ECONMY) GO TO 390
WRITE(LIST,370) WEIGHT,J
WRITE(LIST,380) (X(I),I=1,ICOL)
390 CALL TRANS
420 DO 430 K=1,JCOL
I=TERM(K)
X(K) = CON(I)
430 CONTINUE
450 CONTINUE
IF(ECONMY) GO TO 4609
WRITE(LIST,460) J
461 WRITE (LIST,380) (X(I),I=1,ICOL)
4609 CONTINUE
IF(.NOT.XSAVE) GO TO 4611
DO 4610 K=1,NOTERM
4610 XCHK(K)=X(K)
XSAVE=.FALSE.
4611 CONTINUE
C
C COMPTE THE ERROR VARIANCE FROM REPLICATED DATA
159
IFI (.NJT. REPS) GO TO 480
160
IGOTO = 1
161
IF (NARY (IREP) GT. 1) IGOTO = 2
162
IFI (.J. 3E. IC) WRITE (6, 462) IREP
163
DO 475 I = 1, NODEP
164
IFI (.I-1) 4629, 4629, 464
165
4629 DO 463 K = 1, NOTERM
166
IFI (.K. NE. XCHK(K)) GO TO 2001
167
463 CONTINUE
168
464 CONTINUE
169
KBAR = NOTERM + I
170
S(I) = S(I) + X(KBAR)
171
SSQ(I) = SSQ(I) + X(KBAR)**2
172
IFI (.I-IC) 475, 465, 465
173
465 GO TO (468, 466), IGOTO
174
466 ZEAN(I) = S(I)/FLOAT(NARY(IREP))
175
SSQ(I) = SSQ(I) - ZEAN(I)*S(I)
176
POOLED(I) = POOLED(I) + SSQ(I)
177
WRITE (LIST, 467) I, SSQ(I), S(I), ZEAN(I)
178
468 IF (.I.LT. NODEP) GO TO 469
179
NPDEG = NPDEG + NARY (IREP) - 1
180
IREP = IREP + 1
181
IFI (.IC.EQ. NODEP) GO TO 496
182
IC = IC + NARY (IREP)
183
WRITE (LIST, 467)
184
469 S(I) = 0.0
185
SSQ(I) = 0.0
186
XSAVE = .TRUE.
187
475 CONTINUE
188
C******************************************************************************
189
C CALCULATE SUMS, SUMS OF SQUARES AND SUMS OF CROSS PRODUCTS.
190
480 CALL SUMUP
191
490 CONTINUE
192
C 490 CONTINUE IS THE END OF THE LOOP FOR READING DATA CARDS
193
IFI (.NJT. REPS) GO TO 496
194
DO 493 I = 1, NODEP
195
REPVAR(I) = POOLED(I)/FLOAT(NPDEG)
196
493 CONTINUE
197
496 CONTINUE
198
C******************************************************************************
199
C ALL DATA HAS BEEN READ IN AND THE XTRANSPOSE X AND XTRANSPOSE Y
200
C MATRIX HAVE BEEN CALCULATED.
201
C NOW WRITE THE MATRICES
202
CALL WFIX
203
REWIND INTER
204
GO TO 640
205
C******************************************************************************
206
C THIS CODING DELETES THE DATA FROM THE SUMXX MATRIX
207
C CORRESPONDING TO THE INDEPENDENT TERM DELETED
208
6500 CONTINUE
209
IR = IDJT-1
210
IC = NTERM - IOUT
211
IFI (.IC.EQ. 0) GO TO 6700
212
INOUCH = IOUT*IR/2
213
INew = INOCH
214
IOL(I) = INEW + IOUT
215
IRC = 0
216
IBC = 0
217
ITC = 0
218
220
DO 6630 I=IOLD,LENGTH
  INEW = INEW + 1
  IOLD = IOLD + 1
  IF (ITC .GT. 0) GO TO 6540
  IRC = IRC + 1
  IF (IRC .EQ. IR) GO TO 6530
  SUMXX(INEW) = SUMXX(IOLD)
  GO TO 6600
6530 IBC = IBC + 1
  ITC = IBC
  IOLD = IOLD + 1
  IRC = 3
6540 ITC = ITC - 1
  SUMXX(INEW) = SUMXX(IOLD)
6600 CONTINUE
6700 LENGTH = LENGTH - NOTERM
  NOTERM = NOTERM - 1
  JCOL = VOTERM + NODEP
C**************************************************************************
C INVERT THE SUMXX MATRIX AND COMPUTE REGRESSION COEFS
C AND SJDMS OF SQUARES DUE TO REGRESSION IN THE MATRIX INVERSION
C ROUTINE
640 CONTINUE
  CALL MATINV
  FIRST=.FALSE.
C**************************************************************************
C WRITE(XTX1 INVERSE, THIS MATRIX TIMES ERROR MEAN SQUARE (ERRMS)
C IS THE VARIANCE-COVARIANCE MATRIX OF REGRESSION COEFFICIENTS.
C IF(ECONMY) GO TO 970
C WRITE(LIST,700)
C CALL TRIXANG(X,SUMXXI,NOTERM,8,FMTTRI,2)
C**************************************************************************
C IF A VARIABLE HAS BEEN DELETED ADJUST COUNTERS AND RECOMPUTE THE
C REGRESSION, IF NO VARIABLE HAS BEEN DELETED CONTROL WILL PASS
C FROM THE TTEST ROUTINE TO THE CHI-SQUARE OPTION.
970 CONTINUE
  IF (.NOT.IFFT) GO TO 1020
  WRITE (LIST,301) IDENT
  CALL TTEST(1020,NTKEEP)
  IF (NODEP. EQ. 1) 985,990,985
  WRITE(LIST,986) NODEP
  NODEP=1
985 J=JCOL-1
  DO 995 K=IOUT,J
  NTERM(K) = NTERM(K+1)
  ZEAN(K) = ZEAN(K+1)
  SUMX(K) = SUMX(K+1)
  SUMX2(K) = SUMX2(K+1)
  IDOUT(K) = IDOUT(K+1)
  SUMXY(K,1) = SUMXY(K+1,1)
995 CONTINUE
  IF (NOTERM. EQ. 1) GO TO 1000
  GO TO 6500
1000 WRITE(LIST,1005)
  NOTERM=0
  GO TO 1035
C**************************************************************************
C**************************************************************************
1020 IF(.NOT.IFCH1) GO TO 1035
1030 WRITE(LIST,301) IDEWT
   CALL JUTPLT(PNCH)

C******************************************************************************
1035 READ(INPUT5,117) PREDCT
   IF(.NOT.PREDCT) GO TO 100
   CALL PREDIC
1040 GO TO 100

C******************************************************************************
2001 WRITE(LIST,1306)
   STOP

C******************************************************************************
8001 FORMAT(1H1)
8002 FORMAT(1H2)
   110 FORMAT (I2,13A6)
   111 FORMAT (1H1,13A6,A2)
   117 FORMAT(7L1,I1,L1)
   118 FORMAT(1H 7L1,I1,L1)
   170 FORMAT(33H THERE IS NO BO TERM IN THE MODEL)
   190 FORMAT(26H THERE IS A BO TO ESTIMATE )
   230 FORMAT(40I2)
   235 FORMAT(11H NTERM(K) = /1H 3014))
   240 FORMAT(25H THE TRANSFORMATIONS ARE /1H 5(4I4,5X)))
   257 FORMAT( 1L1, F3.3)
   260 FORMAT(5E15.7)
   262 FORMAT(19H THE CONSTANTS ARE /(1H 8G15.7)))
   282 FORMAT(1H2014)
   283 FORMAT(1H 2014)
   284 FORMAT(11H THERE ARE I5,16H REPLICATE SETS )
   300 FORMAT(13A6,1A2)
   301 FORMAT(1H 13A6,1A2)
   370 FORMAT(1H0,29H OBSERVED VARIABLES, WEIGHT = G14.6,6X,15H OBSERVATION
   1 = ,15)
   380 FORMAT(1H 9G14.6)
   460 FORMAT(1H ,37H TERMS OF THE EQUATION, OBSERVATION = ,15)
   462 FORMAT(18H K** REPLICATE SET IS 15,3X,100(1H*))
   4671 FORMAT(1H125(1H*))
   467 FORMAT(14H DEP. VAR. 16,8H SSQ=G14.7,8H SUM=G14.7,8H M
   XEAN= G14.7)
   540 FORMAT(1H 8G14.7)
   560 FORMAT(21H2X TRANSPOSE X MATRIX )
   670 FORMAT(25H2CORRELATION COEFFICIENTS )
   700 FORMAT(32H2X TRANSPOSE X) INVERSE MATRIX )
   986 FORMAT(39H THE NUMBER OF DEPENDENT VARIABLES WAS 13,83H IT IS BE
   XING SET TO ONE AND THE REJECTION OPTION EXERCISED ON DEPENDENT VAR
   XIBLE 1 )
   1005 FORMAT(39H THERE IS NO EVIDENCE OF A REGRESSION, /331
   X 74H USE THE MEAN RESPONSE FOR THE BEST ESTIMATE OF THE DEPEND
   XENT VARIABLES(5)* )
   1282 FORMAT(314,15,14)
   1283 FORMAT(1I1H 314,15)
   1306 FORMAT(40H REPLICATE SETS ARE NOT GROUPED PROPERLY )
   1307 FORMAT(11H THE FIRST 12,64H TERMS OF THE MODEL WILL BE RETAINED R
   XEGARDLESS OF SIGNIFICANCE )
END

52
SUBROUTINE MATINV

C*************************************************************************

C PURPOSE
C 1) COMPUTE EIGENVALUES AND EIGENVECTORS OF (X-TRANSPOSE X)
   MATRIX IF REQUESTED. (STORYX=.TRUE.)
C 2) COMPUTE (X TRANSPOSE X) INVERSE
C 3) COMPUTE REGRESSION COEFFICIENTS
C 4) COMPUTE OTHER REGRESSION STATISTICS
C
C SUBROUTINES NEEDED
C BORD
C EIGEN
C INVXTX
C RECT
C RSTATS
C TRIANG
C
C REMARKS
C THE EIGENVALUES ARE COMPUTED AS AN AID IN DETERMINING THE
C CONDITION OF THE SYSTEM OF EQUATIONS FOR THE REGRESSION
C COEFFICIENTS. EXAMINATION OF THEM AND THEIR ASSOCIATED
C EIGENVECTORS MAY SHOW THAT CERTAIN SETS OF INDEPENDENT
C VARIABLES ARE HIGHLY CORRELATED AND NOT EASILY LIABLE TO
C INDEPENDENT STUDY.
C
SUBROUTINE MATINV

C*************************************************************************

COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY2(9),ZEAN(69),
   X CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
   X NTRAN(100),NCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
DOUBLE PRECISION BD, SUMX, SUMX2, ZEAN
COMMON /FMTS/ FMT(13),FMTTRI(14)
COMMON /SMALL/ BYPASS,BZERO,DELETE, FIRST, IFCHI, IFSSR,
   X IFTT, IFF, INPUT, INPUT5, INTER,
   X ISTRAT, JCOL, KONNO, LENGTH, LIST,
   X NERRR, NDEP, NOOB, NOTERM,
   X NOVAR, NPE, NRES, NTRANS, NWHERE,
   X P, PREDCT, REPS, RWT,
   X STORYI, STORYC, STORYX, TOTWT, WEIGHT,
   X ERRFXD, ECONMY, IOUT, ICOL
LOGICAL ECONMY
LOGICAL BYPASS, BZERO, DELETE, IFCHI,
   X IFSSR, IFTT, IFF, REPS, PREDCT,
   X STORYI, STORYX, FIRST, ERRFXD
DOUBLE PRECISION RWT, TOTWT, WEIGHT
DIMENSION A(1),C(1),XTX(3)
EQUIVALENCE (A, SUMXI), (C, SUMXXI(915))
DOUBLE PRECISION SUM
DATA (XTX(I),I=1,3) /6HX TRAN, 6HSPOSE , 6HX /

C*************************************************************************

C ORDER= NOTERM
IF(NOTERM-1) 10,10,12
10 SUMXXI(I) = 1.0/SUMX2(I)
GO TO 350

12 IF(*N3To STORYX) GO TO 30
DO 14 I=1,LENGTH
A(I) = SUMXX(I)
14 CONTINUE

16 CALL EIGEN(A,C , IORDER,0)
WRITE(LIST,17)(XTX(I),I=1,3)
J=0
DO 18 I=1, IORDER
J=J+1
18 A(I) = A(J)
WRITE(LIST,19) (A(I), I=1, IORDER)
WRITE(LIST,20)
CALL RECT(IORDER, IORDER, IORDER, IORDER, C, X ,FMTTRI, 1)
30 DO 35 I=1, LENGTH
35 SUMXXI(I) = SUMXXII

C**************************************************************************
C NO SUBMODELS TO ANALYZE SO INVERT A DIRECTLY BY GAUSS
49 IF(FSSR) GO TO 50
CALL INVXTX(SUMXXII, NOTERM, D, 1.0)
GO TO 60
C**************************************************************************
C SUBMODELS HAVE BEEN REQUESTED SO WE USE BORDERING
50 IORDER = 0
55 IORDER = IORDER + 1
CALL BORD(IORDER, SUMXXI)
60 CONTINUE
C**************************************************************************
C COMPUTE COEFFICIENTS AND PRINT THEM
350 DO 373 J=1, NODEP
DO 373 K=1, IORDER
B(K,J) = 0.000
DO 373 L=1, IORDER
CALL LOC(L,K, IR)
B(K,J) = B(K,J) + SUMXXI(IR)*SUMXY(L,J)
370 CONTINUE

C WRITE(LIST, 380) IDENT
WRITE(LIST, 382)
IF(*NT, BZERO) GO TO 400
DO 390 J=1, NODEP
SUM = 0.000
KBAR = NOTERM + J
DO 385 K=1, IORDER
SUM = SUM + B(K,J)*ZEAN(K)
385 CONTINUE
BO(J) = ZEAN(KBAR) - SUM
390 CONTINUE
WRITE(LIST, 395)
WRITE(LIST, 397) (BO(K), K=1, NODEP)
400 WRITE(LIST, 410)
DO 430 J=1, IORDER
WRITE(LIST, 432) IDOUT(J), (B(J,K), K=1, NODEP)
430 CONTINUE
C******************************************************************************
C COMPUTE REGRESSION STATISTICS IN RSTATS
C
CALL RSTATS(IORDER)
C******************************************************************************
C IF IORDER IS LESS THAN NOTERM WE HAVE USED THE BORDERING OPTION
C AND MUST GO BACK TO FINISH.
C
IF(IORDER-NOTERM) 55,500,500
500 STORYX=.FALSE.
IFSSR=.FALSE.
RETURN
17 FORMAT(34H2THE FOLLOWING ARE EIGENVALUES OF 2A6,A1, 7H MATRIX)
19 FORMAT(1H 8G16.7)
20 FORMAT(132H2THIS IS THE MODAL MATRIX OR MATRIX OF EIGENVECTORS. EIGENVECTORS ARE WRITTEN IN COLUMNS LEFT TO RIGHT IN SAME ORDER AS 2EIGENVALUES )
380 FORMAT(1H13A6,1A2)
382 FORMAT( 6H EACH COLUMN CONTAINS THE COEFFICIENTS FOR ONE DEPENDENT XT TERM )
395 FORMAT(20H CONSTANT TERM (BO ) )
397 FORMAT(4X,9G14.6)
410 FORMAT(36H REGRESSION COEFFICIENTS (B1,...,BK ) )
432 FORMAT(1H 3,9G14.6)
END

$IBFC TT-STX

C******************************************************************************
C SUBROJTIME TTEST
C
C PURPOSE
C COMPUTE THE T-STATISTICS FOR EACH REGRESSION TERM AND ITS TWO TAILED SIGNIFICANCE LEVEL. THEN DETERMINE THE TERM WITH THE LEAST SIGNIFICANCE AND RETURN THIS INFORMATION TO NEWRAP
C
C******************************************************************************
C SUBROJTIME TTEST(*,NTKEEP)***************

C******************************************************************************
COMMON /FRMTS/ FMT(13),FMTTRI(14)
COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
X , DUMMY(1)
DOUBLE PRECISION B,SUMXY,SUMXX,SUMXI
COMMON/MED/B0(9),SUMX(69),SUMX2(69),SUMY2(9),ZEN(69),
X CON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
X NTRAN(100),NXGON(100),POOLED(9),REPSV(9),RESMS(9),X(99)
DOUBLE PRECISION BD,SUMX,SUMXX2,SUMY2,ZEN
COMMON/SMALL/ BYPASS,BZERO,DELETE, FIRST, IFCHI, IFSSR,
X IFIT, IFWT, INPUT, INPUTS, INTER,
X ISTRAT, JCOL, KONNO, LENGTH, LIST,
X NERROR, NOSEP, NOOB, NOTERM,
X NPOVAR, NPODEG, NRES, NTRANS, NWHERE,
X P, PREDICT, REPS, RWT,
X STORY1, STORY2, STORYX, TOTWT, WEIGHT, 29
X ERRFXD, ECONMY, IOUT, ICOL
LOGICAL ECONMY
LOGICAL SATRTD
LOGICAL BYPASS, BZERO, DELETE, IFCHI,
XIFSSR, IFTI, IFWT, REPS, PREDCT,
XSTORY, STORtY, STORYI, FIRST, ERRFXD
DQUBLE PRECISION RWT, TOTWT, WEIGHT

C

LOGICAL MAKEUP, NOZERO
DIMENSION T(35,13), PLEVEL(13)
DIMENSION DEVBI60.9), PROB(60,9), TT(60,9)
EQUIVALENCE (DUMMY(91), DEVBI60,9), T), (DUMMY(650), PROB)
EQUIVALENCE (P, PWANT)

C

DATA (PLEVEL(JJ), JJ=1,13) /0.10,0.20,0.30,0.40,0.50,0.60,0.70, 46
10.80,0.90,0.95,0.98,0.99,0.999 / 47
DATA (T(1, JJ), JJ=1,13) /0.158,0.325,0.510,0.727,1.000,1.376, 48
1.963,3.078,6.314,12.706,31.821,63.057,126.619, / 49
2 (T(2, JJ), JJ=1,13) /0.142,0.289,0.445,0.617,0.816,1.061, 50
3 (T(3, JJ), JJ=1,13) /0.137,0.277,0.424,0.584,0.765,0.978, 51
4 (T(4, JJ), JJ=1,13) /0.134,0.271,0.414,0.569,0.741,0.941, 52
5 (T(5, JJ), JJ=1,13) /0.130,0.267,0.408,0.559,0.727,0.920, 53
6 (T(6, JJ), JJ=1,13) /0.131,0.265,0.404,0.553,0.718,0.906, 54
7 (T(7, JJ), JJ=1,13) /0.130,0.263,0.402,0.549,0.711,0.896, 55
8 (T(8, JJ), JJ=1,13) /0.130,0.262,0.399,0.546,0.706,0.889, 56
9 (T(9, JJ), JJ=1,13) /0.129,0.261,0.398,0.543,0.703,0.883, 57
10 (T(10, JJ), JJ=1,13) /0.129,0.260,0.397,0.542,0.700,0.879, 58
10 (T(11, JJ), JJ=1,13) /0.129,0.259,0.396,0.540,0.697,0.876, 59
10 (T(12, JJ), JJ=1,13) /0.128,0.259,0.395,0.539,0.695,0.873, 60
10 (T(13, JJ), JJ=1,13) /0.128,0.259,0.394,0.538,0.694,0.870, 61
10 (T(14, JJ), JJ=1,13) /0.128,0.258,0.393,0.537,0.692,0.868, 62
10 (T(15, JJ), JJ=1,13) /0.128,0.258,0.392,0.536,0.691,0.866, 63
10 (T(16, JJ), JJ=1,13) /0.128,0.258,0.391,0.535,0.690,0.865, 64
10 (T(17, JJ), JJ=1,13) /0.128,0.257,0.390,0.534,0.689,0.863, 65
10 (T(18, JJ), JJ=1,13) /0.127,0.257,0.390,0.534,0.688,0.862, 66
10 (T(19, JJ), JJ=1,13) /0.127,0.257,0.390,0.533,0.688,0.861, 67
10 (T(20, JJ), JJ=1,13) /0.127,0.257,0.390,0.533,0.687,0.860, 68
10 (T(21, JJ), JJ=1,13) /0.127,0.257,0.390,0.532,0.686,0.859, 69
10 (T(22, JJ), JJ=1,13) /0.127,0.257,0.390,0.532,0.686,0.858, 70
T(I,J) = IS THE T-STATISTIC AT THE TABULATED DEGREES OF FREEDOM
(I) AND AT THE TABULATED PROBABILITY LEVELS (JJ).
II = DEGREES OF FREEDOM, EXCEPT FOR
II = 3 IS FOR 40 DEGREES
II = 32 IS FOR 60
II = 33 IS FOR 120
II = 34 IS FOR INFINITY

<table>
<thead>
<tr>
<th>JJ</th>
<th>PROBABILITY LEVEL</th>
<th>JJ</th>
<th>PROBABILITY LEVEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.10</td>
<td>8</td>
<td>0.80</td>
</tr>
<tr>
<td>2</td>
<td>0.20</td>
<td>9</td>
<td>0.90</td>
</tr>
<tr>
<td>3</td>
<td>0.30</td>
<td>10</td>
<td>0.95</td>
</tr>
<tr>
<td>4</td>
<td>0.40</td>
<td>11</td>
<td>0.98</td>
</tr>
<tr>
<td>5</td>
<td>0.50</td>
<td>12</td>
<td>0.99</td>
</tr>
<tr>
<td>6</td>
<td>0.60</td>
<td>13</td>
<td>0.999</td>
</tr>
<tr>
<td>7</td>
<td>0.70</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CALCULATE T STATISTICS

WRITE (LIST, 230)
230 FORMAT(140, 23H CALCULATED T STATISTICS /75H THE T STATISTICS CAN BE
1 USED TO TEST THE NET REGRESSION COEFFICIENTS B(I). )
DO 260 J = 1, NTERM
      DO 240 K = 1, NDEP
        TT(J,K) = ABS(B(J,K)/DEVB(J,K))
      240 CONTINUE
WRITE (LIST, 250) (TT(J,K), K = 1, NDEP)
250 FORMAT(1H 9G14.6)
260 CONTINUE

SEARCH THE TABLE OF TABULATED DEGREES OF FREEDOM
C
    MAKEJ = .FALSE.
    IF(NDEG=30)290,290,300
290 II=NDEG
    GO TO 400
300 IF(NDEG=40)310,320,330
310 FINV=1.0/40.0
    FM1INV=1.0/30.0
    MAKEJ = .TRUE.
320 II=31
    GO TO 400
330 IF(NDEG=60)340,350,360
340 FINV=1.0/60.0
    FM1INV=1.0/40.0
    MAKEJ = .TRUE.
350 II=32
    GO TO 400
360 IF(NDEG=120)370,380,390
370 FINV=1.0/120.0
    FM1INV=1.0/60.0
    MAKEJ = .TRUE.
380 II=33
    GO TO 400
390 II=34
    FINV=0.0
    FM1INV=1.0/120.0
    MAKEJ = .TRUE.
C
C 400 WRITE(LIST,410)
410 FORMAT(104H UNDER NULL HYPOTHESIS THE INTERVAL (-T,T) WHERE T IS G
XIVEN ABOVE, HAS APPROX PROBABILITY LISTED BELOW. /42H MINUS S
XIGN INDICATES PROB EXCEEDS .999. )
    IF(.NOT.MAKENU) GO TO 430
    FNDEG=NDEG
    DO 420 JJ=1,13
        T(35,JJ)=T(JJ,JJ1+((1.0/FNDEG - FINV)/(FM1INV-FINV))*T(JJ,JJ)
        1 -T(JJ,JJ))
    420 CONTINUE
    II=35
430 DO 550 J=1,NOTERM
    DO 540 K=1,NDDEP
    DO 440 JJ=1,13
        JJ=JJ
        IF(T(JJ,JJ)-TT(J,J))/440,450,460
    440 CONTINUE
    PROB(J,K)=-0.999
    GO TO 540
450 PROB(J,K)=PLEVEL(JJ)
    GO TO 540
460 IF(JJ.LE.9) GO TO 470
        JJ1=JJ-2
        JJ2=JJ-1
        JJ3=JJ
        GO TO 490
470 IF(JJ.LE.4)GO TO 480
        JJ1=JJ-1
        JJ2=JJ
        JJ3=JJ+1
        GO TO 490
480 JJ1=JJ
        JJ2=JJ+1
        JJ3=JJ+2
490 CONTINUE
C PERFORM A THREE-POINT LAGRANGE INTERPOLATION

X=ALO5(TT(J,K))
X1=ALOG(TT(II, JJ1))
X2=ALOG(TT(II, JJ2))
X3=ALOG(TT(II, JJ3))

IF(TT(J,K).LE.1.0) GO TO 500
Y1=ALOG(1.0-PLEVEL(JJ1))
Y2=ALOG(1.0-PLEVEL(JJ2))
Y3=ALOG(1.0-PLEVEL(JJ3))

GO TO 510

500 Y1=ALOG(PLEVEL(JJ1))
Y2=ALOG(PLEVEL(JJ2))
Y3=ALOG(PLEVEL(JJ3))

510 PROB(J,K) = ((X-X2)*(X-X3)*Y1)/((X1-X2)*(X1-X3)) + ((X-X1)*(X-X3)*Y2)/((X2-X1)*(X2-X3)) + ((X-X1)*(X-X2)*Y3)/((X3-X1)*(X3-X2))

IF(TT(J,K)-1.0) GO TO 520

520 PROR(J,K) = EXP(PROB(J,K))
GO TO 540

530 PROB(J,K) = 1.0 - EXP(PROB(J,K))

540 CONTINUE

C*******************************
C WRITE THE PROBABILITIES (1.0-ALPHA)
WRITE(LIST,550) IDOUT(J), (PROB(J,K), K=1, NDEP)
550 FORMAT(1H13,9(8X,F6.3))
560 CONTINUE

C*******************************
C LIST THE DESIRED VALUE OF PROBABILITY (PWANT)

570 PERCS=PWant*100.0
WRITE(LIST,580) PERCENT
580 FORMAT(1H10,36H THE DESIRED VALUE OF PROBABILITY IS ,F5.1, 8H PERCENT )

C DELTTE THE TERM WITH THE LOWEST COMPUTED PROBABILITY IF THAT
C PROBABILITY IS LESS THAN THAT DESIRED (PWANT)

IF(.NOT. DELETE) GO TO 660
IF(NTKEEP.EQ.NOTERM) GO TO 660
IOUT=J

590 AMIN=PWANT
JLO=MAXO(1,NTKEEP)
DO 620 J=JLO,NOTERM
IF(ABS(PROB(J,1))<PWANT) 600, 620

600 IF(ABS(PROB(J,1))<AMIN) 610, 620

610 AMIN=ABS(PROB(J,1))
IOUT=J
620 CONTINUE

IF(IOUT) 660, 660, 630

630 WRITE(LIST,550) IDOUT(IOUT)
650 FORMAT(1H10,36H THE TERM X(J,12,18H) IS BEING DELETED )
GO TO 670

C ALL VARIABLES REMAINING HAVE BEEN CONCLUDED SIGNIFICANT
660 RETURN
670 RETURN
END
SUBROUTINE KSTATS

PURPOSE
1) COMPUTE AND PRINT THE ANALYSIS OF VARIANCE TABLES ON
REGRESSION AND LACK-OF-FIT IF APPROPRIATE.
2) COMPUTE AND PRINT R-SQUARED AND STANDARD ERROR OF
ESTIMATE
3) COMPUTE AND PRINT SUMS OF SQUARES DUE TO EACH VARIABLE
IF IT WERE LAST TO ENTER REGRESSION
4) COMPUTE AND PRINT THE STANDARD DEVIATIONS OF EACH
REGRESSION COEFFICIENT.

SUBROUTINE RSTATS(IORDER)

COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMX(1830)
X DUMMY(1)
DOUBLE PRECISION B,SUMX,SUMXX,SUMXY
COMMON/MED/60(9),SUMX(69),SUMXX(69),SUMYY(2),(ZEAN(69),
X CONC(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
X NTRAN(100),NPOLED(100),REVPAR(9),RESMS(9),X(99)
DOUBLE PRECISION B0,SUMX,SUMXY,SUMYY,ZEAN
COMMON /FRMTS/ FMT(13)
COMMON / SMALL/ BYPASS,BZERO,DELETE, FIRST, IFCHI, IFSSR,
X IFTT, IFWT, INPUT, INPUT5, INTER,
X ISTRAT, JCOL, KONNO, LENGTH, LIST,
X VERROR, NODEP, NODEB, NOTERM,
X VOWAR, NPOED, NRES, NTRANS, N WHERE,
X P, PREDCT, REPS, RWT,
X STORI, STORYC, STORIX, TOTWT, WEIGHT,
X ERRFXD, ECONMY, IOUT, ICOL
LOGICAL ECONMY
LOGICAL SATRD
LOGICAL BYPASS, BZERO, DELETE, IFCHI, IFSSR
X IFSSR, IFTT, IFWT, REPS, PREDCT,
X STORY, STORYC, STORIX, FIRST, ERRFXD
DOUBLE PRECISION RWT,TOTWT,WEIGHT

DIMENSION SSQREG(9), SSQRES(9), REGMS(9),
S XLOF(9), XLOFMS(9), FRATIO(9), RSQD(9), R(9),
S SSLST(9), DEVB(60,9)
EQUIVALENCE (DUMMY(10),SSQRES),(DUMMY(19),REGMS),(DUMMY(37),XLOF)
S , (DUMMY(46),XLOFMS), (DUMMY(55),FRATIO), (DUMMY(64),RSQD)
S , (DUMMY(73),R), (DUMMY(82),SSLST), (DUMMY(91),DEVB)
S , (DUMMY(700),SSQREG)
DOUBLE PRECISION SSQREG

COMPUTE DEGREES OF FREEDOM AND RECIPROCALS

IF(NREG= IORDER)
NTOT= IFIX(TOTWT)-1
IF(IF(NOT.BZERO) NTOT= NTOT+1
NRES= VTOT-NREG
NLUP= VRES - NPDEG
RNREX= 1.0/FLOAT(NREG)
IF(NRES.EQ.0) GO TO 980
RNREX= 1.0/FLOAT(NREG)
SATRT=.FALSE.
IF(NF3F.EQ.0) GO TO 90

60
RNL0F=1.0/FLOAT(NLOF)
GO TO 100
90 SATRTD=.*TRUE.*
100 CONTINUE
NXTFRM=1ORDER
RNDB=RMT
C
C*******************************************************************************
C COMPJTE RESIDUAL SUM OF SQUARES, RESIDUAL VARIANCE, VARIANCE
C FROM REPLICATIONS IF APPROPRIATE, AND THE F-RATIO OF MEAN SQUARE
C LACK-OF-FIT AND MEAN SQUARE RESIDUALS.
C CONTINUE
DO 213 J=1,NODEP
SSQRES(J)=0.0
DO 203 I=1,NXTERM
SSQRE(I,J)=SSQREG(J) + B(I,J)*SUMXY(I,J)
CONT
I
SSQRES(J)=SUMY2(J)-SSQREG(J)
REGMS(J)= SSQREG(J)* RNREG
RESMS(J)= SSQRES(J)*RNRES
RSQD(J)=SSQREG(J)/SUMY2(J)
R(J)=SQRT(RSQD(J))
IF((.NOT.REPS).OR.*SATRTD) GO TO 210
XLOF(J)=SSQRES(J)-POOLED(J)
XLOFMS(J)= XLOF(J)*RNL0F
FRATIJ(J)=XLOFMS(J)/REPVAR(J)
210 CONTINUE
C
C*******************************************************************************
C DETERMINE WHICH ESTIMATE OF SIGMA SQUARED SHOULD BE USED IN
C HYPOTHESIS TESTS. PUT THE PROPER ONE IN ERRMS AND SET ERRFDX
C TO TRUE IF THE PRESENT VALUE IS TO BE USED FOR ALL FOLLOWING
C TESTS AND T-STATISTICS.
IOUT=ISTRAT
IF(ERRFXD) GO TO 250
IF(ISTRAT .NE. 3) GO TO 214
211 DO 213 J=1,NODEP
ERRMS(J)= RESMS(J)
NERR= NRES
IOUT=3
GO TO 250
214 IF(ISTRAT .NE. 1) GO TO 218
IF(.NOT.REPS) GO TO 211
DO 215 J=1,NODEP
215 ERRMS(J)= REPVAR(J)
NERR= NPDEG
ERRFXD=.*TRUE.*
IOUT=1
GO TO 250
218 IF(FIRST. .AND.* (IORDER .EQ. NOTERM)) GO TO 220
GO TO 211
220 ERRFXD=.*TRUE.*
DO 222 J=1,NODEP
222 ERRMS(J)= RESMS(J)
NERR= NRES
ISTRAT=2
IOUT=2
C
C*******************************************************************************
C WRITE ANOVA TABLES
250 DO 500 J=1,NODEP
IF(ERRMS(J .EQ. 0.0)) ERRMS(J)=1.0E-30
WRITE(LIST,1001) J
WRITE(LIST,1002)  
WRITE(LIST,1003) SSQREG(J), NREG, REGMS(J)  
WRITE(LIST,1004) SSQRES(J), NRES, RESMS(J)  
WRITE(LIST,1005)  
WRITE(LIST,1006) SUMY2(J), NTOT  
WRITE(LIST,1007)  
WRITE(LIST,1500) RSQD(J), R(J)  
STD=SQRT(RESMS(J))  
WRITE(LIST,1600) STD  
WRITE(LIST,1700) IOUT, ERRMS(J), NERROR  
F=REGMS(J)/ERRMS(J)  
WRITE(LIST,1750) F, NREG, NERROR  
IF((.NOT. REPS).OR.SATRTD) GO TO 500  
WRITE(LIST,2001)  
WRITE(LIST,1002)  
WRITE(LIST,2005) XLOF(J), NLOF, XLOFMS(J)  
WRITE(LIST,2006) POOLED(J), NPDEG, REPVAR(J)  
WRITE(LIST,1004) SSQRES(J), NRES, RESMS(J)  
WRITE(LIST,1005)  
WRITE(LIST,2008) FRATIO(J)  
WRITE(LIST,1007)  
500 CONTINUE  
C

**COMPUTE CONTRIBUTION OF EACH INDEPENDENT VARIABLE TO REG SUM OF SQUARES AS IF IT WERE LAST TO ENTER**

WRITE(LIST,370)  
IR= 0  
DO 8635 K=1,NXTERM  
IR= IR+K  
DO 8632 J=1,NODEP  
8632 SSQLST(J)= S(K)+Z/SUMXXI(IR)  
WRITE(LIST,380) IDOUT(K),(SSQLST(J),J=1,NODEP)  
8635 CONTINUE  

C

**COMPUTE STANDARD DEVIATION OF REGRESSION COEFFICIENTS**

WRITE(LIST,375)  
IF(.NOT.BZERO) GO TO 959  
DO 910 J=1,NXTERM  
R(J)=0.0  
DO 910 I=1,NXTERM  
CALL LOC(I,J,IR)  
R(J)=R(J)+ZEAN(I)*SUMXXI(IR)  
910 CONTINUE  

XXT=0.0  
DO 920 J=1,NXTERM  
920 XXT=XXT+ZEAN(J)*R(J)  
DO 930 K=1,NODEP  
930 DEVB(1,K)=SQRT(ERRMS(K)*(RNOOB+XXT))  
K=0  
WRITE(LIST,380) K,(DEVB(1,J),J=1,NODEP)  
959 IR=IR  
DO 970 J=1,NXTERM  
IR= IR+J  
DO 970 K=1,NODEP  
DEVB(J,K) =SQRT(ERRMS(K)*SUMXXI(IR))  
960 CONTINUE  
WRITE(LIST,380) IDOUT(J),(DEVB(J,K),K=1,NODEP)  
970 CONTINUE  
RETURN  

C
C FORMATS
1001 FORMAT(4H4-AN0VA OF REGRESSION ON DEPENDENT VARIABLE I5)
1002 FORMAT(1H 79(1H*)/79H SOURCE SUMS OF SQUARES DEG
XRES OF FREEDOM MEAN SQUARES /1H 79(1H*))
1003 FORMAT(17H REGRESSION G20.8, 5X, I10, 5X, G20.8)
1004 FORMAT(17H RESIDUAL G20.8, 5X, I10, 5X, G20.8)
1005 FORMAT(1H 79(1H*))
1006 FORMAT(17H TOTAL G20.8, 5X, I10)
1007 FORMAT(1H 79(1H*))
2001 FORMAT(1X/1X/22H ANOVA OF LACK OF FIT )
2005 FORMAT(17H LACK OF FIT G20.8, 5X, I10, 5X, G20.8)
2006 FORMAT(17H REPLICATION G20.8, 5X, I10, 5X, G20.8)
2008 FORMAT(28H F = MS(LOF)/MS(REPS) = F10.3 )
370 FORMAT((74H1 SUMS OF SQUARES DUE TO EACH VARIABLE IF IT WERE LAST T
XO ENTER REGRESSION )
375 FORMAT(115H2 STANDARD DEVIATION OF REGRESSION COEFFICIENTS (DERIVE
XO FROM DIAGONAL ELEMENTS OF (X TRANSPOSE X)INVERSE MATRIX ))
380 FORMAT(1H I3,9G14.6)
1500 FORMAT(40H R SQUARED = SSQ(REG) / SSQ(TOT) = F8.6,
X 5X, 4HR = F7.6)
1600 FORMAT(34H STANDARD ERROR OF ESTIMATE G14.6)
1700 FORMAT(24H USING POOLING STRATEGY 12;29H THE ERROR MEAN SQUARE =
X G14.7, 26H WITH DEGREES OF FREEDOM = I6)
1750 FORMAT(5X,19HF = MS(REG)/MS(ERR) = F6.2, 5X, 13HCOMPARE TO F(12,1H,13,1
XH))
980 WRITE(LIST,981)
981 FORMAT(41H ZERO RESIDUAL DEGREES OF FREEDOM. STOP. )
STOP
END

LIBFTC RECTK

SUBROUTINE RECT(IRW, JJCOL, IMAX, JMAX, A, B, FMT, II)
DIMENSION A(IMAX, JMAX), FMT(14), XOUT(8)
DOUBLE PRECISION B, DXOUT
DIMENSION B(IMAX, JMAX), DXOUT(8)
COMMON/SMAL/DUM(15), LIST
DATA J8/8/
LOGICAL OUT
OUT = .FALSE.
JTIMES=0
JCOL=JJCOL
5 JNXT=JCOL-J8
IF(JNXT) 10,20,30
10 JP=JCOL
GO TO 40
20 JP=J8
GO TO 40
30 JCOL=JNXT
JP=J8
GO TO 50
40 OUT= .TRUE.
50 DO 10 I=1, IRW
GO TO (55, 75, II)
55 CONTINUE
DO 60 J=1, JP
JJ=JTIMES +J
10 CONTINUE
20 CONTINUE
**FIAFTC PRODIX**

C SUBROUTINE PREDIC

C PURPOSE

1) READ INPUT LEVELS OF INDEPENDENT VARIABLES AND COMPUTE
   A PREDICTED RESPONSE FROM THE ESTIMATED REGRESSION EQUATION.

2) COMPUTE VARIANCE AND STANDARD DEVIATION OF THE PREDICTED
   MEAN VALUE AND A SINGLE FURTHER OBSERVATION.

C SUBROJNTINES NEEDED

C TRANS

C LOC

C REMARKS

VALUES FOR DEPENDENT VARIABLES ARE NOT NECESSARY FOR THE
PREDICTING OF VALUES. HOWEVER, A DUMMY VALUE MAY NEED TO
BE SUPPLIED IF A ZERO (BLANK) INPUT VALUE WILL CAUSE AN
IMPOSSIBLE OPERATIOM TO BE ATTEMPTED DURING THE
TRANSFORMATIONS.

******************************************************************************

C SUBROUTINE PREDIC

COMMON /FRMTS/, FMT(13), FMTTRI(14)
COMM/V/BIG/B(60,9), SUMXY(60,9), SUMXX(1830), SUMXXI(1830)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXXI
COMM/V/MED/B(60,9), SUMX(69), SUMX2(69), SUMY(9), ZEAN(69),
X CON(99), ERRMS(9), IDENT(13), IDOUT(60), NCON(200), NTERM(69),
X NTRAN(100), NXCOD(100), POOLD(9), RCPVAR(9), RMS(9), X(99)
DOUBLE PRECISION BO, SUMX, SUMS, SUMY2, ZEAN
COMMON /SMALL/, BYPASS, BZERO, DELETE, FIRST, IFCHI, IFSSR,
X IFTT, IFMT, INPUT, INPUTS, INTER,
X ISTRAT, JCOL, KUNNO, LENGTH, LIST,
X NERROR, NODEP, NODEB, NOTERM,
X NOVAR, NPDEG, NES, NTRANS, WHERE,
X P, PRECT, REPS, RMT,
X STORI, STORY, STORYX, FTMT, WEIGHT,
X ERRFXO, ECUINUM, IOUT, ICOL

LOGICAL BYPASS, BZERO, DELETE, IFCHI, IFSSR,
X IFST, IFMT, REPS, PREDICT,
X STORY, STORYX, STORYI, FIRST, ERRFXO

60 XOUT(J) = A(I,J)
    WRITE(LIST,FMT) 1,(XOUT(K),K=1,JP)
    GO TO 100
75 DO 40 J = 1, JP
    JJ = JTIMES + J
40    WRITE(LIST,FMT) 1,(DXOUT(J),J=1,JP)
100 CONTINUE
    IF(OUT) RETURN
    WRITE(LIST,110)
110 FORMAT(1H /1H)
    JTIMES = JTIMES + JP
    GO TO 5
END
DOUBLE PRECISION WEIGHT, RWT, TOTWT
DIMENSION YCALC(9), V(60), VARP(9), SEEP(9)
X WAR (9), SEEP (9)
DOUBLE PRECISION XXT, V
EQUIVALENCE (YCALC(1), SUMX(1)), (V(1), SUMXX(150)),
X (VARN(1), SUMXX(71)), (SEEM(1), SUMXX(80)), (VARP(1), SUMXX(89))
X (SEEP(1), SUMXX(98))
EQUIVALENCE (RNODB, RWT)

C******************************************************
C
C IF (NOTERM, EQ, 0) RETURN
WRITE (LIST, 3)
READ (INPUT5, 5) NPRED
C
DO 503 KK = 1, NPRED

105 READ (INPUT5, FMT) (X(I), I = 1, ICOL)
WRITE (LIST, 110) (X(I), I = 1, ICOL)
125 CALL TRANS
DO 133 K = 1, ICOL
I = TERM(K)
X(K) = CON(I)
130 CONTINUE
WRITE (LIST, 135) (X(I), I = 1, NOTERM)

C
C COMPUTE PREDICTED RESPONSE
140 DO 153 K = 1, NODEP
YCALC(K) = B0(K)
IF (.NOT. BZERO) YCALC(K) = 0.0
DO 153 J = 1, NOTERM
YCALC(K) = YCALC(K) + B(J, K) * X(J)
150 CONTINUE

C
C COMPUTE VARIANCE AND STANDARD DEVIATION OF REGRESSION LINE
C AND VARIANCE AND STANDARD DEVIATION OF PREDICTED VALUE
C AT THE POINT XO
C
DO 253 K = 1, NOTERM
V(K) = J.000
DO 253 J = 1, NOTERM
CALL LOC(J, K, IR)
V(K) = V(K) + (X(J) - ZEAN(J)) * SUMXI(IR)
250 CONTINUE

XXT = 0.000
DO 273 K = 1, NOTERM
XXT = XXT + (X(K) - ZEAN(K)) * V(K)
275 CONTINUE

XRNUOB = RNODB
IF (.NOT. BZERO) XRNUOB = 0.0
DO 300 K = 1, NODEP
VAR(K) = ERRMS(K) * (XRNUOB + XXT)
SEE(K) = SQRT (VAR(K))
VARP(K) = ERRMS(K) + VARP(K)
SEEP(K) = SQRT (VARP(K))
300 CONTINUE
WRITE (LIST, 310) (YCALC(K), K = 1, NODEP)
WRITE (LIST, 320) (VARP(K), K = 1, NODEP)
WRITE (LIST, 320) (SEEP(K), K = 1, NODEP)
WRITE (LIST, 320) (VARM(K), K = 1, NODEP)

C
CONTINUE
RETURN
3 FORMAT(54H1 FOR EACH SET OF INDEP VARIABLES THERE IS COMPUTED...
   20H PREDICTED RESPONSE
   29H VARIANCE OF REGRESSION LINE
   34H STANDARD DEVIATION OF REGRESSION
   29H VARIANCE OF PREDICTED VALUE
   39H STANDARD DEVIATION OF PREDICTED VALUE )
5 FORMAT(14)
110 FORMAT(39H INPUT DATA FOR THIS PREDICTED RESPONSE
   9G14.6))
135 FORMAT(56H INDEPENDENT TERMS ACCORDING TO FINAL REGRESSION MODEL
   9G14.6))
310 FORMAT( 55HPREDICTED RESPONSE FOR ABOVE INDEP VARIABLES
   9G14.6))
320 FORMAT(1H 9G14.6)
END

SUBROUTINE SUMUPS
C C PURPOSE
C 1)CALCULATE (X TRANSPOSE XI AND (X TRANSPOSE Y) MATRICES ON
C OBSERVATION AT A TIME.
C 2)COMPUTE TOTAL OF THE WEIGHTS
C ** BOTH CALCULATIONS ARE IN DOUBLE PRECISION
C SUBROUTINES NEEDED
C LOC
C
C********************************************************************
SUBROUTINE SUMUPS
COMMON/BIG/8(60,9),SUMXY(60,9),SUMXX(1830),SUMXI(1830)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXI
COMMON/MED/60(9),SUMX(69),SUMX2(69),SUMY2(69),ZEN(69),
 X CON(99), ERRMS(9), IDENT(13), IDOUT(60), NCON(20), NTERM(69),
 X NTRAN(100), NXCOD(100), POLE(100), REPVAR(9), RESMS(9), X(99)
DOUBLE PRECISION B0, SUMX, SUMX2, SUMY2, ZEN
COMMON/SMALL/ BYPASS, BZERO, DELETE, FIRST, IFCHI, IFSK,
 X IFFT, IFWT, INPUT, INPUT5, INTER,
 X ISTRAT, JCOL, KONNO, LENGTH, LIST,
 X NERROR, NDDEP, NOOB, NOTERM,
 X NOVAR, NPDEG, NRES, NTRANS, NWHERE,
 X P, PREDCT, REPS, RWT,
 X STORY, STORYG, STORYX, TOTWT, WEIGHT,
 X ERRFXD, ECONMY, IOUT, ICOL
LOGICAL ECONMY,
LOGICAL BYPASS, BZERO, DELETE, IFCHI,
XIFSR, IFFT, IFWT, REPS, PREDCT,
XSTORY, STORYX, STORY, FIRST, ERRFXD
DOUBLE PRECISION RWT, TOTWT, WEIGHT
DOUBLE PRECISION DUB1, DUB2
C C********************************************************************
DO 113 I=1,JCOL
   SUMX(I)=SUMX(I)+X(I)*WEIGHT
110 CONTINUE
    IR = 0
    DO 100 K = 1, NOTERM
        DUB1 = X(K)
        DO 90 J = 1, NODEP
            KBAR = J + NOTERM
            DUB2 = X(KBAR)
            SUMXY(K, J) = SUMXY(K, J) + DUB1 * DUB2 * WEIGHT
90    CONTINUE
    DO 50 I = 1, K
40
C
    IR = IR + 1
    DUB2 = X(I)
    SUMXX(IR) = SUMXX(IR) + DUB1 * DUB2 * WEIGHT
50    CONTINUE
100 CONTINUE
    DO 15 J = 1, NODEP
        KBAR = NOTERM + J
        DUB1 = X(KBAR)
        SUMY2(J) = SUMY2(J) + DUB1 * DUB1 * WEIGHT
15    CONTINUE
    TOTWT = TOTWT + WEIGHT
    RETURN
END

$IBFTC BORDXX

SUBROUTINE BORD
  PURPOSE
  TO COMPLETE THE INVERSION OF A SYMMETRIC POSITIVE DEFINITE
  MATRIX A OF ORDER N GIVEN THAT THE UPPER LEFT SUB-
  MATRIX OF ORDER N-1 HAS ALREADY BEEN INVERTED.

SUBROJTINES NEEDED
  LOC

REMARKS
  ONLY THE UPPER TRIANGULAR PART OF A IS STORED AS A
  VECTOR IN THE ORDER A(1,1), A(1,2), A(2,2), A(1,3), . . . ETC
  SUBROJTINE BORD(IORDER,A)

DIMENSION BETA(60), A(1)
DOUBLE PRECISION A, ALPHA, RALPHA, BETA

ALPHA = 0.000
RM1 = IORDER - 1
IF(NM1) 100, 100, 200
100 A(1) = 1.0/A(1)
   GO TO 600
200 M = NM1*(NM1+1)/2
   LEN = M + IORDER
   DO 400 I = 1, NM1

110 CONTINUE
    IR = 0
    DO 100 K = 1, NOTERM
        DUB1 = X(K)
        DO 90 J = 1, NODEP
            KBAR = J + NOTERM
            DUB2 = X(KBAR)
            SUMXY(K, J) = SUMXY(K, J) + DUB1 * DUB2 * WEIGHT
90    CONTINUE
    DO 50 I = 1, K
40
C
    IR = IR + 1
    DUB2 = X(I)
    SUMXX(IR) = SUMXX(IR) + DUB1 * DUB2 * WEIGHT
50    CONTINUE
100 CONTINUE
    DO 15 J = 1, NODEP
        KBAR = NOTERM + J
40
C
        DUB1 = X(KBAR)
        SUMY2(J) = SUMY2(J) + DUB1 * DUB1 * WEIGHT
15    CONTINUE
    TOTWT = TOTWT + WEIGHT
    RETURN
END
BETA(I) = 0.000
MI = M+I
DO 350 J=1,NM1
    CALL LOC(I,J,II)
    MJ = M+J
    BETA(I) = BETA(I) - A(II)*A(MJ)
350 CONTINUE
ALPHA = ALPHA + A(MI)*BETA(I)
400 CONTINUE
C
C    ALPHA = ALPHA + A(LEN)
RALPHA = 1.000/ALPHA
A(LEN) = RALPHA
C
DO 500 I=1,NM1
DO 500 J=1,I
    CALL LOC(I,J,II)
    A(II) = A(II) + BETA(I)*BETA(J)*RALPHA
500 CONTINUE
C
DO 550 J=1,NM1
    MJ = M+J
    A(MJ) = BETA(J)*RALPHA
550 CONTINUE
C
600 CONTINUE
RETURN
END

$IBTC MFIXXX
SUBROUTINE MFIX
C
C*******************************************************************************
COMM /FMRTS/ FMT(13),FMTTRI(14)
COMMON/BIG/B(60,9),SUMXY(60,9),SXXX(1830),SUMXXI(1830)
DOUBLE PRECISION B,SUMXY,SUMXX,SUMXI
COMMON/MED/BO(9),SUMX(69),SUMX2(69),SUMY(9),ZEAN(69),
  NCON(99),ERRMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
  NTRANS(100),NXCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
DOUBLE PRECISION B0,SUMX,SUMX2,SUMY,ZEAN
COMMON/SMALL/ BYPASS,BZERO,DELETE, FIRST, IFCHI, IFSSR,
  IFTT, IFWT, INPUT, INPUT5, INTER,
  IISTRAT, JCOL, KONNO, LENGTH, LIST,
  NERROR, NODEP, NOOB, NOTERM,
  NDEG, NRES, NTRANS, NWHERE,
  P, PREDCT, REPS, RWT,
  STORY, STORYC, STORYX, TOTWT, WEIGHT,
  ERRFXD, ECONMY, IOOT, ICOL
LOGICAL ECONMY
DOUBLE PRECISION RWT,TOTWT,WEIGHT
LOGICAL BYPASS, BZERO, DELETE, IFCHI,
XIFSSR, IFTT, IFWT, REPS, PREDCT,
XSTORY, STORYX, STORYI, FIRST, ERRFXD
C*******************************************************************************
C
IF(ECJNMY) GO TO 500
IF(IZERO) GO TO 500
WRITE (LIST,530)
WRITE (LIST,540) (SUMX(I),I=1,JCOL)
WRITE (LIST,560)
CALL TRIANG(X,SUMXX,NOTERM,8,FMTTRI,2)
WRITE (LIST,565)
CALL RECT(NOTERM,NODEP,60,9,X,SUMXY,FMTTRI,2)
C***************************************************************************
C COMPUTE AND PRINT MEANS. COMPUTE AND PRINT THE (X TRANSPOSE X)
C MATRIX IN TERMS OF DEVIATIONS FROM MEAN. THE DEVIATIONS FORM
C OF (X T X) IS THE VARIANCE-COVARIANCE MATRIX OF THE
C INDEPENDENT VARIABLES.
C
500 CONTINUE
RWT=1.000/TUTWT
DO 570 I=1,JCOL
570 ZEAN(I)=SUMX(I)*RWT
WRITE (LIST,580)
WRITE (LIST,540) (ZEAN(I),I=1,JCOL)
IR = J
DO 600 J=1,NOTERM
IR=IR+J
IF(.NT.BZERO) GO TO 601
SUMX2(I)=SUMXX(IR)-SUMX(J)**2*RWT
GO TO 600
601 SUMX2(I)=SUMXX(IR)
600 CONTINUE
602 CONTINUE
IR=1
DO 620 J=1,NOTERM
DO 618 K=1,NODEP
IF(.NT.BZERO) GO TO 618
KBAR=NOTERM+K
SUMXY(J,K)=SUMXY(J,K)-SUMX(J)*SUMX(KBAR)*RWT
618 CONTINUE
619 DO 620 K=1,J
620 CONTINUE
IF(.NT.BZERO) GO TO 622
DO 6210 J=1,NODEP
K=NOTERM+J
SUMY2(J)=SUMY2(J)-SUMX(K)**2*RWT
6210 CONTINUE
6220 CONTINUE
C***************************************************************************
C FOR SUMS OF INDEP AND DEP VARIABLES
C
622 CONTINUE
$\text{IBFTC LOCXXX}$

```fortran
SUBROUTINE LOC(I,J,IR)
    IX = I
    JX = J
20 IF(IX-JX) 22,24,24
22 IRX = IX + (JX*JX-JX)/2
       GO TO 36
24 IRX = JX + (IX*IX - IX)/2
36 IR = IRX
    RETURN
END
```

$\text{IBFTC XTRANS}$

```fortran
SUBROUTINE TRANS
C***************************************************************************
C
COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
DOUBLE PRECISION B, SUMXY, SUMXX, SUMXXI
COMMON/MED/B0(9),SUMX(69),SUMXZ(69),SUMY2(9),ZEB(69),
   X CIN(99), ERM(9), IDENT(13), IDI(60), NCON(290), NTFRM(69),
   X NTRAN(100), NXCIC(100), POOLED(9), REPVAR(9), RESMS(9), X(99)
DOUBLE PRECISION B0, SUMX, SUMZ, SUMY2, ZEB
COMMON/SML/ BYPASS, BZERO, DELFLTE, FIRST, IFCHI, IFSSR,
   X IFTT, IFWT, INPUT, INPUT5, INTER,
   X ISTRAT, JCOL, KUNQO, LENGTH, LIST,
   X VERRROR, NDEP, ND IB, NOTERM,
   X UVAR, NPDEG, NRRES, NTRANS, NWHER EM,
   X P, PREDCT, REPS, RWT,
   X STORY, STORY, STORYX, TOTWT, WEIGHT,
   X ERRFXD, EONMY, IOUT, ICOL
LOGICAL ECONMY
   X IFCHI, IFSSR, IFTT, IFWT, REPS, PREDCT,
   X STORY, STORY, STORYI, FIRST, FRRFXD
DOUBLE PRECISION RWT, TOTWT, WEIGHT
C***************************************************************************
C THIS SUBROUTINE PERFORMS TRANSFORMATIONS IF THIS OPTION IS
C REQUESTED.
C***************************************************************************
```

70
I

L

28

C K TRANSFORMATION SET NUMBER.
C NCON(2*K-1) CONSTANT NUMBER TO USE.
C NCUN(2*K) DERIVED CONSTANT.
C NTRAN(K) NUMBER OF TRANSFORMATION REQUESTED.
C NXCD(K) VARIABLE NUMBER

80 DO 503 K=1,NTRANS
   I=NCON(2*K-1)
   IF(I).GT.100,110,10
100 CONS=0.0
   GO TO 120
110 CONS=CON(I)
120 I=NXCD(K)
   Y=X(I)
   MTRAN = NTRAN(K)
   IF(MTRAN.LE.0) MTRAN=32
140 DO 150,160,170,180,190,200,210,220,230,240,250,260,270,280,290,
   X300,310,320,330,340,350,360,370,380,390,400,410,420,430,440,
   X 442,450),MTRAN
150 CONS=Y+CONS
   GO TO 460
160 CONS=Y*CONS
   GO TO 460
170 CONS=CONS/Y
   GO TO 460
180 CONS=EXP(Y)
   GO TO 460
190 CONS=Y**CONS
   GO TO 460
200 CONS=ALOG(Y)
   GO TO 460
210 CONS=ALOG10(Y)
   GO TO 460
220 CONS=SIN(Y)
   GO TO 460
230 CONS=COS(Y)
   GO TO 460
240 CONS=SIN(3.14159265*(CONS*Y))
   GO TO 460
250 CONS=COS(3.14159265*(CONS*Y))
   GO TO 460
260 CONS=1.0/Y
   GO TO 460
270 CONS=EXP(CONS/Y)
   GO TO 460
280 CONS=EXP(CONS/(Y*Y))
   GO TO 460
290 CONS=SQRT(Y)
   GO TO 460
300 CONS=1.0/SQRT(Y)
   GO TO 460
310 CONS=CONS**Y
   GO TO 460
320 CONS=1.0**Y
   GO TO 460
330 CONS=SINH(Y)
   GO TO 460
340 CONS=COSH(Y)
   GO TO 460
350 CONS=(1.0-COS(Y))/2.0
   GO TO 460
360 CONS=ATAN(Y)
   GO TO 460
370 CONS=ATAN2(Y,CONS)
   GO TO 460
380 CONS=Y*Y
   GO TO 460
390 CONS=Y*Y*Y
   GO TO 460
400 CONS=ARSIN(SQRT(Y))
   GO TO 460
410 CONS=2.0*3.14159265*Y
   GO TO 460
420 CONS=1.0/(2.0*3.14159265*Y)
   GO TO 460
430 CONS=ERF(Y)
   GO TO 460
440 CONS=ERF2(Y)
   GO TO 460
442 CONS=Y/CONS
   GO TO 460
450 CONS=Y
460 I=NCOV(2*K)
   CON(I)=CONS
   IF(I-53)
      500,500,490
490 XI(I)=DNS
500 CONTINUE
600 RETURN
   END

$IBFCTC OUTPLX

C******************************************************************************
C
C******************************************************************************

SUBROUTINE OUTPLT(PNCH)
COMMON/BIG/B(60,9),SUMXY(60,9),SUMXX(1830),SUMXXI(1830)
   DOUBLE PRECISION B, SUMXY, SUMXX, SUMXXI
   COMMON/ME/MED/B(60,9),SUMX(69),SUM2(69),SUMY2(9),ZEN(69),
   X CON(99),ERMS(9),IDENT(13),IDOUT(60),NCON(200),NTERM(69),
   X NTRAN(100),NCOD(100),POOLED(9),REPVAR(9),RESMS(9),X(99)
   DOUBLE PRECISION BD,SUMX,SUM2, SUMY, ZEN
   COMMON/SMALL/ BYPASS,BZERO,DELETE, FIRST, IFCH1, IFSSR,
   X IFTT, IFWT, INPUT, INPUT5, INTER,
   X ISTRAT, JCOL, KUNNO, LENGTH, LIST,
   X NERROR, NODEP, NOOB, NOTERM,
   X NOVAR, NPDEG, NRES, NTRANS, NWHERE,
   X P, PREDCT, REPS, RWT,
   X STORY1, STORYC, STORYX, TOTWT, WEIGHT,
   X ERF2DYD, ECONMY, IDOUT, ICOL
   LOGICAL ECONMY
   DOUBLE PRECISION RWT,TOTWT,WEIGHT
   LOGICAL BYPASS, BZERO, DELETE, IFCH1,
   X IFSSR, IFTT, IFWT, REPS, PREDCT,
   X STORY1, STORYC, STORYX, FIRST ,ERRFXD
   COMMON/MAX/MAXPLT
C

72
C

INTEGER CELLS, PLUS1

DIMENSION BOUND(45), CELLDI(21), OBS(20,9), RCT(212),
X STDERR(9), VAR(9), YCALC(9), YDIFR(9),
X Z(9)

EQUIVALENCE (VAR, RESMS)

ODATA BOUND(45),67448907
1,1,31839392,67448907
1,1,52439799,84162001
1,1,43072721,67448922
1,1,18001235,36610621
1,1,6356653,46521688
1,1,48377614,6744930
1,1,53411831,13971028
1,1,58945544,76470731
1,1,59323355,12566134
1,1,52440000,67448801
1,1,28154233,64490172

LOGICAL SAVRES

DIMENSION RESPLT(1)

DIMENSION SKUR(9)

EQUIVALENCE (RESPLT, SUMXY)

DIMENSION ICHAR(9), HCHAR(9)

DATA(ICHAR(I)~I=1,9)/*HR DEP, HDVAr
X 96-1 VS P6HR REDICT, 6H VAR

LOGICAL PNCH

JC0L=N0TERM+ NODEP

NUVAR=N0TERM+1

BYPASS= .FALSE.

KOUNT= 0

SAVRES= .FALSE.

ITPLT=NOOB*(NWHERE+2*NODEP)

IF(ITPLT.LE.MAXPLT) SAVRES= .TRUE.

CALL LRLEGN(IDENT(10), 24, 0, 1, 4, 5, 1, 0)

LOGICAL PNCH

IF(NO3B-20) 1109120r 120

110 BYPASS= .TRUE.

GO TO 125

120 CELLS=NOOB/5

CELLS=MNO(CELLS, 20)

I= MOD(CELLS, 2)

IF(1.VE.0) CELLS=CELLS + 1

FCELLS= FLOAT(CELLS)

PLUS1= CELLS + 1

MINUS1= CELLS -1

NDEGC= CELLS -3

IR= CELLS/2-1

IC=IR*(IR-1)/2

IS=IR+2

DO 122 J=1, IR

IC=IC+1

IBC=IS-J

IRC=IS+J

CELLDI(IRC)=BOUND(IRC)

CELLDI(IBC)=BOUND(IBC)

122 CONTINUE

END
CONTINUE

CELLBD(i)=-1.0E+37
CELLBD(PLUS1)=1.0E37
CELLBD(IS)=0.0
DO 124 K=1,NODEP
DO 124 I=1,CELLS
OBS(I,K)=0.0

CONTINUE

C

DO 125 K=1,NODEP
SKEW(K)=0.0
SKUR(K)=0.0
STDERR(K)= SQRT(ERRMS(K))

CONTINUE

WRITE(LIST,135)

C

DO 430 J=1,NOOB
READ(INTER) (X(I),I=1,69),WEIGHT
IF(.NOT.SAVRES) GO TO 141
INOPLT=NWHERE
DO 140 I=1,INOPLT
K=(I-1)*NOOB+J

RESPLT(K)=X(I)

CONTINUE

DO 142 I=1,NOTERM
K= IDOUT(I)
X(I)= X(K)

CONTINUE

KBAR=NWHERE
DO 143 I=1,NODEP
IC= NOTERM+I
KBAR=KBAR+1
X(IC)= X(KBAR)

CONTINUE

DO 160 K=1,NODEP
YCALC(K)= BO(K)
IF(.NOT.BZERO) YCALC(K)= 0.0
KBAR= K+NOTERM
DO 150 I=1,NOTERM

YCALC(K) = YCALC(K) + B(I,K)*X(I)

CONTINUE

ACTDEV= X(KBAR) - YCALC(K)
YDIFR(K)= ACTDEV
Z(K)=ACTDEV/STDERR(K)
A=ACTDEV**3
SKEW(K)=SKEW(K)+A
SKUR(K)=SKUR(K)+A*ACTDEV

CONTINUE

IF(.NOT.SAVRES) GO TO 179
K=INOPLT*NOOB+J
KBAR<K+NOOB*NODEP
DO 175 I=1,NODEP
ITC=(I-1)*NOOB
ISC=K+ITC
IS=KBAR+ITC
RESPLT(ISC)=YCALC(I)
RESPLT(IS)=Z(I)

CONTINUE
CONTINUE
WRITE(LIST, 180) (X(K), K = NUVAR, JCOL)
WRITE(LIST, 190) (YCALC(K), K = 1, NODEP)
WRITE(LIST, 200) (YDIFR(K), K = 1, NODEP)
IF(PNZH) PUNCH 5250, J(YDIFR(K), YCALC(K), K = 1, NODEP)
WRITE(LIST, 210) (Z(K), K = 1, NODEP)
IF(BYPASS) GO TO 410
CONTINUE
DO 250 K = 1, NODEP
DO 230 I = 1, PLUS1
IF(Z(K) - CELLBD(I)) 220, 220, 230
OBS(I-1, K) = OBS(I-1, K) + 1.0
GO TO 250
230 CONTINUE
250 CONTINUE
CONTINUE
C
KOUNT = KOUNT + 1
IF(KOUNT .LT. 10) GO TO 430
WRITE(LIST, 270) IDENT
KOUNT = 0
430 CONTINUE
CONTINUE
C
IF(.NOT. SAVRES) GO TO 439
NC = NWHERE + NODEP
DO 435 IRC = 1, NODEP
IC = (IRC + IRC-1) * NOOB + 1
DO 434 K = 1, NWHERE
IS = (K-1) * NOOB + 1
ICCHAR(5) = IRC
ICCHAR(9) = K
CALL LRNCVT(ICCHAR(5), 1, ICCHAR(5), 1, 6, 0)
CALL LRNCVT(ICCHAR(9), 1, ICCHAR(9), 1, 6, 0)
CALL LRTLGEN(ICCHAR, 54)
CALL LRPLT(RESPLT(IS), RESPLT(IC), NOOB)
434 CONTINUE
IS = (NWHERE + IRC-1) * NOOB + 1
MCHAR(5) = IRC
CALL LRNCVT(MCHAR(5), 1, MCHAR(5), 1, 6, 0)
CALL LRTLGEN(MCHAR, 54)
CALL LRPLT(RESPLT(IS), RESPLT(IC), NOOB)
435 CONTINUE
C
C
IF(BYPASS) RETURN
DO 650 I = 1, NODEP
SKW(I) = SKW(K)**2/(FLOAT(NOOB)**2*ERRMS(K)**3)
SKUR(I) = SKUR(K)/(FLOAT(NOOB)*ERRMS(K)**2)
CHISQ = 0.0
DO 640 I = 1, CELLS
RCT(I) = OBS(I, K)
CHISQ = CHISQ + RCT(I)**2
640 CONTINUE
CHISQ = FCCELLS*CHISQ/FLOAT(NOOB)**2 - FLOAT(NOOB)
WRITE(LIST, 280) NDEGC, CHISQ, SKW(K), SKUR(K)
CALL -LIST(K, RCT, CELLS)
650 CONTINUE
RETURN
135 FORMAT( 51H FOR EACH DEPENDENT TERM AND OBSERVATION IS PRINTED
X /31H OBSERVED RESPONSE (Y OBSERVED)
X /29H CALCULATED RESPONSE (Y CALC)
CRSPLT PROGRAM

CRSPLT accepts a subset of the data used for a NEWRAP problem. It can be used as a preregression analysis program to help formulate model equations to be analyzed with NEWRAP or it may be used as a postregression program by using punched output from NEWRAP to obtain more complex residual plots than direct use of NEWRAP allows.

When used as a preregression program, it will compute an \( X'X \) and \( C \) matrix including all the terms (independent and dependent) if requested. It can also compute eigenvalues and eigenvectors of the submatrix of \( X'X \) corresponding to the independent variables.

When used as a postregression program, the punched output of residuals and predicted values from NEWRAP can be plotted against new functions of the independent variables.

The input is much the same as for NEWRAP. The seven sets of input are as follows:

1. **IDENTIFICATION** (I, IDENT)(12, 13A6): IDENT is Hollerith data used to identify the problem. I indicates the number of additional cards to be read for identification (columns 1 to 78).

2. **PROBLEM SIZE** (NOVAR, NODEP, NOTERM, NOOB)(314, 15)
   - NOVAR Number of input independent variables
   - NODEP Number of input dependent variables
   - NOTERM Number of terms in model equation
   - NOOB Number of observations

3. **TRANSFORMATIONS**: This input is the same as the transformations of NEWRAP except for the upper limit of 150 transformations.

4. **FORMAT** (INPUT, FMT)(12, 13A6): INPUT specifies the unit number the input data is stored on and FMT indicates the format for reading it.
(5) PLOTTING REQUESTS

NOPLTS (I4)
(IXPLT, IYPLT)

One card supplies NOPLTS, the number of plots desired. The following
cards supply pairs of integers indicating which pairs of terms to plot. The for­
mat is 4012 (i.e., 20 plots per card). The first number of the pair (IXPLT)
specifies the sequence number of the term to be used as the abscissa. The
second number (IYPLT) specifies the sequence number of the term to be used
as the ordinate. As an example, the following sequence of transformations and
subsequent plotting requests would cause $X_1^2$ to be plotted against $x_1$ as well
as against $x_3$:

MODEL SIZE 0003
TERMS 616263
TRANSFORMATIONS 0100061 01026162 01026263
CONSTANTS blank card
NOPLTS 0002
(IXPLT, IYPLT) 0102 0302

(6) MATRIX REQUESTS (XTXC, EIGENC)(2L1): If XTXC is F, no matrix calcula­
tions are executed. If it is T, than an $X'X, X'X$ deviation and a correlation
matrix of all the NOTERM + NODEP terms appearing on the TERMS card are
computed. If EIGENC is T, then the eigenvalues and eigenvectors of the sub­
matrix corresponding to the independent terms (the first NOTERM terms) are
calculated.

(7) DATA: Same usage as in NEWRAP.

An illustrative set of input is given, followed by the corresponding sample of output
and a main program listing. The subprograms TRIANG, RECT, and EIGEN are re­
quired and are the same as in NEWRAP.

15 SAMPLE CRNFLT PROBLEM
DATA IS FROM DRAPER AND SMITH
APPLIED REGRESSION ANALYSIS (REFERENCE 4 OF NEWRAP REPORT)
CHAPTER 7

INITIAL MODEL EQU WAS
$Y = \text{CHAMBER PRESSURE}$
$X_1 = \text{TEMPERATURE OF CYCLE}$
$X_2 = \text{VIBRATION LEVEL}$
$X_3 = \text{DROP(SHOCK)}$
$X_4 = \text{STATIC FIRE}$
$Y = b_0 + b_1X_1 + b_2X_2 + b_3X_3 + b_4X_4 + \text{ERR}$
The following terms are being created for residual plots:

<table>
<thead>
<tr>
<th></th>
<th>XI</th>
<th></th>
<th></th>
<th>XIx2</th>
<th>XIx3</th>
<th>XIx4</th>
<th>X2x3</th>
<th>X2x4</th>
<th>X3x4</th>
<th>Y(PRED)</th>
<th>UNIT NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>12</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td></td>
<td></td>
<td>10</td>
<td>11</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Unit NO. 

6   12 24

---

1.2

---

0.5 (8x-F20.2x.4F6.0F6.2E16.8)

---

11.3 6 31 3 4 13 5 13 6 13 7 13 8 13 9 13 13 11 13 12 13
$IRFTC  CASPLX

COMMON/A1/, X(99), CUN(99), SUMX(70), SUMXX(2485), A(70, 7C)
x, XDATA(12000)
COMMON/A2/, XMEAN(70), XSTD(70), SUMX2(7C), NTRANS, NCON(3*0)
x, NTERM(71), NTRAN(15C), NXCOND(15C)
DIMENSION IDENT(13), FMT(13), FMTTR(14), CORR(1), FMTSGL(3)
DATA(FMTTR(1), I=1, 3)/6H(1H 16, 6H, 8D15, 2H6)/
DATA(FMTSGL(1), I=1, 3)/6H(1H 16, 6H, 8D15, 2H6)/
LOGICAL XTXC, EIGENC
EQUIVALENCE (CORR, A)
DATA ICHAR(27), 6H VS /
COMMON IXPLT(4CC), IYPLT(4CC)
DIMENSION ICHAR(3)

C********************************************************************
C
1C READ(5, 110) I, IDENT
WRITE(6, 111) IDENT
DO 100 J=1, 19863
X(J)=0.0
100 CONTINUE

11C IF(I(1) EQ 120, 120, 115)

115 READ(5, 300) FMT
WRITE(6, 171) FMT
I=I-1
GO TO 113

12C READ(5, 112) NOVAR, NUDEP, NTERM, NJOB
WRITE(6, 305) NOVAR, NUDEP, NTERM, NJOB
MTERM=NUTERM+NUDEP
IF(L.GT.12000) GO TO 1000
READ(5, 282) NTRANS, KONNU
IF(NTRANS).LT.255, 255, 22C

22C READ(5, 239)(NTERM(K), K=1, MTERM)
WRITE(6, 235) NTERM(K), K=1, MTERM
READ(5, 230)(NXCOND(I), NTRAN(I), NCON(2*I-1), NCON(2*I), I=1, NTRANS)
WRITE(6, 240) NXCOND(I), NTRAN(I), NCON(2*I-1), NCON(2*I), I=1, NTRANS
IF(KONNU).GT.255, 255, 22C

25C READ(5, 260) (CON(I), I=1, KONNU)
WRITE(6, 262) (CON(I), I=1, KONNU)

255 READ(5, 3000) IUIN, FMT
WRITE(6, 3001) IUIN, FMT
READ(5, 282) NOPLTS
IF(NOPLTS.LT.0) GO TO 32C
IF(NOPLTS.GT.300) GO TO 200C
READ(5, 230) (IXPLT(I), IYPLT(I), I=1, NOPLTS)
WRITE(6, 5000) (IXPLT(I), IYPLT(I), I=1, NOPLTS)

32C CONTINUE
READ(5, 6000) XTXC, EIGENC
NORD=NOVAR+NUDEP

C********************************************************************
KNOOB = 1, 0 /FLOAT(KNOOB)
DO 690 J=1, KNOOB
READ(IUIN, FMT) (XI(I), I=1, NORD)
IF(NTRANS).EQ.45C, 45C, 34C

34C CALL TRANS
DO 430 K=1, MTERM
I= NTERM(K)
X(K)=CON(I)
34C CONTINUE

43C CALL TRANS
DO 430 K=1, MTERM
I= NTERM(K)
X(K)=CON(I)
43C CONTINUE

C********************************************************************
CONTINUE
IF(.NOT.XTXC) GO TO 620
IR=0
DO 610 I=1,MVTERM
SUMX(I)=SUMX(I)+X(I)
GO TO 600
600 CONTINUE
610 CONTINUE
DO 620 I=1,MVTERM
SIX=SUMX(I)+X(I)
620 CONTINUE
IA=J-NOOB
IX=IA+J*NOOB
XDATA(IX)=X(I)
650 CONTINUE
IF(.NOT.XTXC) GO TO 720
WRITE(6,1010)
CALL RECT(NOOG,MVTERM,NOOB,MVTERM,XDATA,FMTSGL)
WRITE(6,1020)
CALL TRIANG(SUMX,MVTERM,8,FMTTRI)
IR=0
DO 700 I=1,MVTERM
IR=IR+1
SUMXX(I)=SUMXX(IR)-SUMX(I)**2*NOOB
XMEAN(I)=SUMX(I)*NOOB
700 CONTINUE
DO 710 J=1,MVTERM
710 CONTINUE
DO 720 K=1,J
SUMXX(IR)=SUMXX(IR)-SUMX(K)*SUMX(J)*NOOB
CORR(IR)=SUMXX(IR)/SQRT(SUMX2(J)*SUMX2(K))
IR=IR+1
720 CONTINUE
WRITE(6,1025)
CALL TRIANG(SUMX,MVTERM,8,FMTTRI)
WRITE(6,1030)
CALL TRIANG(CORR,MVTERM,8,FMTTRI)
IF(.NOT.EIGENC) GO TO 720
CALL EIGEN(SUMX,A,NOTERM,C)
WRITE(6,1040)
J=0
DO 718 I=1,NOTERM
J=J+1
718 SUMXX(I)=SUMXX(J)
WRITE(6,1050)(SUMXX(I),I=1,NOTERM)
WRITE(6,1060)
CALL RECT(NOTERM,NOTERM,NOTERM,NOTERM,A,FMTTRI)
720 CONTINUE
CALL LRLEGN(IDENT,54,C,15,C,0)
CALL LRLEGN(IDENT(1C),24,C,5,C,0)
DO 990 K=1,NOPLTS
ICHAR(K)=IXPLT(K)
ICCHAR(K)=IYPLT(K)
IS1=IXPLT(K)-1*NOOB+1
IS1=IYPLT(K)-1*NOOB+1
CALL LGCNVT(ICHAR(K),I,ICCHAR(K),1,6,C)
CALL LGCNVT(ICHAR(K),I,ICCHAR(K),1,6,C)
CALL LRLEGEN(ICHAR,18)
CALL LRPLTO(XDATA(IS1),XDATA(IS2),NOOB)
990 CONTINUE
GO TO 10
I. SIFITC TRANSX

SUBROUTINE TRANS
C******************************************************************************
C  COMMON/H1/ X(99),CUN(99),SUMX(7C),SUMXX(2485),A(7C,70)
X,XDATA(12000)
  COMMON/H2/XMEAN(70),XSTD(7C),SUMX2(7C),NTRANS,NCON(300),
X,NTERM(70),NTRAN(150),NXCOD(150)
C******************************************************************************
C  THIS SUBROUTINE PERFORMS TRANSFORMATIONS IF THIS OPTION IS
C  REQUESTED.
C
C  K  TRANSFORMATION SET NUMBER.
C  NCON(2*K-1)  CONSTANT NUMBER TO USE.
C  NCON(2*K)  DERIVED CONSTANT.
C  NTRANS(K)  NUMBER OF TRANSFORMATION REQUESTED.
C  NXCOD(K)  VARIABLE NUMBER.
C
AC DO 500 K=1,NTRANS
  I=NCON(2*K-1)
  IF(I).EQ.100,100,110
  100 FORMAT(25H THE REQUIRED NUMBER OF LOCATIONS EXCEEDS THE 12000 AVAIL
XYABLE I8)
  101 FORMAT(6,20X) NOLTS
  STOP
  102 FORMAT(25H MAX NO. OF PLOTS IS 300 I8)
  103 FORMAT(12,13A6)
  104 FORMAT(1H1,13A6)
  105 FORMAT(314.15)
  400 FORMAT(2L1)
  500 FORMAT(1HK/(15(1X,12,1X,12,2X)))
  301 FORMAT(13A6,A2)
  302 FORMAT(1H,13A6,A2)
  303 FORMAT(1H,316)
  282 FORMAT(2I4)
  230 FORMAT(4012)
  235 FORMAT(11H TERMS ARE / (1H 3014))
  246 FORMAT(25H THE TRANSFORMATIONS ARE / (1H 5(4I4,5X)))
  26F FORMAT(5E15.7)
  262 FORMAT(19H THE CONSTANTS ARE / (1H, 8G15.7))
  101I FORMAT(16H1THE DATA MATRIX )
  102P FORMAT(26H2THE X TRANSPOSE X MATRIX )
  102T FORMAT(32H X TRANSPOSE X DEVIATIONS MATRIX )
  103F FORMAT(23H2THE CORELATION MATRIX )
  104F FORMAT(44H2FOLLOWING ARE EIGENVALUES OF X TRANS X MATRIX)
  1050 FORMAT(1H, 8G15.7)
  106F FORMAT(53H2EIGENVECTORS BY COLUMNS IN SAME ORDER AS EIGENVALUES)
  300G FORMAT(12,13A6)
  300J FORMAT(1H,16,2X,13A6)
  FND

SIFITC TRANSX
CNS = 'A
GO TO 120

110 CNS = CNS(I)
120 I = N(XO)D(I)
Y = X(I)
MTRAN = NTRAN(K)
IF (MTRAN .LE. 1) MTRAN = 32
140 GO TO (150), (160), (170), (180), (190), (200), (210), (220), (230), (240), (250), (260), (270), (280), (290), (300), (310), (320), (330), (340), (350), (360), (370), (380), (390), (400), (410), (420), (430), (440),
X = 442.490, MTRAN
150 CNS = Y * CNS
GO TO 460
160 CNS = Y * CNS
GO TO 460
170 CNS = CNS / Y
GO TO 460
180 CNS = FXP(Y)
GO TO 460
190 CNS = Y ** CNS
GO TO 460
200 CNS = ALN(Y)
GO TO 460
210 CNS = ALGN(Y)
GO TO 460
220 CNS = SIN(Y)
GO TO 460
230 CNS = COS(Y)
GO TO 460
240 CNS = SIN((3.14159265 * (CNS * Y))
GO TO 460
250 CNS = COS((3.14159265 * (CNS * Y))
GO TO 460
260 CNS = 1.0 / Y
GO TO 460
270 CNS = EXP(CNS / Y)
GO TO 460
280 CNS = EXP(CNS / (Y * Y))
GO TO 460
290 CNS = SQRT(Y)
GO TO 460
300 CNS = 1.0 / SQRT(Y)
GO TO 460
310 CNS = CNS ** Y
GO TO 460
320 CNS = 1.0 ** Y
GO TO 460
330 CNS = SINH(Y)
GO TO 460
340 CNS = COSH(Y)
GO TO 460
350 CNS = (1.0 - COS(Y)) / 2.0
GO TO 460
360 CNS = ATAN(Y)
GO TO 460
370 CNS = ATAN2(Y / CNS)
GO TO 460
380 CNS = Y ** Y
GO TO 460
390 CNS = Y ** Y
GO TO 460
400 CNS = 1.0 * SIN(SQRT(Y))
GO TO 460
SAMPLE CASES PLT PROBLEM
DATA IS FROM DRAPER AND SMITH
APPLIED REGRESSION ANALYSIS (REFERENCE 4 OF NEWRAP REPORT)
CHAPTER 7
INITIAL MODEL EU was
Y = CHAMBER PRESSURE
X1 = TEMPERATURE OF CYCLE
X2 = VIBRATION LEVEL
X3 = UNFIRED SHOCK
X4 = STATIC FIRE
Y = X1 + X2 + X3 + X4 + ERR
THE FOLLOWING TERMS ARE BEING CREATED FOR RESIDUAL PLOTS
1 X1 2 X2 3 X3 4 X4
5 X1X2 6 X1X3 7 X1X4 8 X2X3
9 X2X4 10 X3X4 11 YIPRED 12 UNIT NO.
13 RESIDUALS
1 2 3 4 5 6 7 8 9 10 11 12
24 TERMS ARE
61 62 63 64 65 66 67 68 69 70 71 72 73
THE TRANSFORMATIONS ARE
1 C 0 72 2 C 0 61 3 C 0 62 4 C 0 63 5 C 0 64
6 2 63 68 2 64 65 3 2 64 70
5 1 0.4Z.F2.0.2X.466.0.70X.2E16a81
1 13 2 13 3 13 4 13 5 13 6 13 7 13 8 13 9 13 10 13 11 13 12 13
THF DATA MATRIX
1 -75.66000 0 -65.00000 0 0 0 0
2 175.08000 0 150.00000 0 0 0 0
3 0 0 -65.00000 0 0 0 0
4 0 0 150.00000 0 0 0 0
5 0 0 150.00000 0 0 0 0
6 -75.66000 -75.06000 0 0 0 0 0
7 175.08000 175.00000 0 0 0 0 0
8 -75.66000 -75.06000 -65.00000 0 0 0 0
9 175.08000 175.00000 -65.00000 0 0 0 0
10 -65.00000 -65.00000 0 0 0 0 0
11 0 165.00000 165.00000 0 0 0 0
12 -75.06000 -65.00000 0 0 0 0 0
13 0 165.00000 165.00000 0 0 0 0
14 0 0 -65.00000 0 0 0 0
15 0 0 150.00000 0 0 0 0
16 0 0 150.00000 0 0 0 0
17 0 0 -65.00000 0 0 0 0
18 0 0 -65.00000 0 0 0 0
19 0 0 175.00000 0 0 0 0
20 0 0 -65.00000 0 0 0 0
21 0 0 175.00000 0 0 0 0
22 0 0 -65.00000 0 0 0 0
23 0 0 175.00000 0 0 0 0
24 0 0 175.00000 0 0 0 0
1 4.975.000 0 5.650.047 14.00000 0 3.655.047
2 2.250.000 0 27.063.32 2.00000 0 2.753.32
3 0 0 27.063.32 7.00000 0 2.373.81
4 0 0 5.650.047 8.00000 0 4.644.81
5 0 0 27.063.32 9.00000 0 5.836.61
6 -11.250.00 0 12.000.00 25.000.00 0 15.766.64
7 -13.137.00 0 11.000.00 7.000.00 0 11.137.00
8 -11.250.00 0 11.000.00 25.000.00 0 15.766.64
9 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
3 5 8 2 13 17 1 18 24 22 21 23 19 12 16 14 20 23
1 4.975.000 0 5.650.047 14.00000 0 3.655.047
2 2.250.000 0 27.063.32 2.00000 0 2.753.32
3 0 0 27.063.32 7.00000 0 2.373.81
4 0 0 5.650.047 8.00000 0 4.644.81
5 0 0 27.063.32 9.00000 0 5.836.61
6 -11.250.00 0 12.000.00 25.000.00 0 15.766.64
7 -13.137.00 0 11.000.00 7.000.00 0 11.137.00
8 -11.250.00 0 11.000.00 25.000.00 0 15.766.64
9 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
10 8 7 9 12 15 24 13 14 11 16 22 21 19 18 20 23 17
1 4.975.000 0 5.650.047 14.00000 0 3.655.047
2 2.250.000 0 27.063.32 2.00000 0 2.753.32
3 0 0 27.063.32 7.00000 0 2.373.81
4 0 0 5.650.047 8.00000 0 4.644.81
5 0 0 27.063.32 9.00000 0 5.836.61
6 -11.250.00 0 12.000.00 25.000.00 0 15.766.64
7 -13.137.00 0 11.000.00 7.000.00 0 11.137.00
8 -11.250.00 0 11.000.00 25.000.00 0 15.766.64
9 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24

The X TRANSPOSE X MATRIX

| 1 | 140750.0 | 2900C.C | 0 |
| 2 | 33730.0 | 1350C.C | 1870C | 0 |
| 3 | -14172.0 | -82250.0 | 5050.0 | 0 |
| 4 | 3931250.0 | -230450E+C7 | 0.21035E+07 | 5211250.0 |
| 5 | 951375.0 | 0.241555E+07 | 6445250.0 | 0.453525E+08 |
| 6 | -593475.0 | -5211250.0 | 6445250.0 | -0.38255E+07 |
| 7 | -0.293755E+07 | -0.315665E+07 | -0.315665E+07 | -0.315665E+07 |
| 8 | -0.293755E+07 | -0.315665E+07 | -0.315665E+07 | -0.315665E+07 |
| 9 | -0.293755E+07 | -0.315665E+07 | -0.315665E+07 | -0.315665E+07 |
| 10 | -0.293755E+07 | -0.315665E+07 | -0.315665E+07 | -0.315665E+07 |
| 11 | -0.293755E+07 | -0.315665E+07 | -0.315665E+07 | -0.315665E+07 |
| 12 | -0.293755E+07 | -0.315665E+07 | -0.315665E+07 | -0.315665E+07 |

The CORRELATION MATRIX

| 1 | 1493000.0 |
| 2 | 3958753.0 |
| 3 | 503990.0 |
| 4 | 829750.0 |
| 5 | 11070750.0 |
| 6 | 0.217595E+07 |
| 7 | 0.247955E+07 |
| 8 | 0.247955E+07 |
| 9 | 0.247955E+07 |
| 10 | 0.247955E+07 |
| 11 | 0.247955E+07 |
| 12 | 0.247955E+07 |

The X TRANSPOSE X DEVIATIONS MATRIX

| 1 | 109000.0 |
| 2 | 67950.0 |
| 3 | 26333.3 |
| 4 | 13737.0 |
| 5 | 4650.0 |
| 6 | 31355.0 |
| 7 | 0.593933E+08 |
| 8 | 0.894533E+08 |
| 9 | 0.794533E+08 |
| 10 | 0.494533E+08 |
| 11 | 0.794533E+08 |
| 12 | 0.794533E+08 |

The X TRANSPOSE X DEVIATIONS MATRIX

| 1 | 109000.0 |
| 2 | 67950.0 |
| 3 | 26333.3 |
| 4 | 13737.0 |
| 5 | 4650.0 |
| 6 | 31355.0 |
| 7 | 0.593933E+08 |
| 8 | 0.894533E+08 |
| 9 | 0.794533E+08 |
| 10 | 0.494533E+08 |
| 11 | 0.794533E+08 |
| 12 | 0.794533E+08 |

The CORRELATION MATRIX

| 1 | 109000.0 |
| 2 | 67950.0 |
| 3 | 26333.3 |
| 4 | 13737.0 |
| 5 | 4650.0 |
| 6 | 31355.0 |
| 7 | 0.593933E+08 |
| 8 | 0.894533E+08 |
| 9 | 0.794533E+08 |
| 10 | 0.494533E+08 |
| 11 | 0.794533E+08 |
| 12 | 0.794533E+08 |
CREDUCE PROGRAM

In optimum-seeking experimentation involving many independent variables, quadratic response surfaces are often used. A development of the most important aspect of the design and analysis of response surface experiments can be found in Davies (ref. 9) and Box and Hunter (ref. 10). A discussion of the interpretation of a quadratic surface fitted to a large experiment is given in reference 11.

The general form of a quadratic surface is given by

\[ y = b_0 + b'X + X'BX + \epsilon \]

\[ = b_0 + (b_1, b_2, \ldots, b_p) \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} \]

\[ + (x_1, \ldots, x_p) \begin{pmatrix} b_{11} & \frac{1}{2} b_{12} & \cdots & \frac{1}{2} b_{1p} \\ \frac{1}{2} b_{12} & b_{22} & \cdots & \frac{1}{2} b_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & b_{pp} \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_p \end{pmatrix} + \epsilon \]  

(26)

The analysis of such an equation is simplified by two calculations: (1) the calculation of the stationary point of the surface and (2) a transformation of axes to new independent variables which changes the \( B \) matrix to a diagonal matrix.

The stationary point of the surface is the solution \( X_s \) to

\[ \frac{\partial y}{\partial x_i} = 0 \quad i = 1, p \]

The transformation of the axes is given by the computation of the orthogonal matrix \( P \) which reduces \( B \) to a diagonal matrix; that is,
The $\lambda_i$ are the eigenvalues of $B$. The new variables are given by

$$Z = P'X$$

where $P$ is the matrix whose columns are the eigenvectors of $B$. If two successive transformations of the variables are made as follows:

$$W = X - X_s$$

$$Z = P'W$$

then equation (26) becomes

$$y - y_s = \lambda_1 Z_1^2 + \lambda_2 Z_2^2 + \ldots + \lambda_p Z_p^2$$

(27)

where

$$y_s = b_0 + b'X_s + X'_sBX_s$$

(28)

From examination of equation (27), some general conclusions can be drawn concerning the attainment of a maximized response. For example, consider just two of the possible results.

(1) Suppose all the $\lambda_i \leq 0$ and $X_s$ is near or in the region of $X$ at which the experiments were performed. Then clearly any deviation of $Z$ from $Z = 0$ will decrease the response. Thus $Z = 0$ (or equivalently $X = X_s$) is a maximum and is the combination of independent variables the experimenter seeks.

(2) Some $\lambda_i < 0$ and some $\lambda_i > 0$, and $X_s$ is close to the region of experimentation. Thus $X_s$ represents what is sometimes called a saddlepoint. Moving in some directions will cause a decrease in $y$ and moving in other directions will cause an increase in $y$. Thus the experimenter could move from $X_s$ in the direction that corresponds to the direction of the $Z$ which has the largest positive coefficient in equation (27). This will increase $y$ most rapidly from the value of $y_s$. 
The INPUT (Sample input is shown in fig. 8) is as follows:

1. Identification: One card, all 80 columns.
2. The number of independent variations JFAC. (14) (JFAC must be less than or equal to 15)
3. Variable Format. One card, all 80 columns.
4. $b_0$ (according to FORMAT in item (3)).
5. $b_1, \ldots, b_{JFAC}$ (according to format in item (3)).
6. $b_{1,1}$

\[
\begin{align*}
&b_{1,2} & b_{2,2} \\
&b_{1,3} & b_{2,3} & b_{3,3} \\
&\vdots & \ddots & \ddots \\
&b_{1,JFAC} & b_{2,JFAC} & \cdots & b_{JFAC,JFAC}
\end{align*}
\]
One READ statement for each line, according to the format in item (3). As an example, consider the following estimated equation:

\[ y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + b_4 x_4 + b_{11} x_1^2 + b_{12} x_1 x_2 + b_{22} x_2^2 + b_{13} x_1 x_3 + b_{23} x_2 x_3 + b_{33} x_3^2 + b_{14} x_1 x_4 + b_{24} x_2 x_4 + b_{34} x_3 x_4 + b_{44} x_4^2 + b_{15} x_1 x_5 + b_{25} x_2 x_5 + b_{35} x_3 x_5 + b_{45} x_4 x_5 + b_{55} x_5^2 \]

\[ y = 147.2686 - 8.989120 x_1 - 6.817975 x_2 - 14.60964 x_3 + 9.248688 x_4 + 14.19698 x_5 + 1.805520 x_1^2 - 2.126719 x_1 x_2 - 1.475730 x_2^2 - 7.314218 x_1 x_3 - 16.10219 x_2 x_3 + 2.274270 x_3^2 + 1.310780 x_1 x_4 - 2.477188 x_2 x_4 - 1.289688 x_3 x_4 + 2.236770 x_4^2 + 2.664464 x_1 x_5 + 1.827434 x_2 x_5 + 2.389934 x_3 x_5 + 6.014934 x_4 x_5 - 12.58198 x_5^2 \]

A canonical reduction of this is given in the sample output. The listing of the main program is supplied. The subprograms TRIANG, DGELG, EIGEN, and RECT are required. TRIANG and RECT are as in NEWRAP. DGELG and EIGEN are the double precision general linear equation and eigenvalue routines from reference 12.
READ(5,1001) FMT
READ(5,FMT) BZERO
WRITE(6,1005) BZERO
READ(5,FMT) (BL(I),I=1,JFAC)
IE=0
DO 5 I=1,JFAC
IE=IE+1
IS=IE-I+1
READ(5,FMT) (BS(K),K=IS,IE)
DO 4 IS=IE,IE
BSAVE(K)=BS(K)
4 SBS(K)=SNGL(BS(K))
BSAVE(I)=BL(I)
SBL(I)=SNGL(BL(I))
BL(I)=-BL(I)
5 CONTINUE
LENGTH = JFAC*(JFAC+1)/2
WRITE(5,1006)
WRITE(5,1007)(SBL(I),I=1,JFAC)
WRITE(5,1008)
CALL TRIANG(SBS,JFAC,8,FMTTRI)
I=1
XIN(I)= BS(I)*2.000
DO 50 I=2,JFAC
II= I-1
IIK=I
IIJ = JFAC*II
DO 50 IJ=IIJ+1
BSAVE(IJ)=0.5000*BS(IJ)
XIN(IJK)=BS(IJ)
IIK=IIK+JFAC
IIJ= IIJ+1
XIN(IJK)= BS(IJ)
50 CONTINUE
IIJ=IIJ+1
IJ= JFAC*II
XIN(IJ)=2.0000*BS(IJ)
5 CONTINUE
EPS=1.0E-10
CALL GELG(BL,XIN,JFAC,1,EPS,IER)
IF(IER.NE.0) WRITE(6,1009) IER
WRITE(5,1010)
WRITE(5,1007)(BL(I),I=1,JFAC)
I=0
YS= BZERO
DO 153 I=1,JFAC
YS=YS+BL(I)*BSAVE(I)
DO 140 J=1,I
140 YS= YS+BL(I)*BL(J)*BSAVE(IJ)
DO 160 J=1,JFAC
160 SBS(IJ)=SNGL(BSAVE(IJ))
CALL EIGEN(BSAVE,XIN,JFAC,0)
I=0
DO 203 I=1,JFAC
I=I+1
203 BL(I)= BSAVE(IJ)
WRITE(5,1012)
WRITE(5,1007)(BL(I),I=1,JFAC)
WRITE(5,1013)
JJ=JFAC*JFAC
DO 210 J=1,JFAC
210 WRITE(5,1011) YS
C
I=0
DO 163 L=1,LENGTH
163 I=I+1
160 SBS(IJ)=SNGL(BSAVE(IJ))
CALL EIGEN(BSAVE,XIN,JFAC,0)
I=0
DO 203 I=1,JFAC
I=I+1
203 BL(I)= BSAVE(IJ)
WRITE(5,1012)
WRITE(5,1007)(BL(I),I=1,JFAC)
WRITE(5,1013)
JJ=JFAC*JFAC
DO 210 J=1,JFAC
210
SAMPLE CANONICAL REDUCTION PROBLEM

THE COEFFS ARE BZERO 147.268600

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ROW 1</td>
<td>1.805520</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROW 2</td>
<td>-2.12619</td>
<td>-1.475730</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROW 3</td>
<td>-7.14218</td>
<td>-16.10219</td>
<td>2.274270</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROW 4</td>
<td>1.30780</td>
<td>-2.477188</td>
<td>-1.289688</td>
<td>2.236770</td>
<td></td>
</tr>
<tr>
<td>ROW 5</td>
<td>2.664464</td>
<td>1.827434</td>
<td>2.389934</td>
<td>6.014934</td>
<td>-12.58198</td>
</tr>
</tbody>
</table>

SECOND ORDER

| ROW 1  | -0.300108   | 0.519711    | 0.750815     | 0.248709   | -0.138316   |
| ROW 2  | -0.749544   | -0.275150   | -0.293631    | 0.690293   | -0.244071   |
| ROW 3  | 0.779700    | 0.809764E-02| 0.804860E-01| 0.577909   | -0.227638   |
| ROW 4  | -0.194699E-02| 0.790049  | -0.582318    | -0.228627E-02| -0.191618  |
| ROW 5  | 0.105815E-02| 0.173062   | -0.669725E-01| 0.364044  | 0.912707    |

THE STATIONARY POINT IS 3.05811526 -1.96462563 0.12068067 -3.89057544 -0.93711068E-01

THE VALUE OF YS IS 120.589275

EIGENVALUES

EIGENVECTORS

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, January 25, 1972, 132-80.
APPENDIX - BORROWED Routines

Some of the routines used in the programs were taken from the literature. Both INVXTX and TRIANG are by Webb and Galley (ref. 13), and EIGEN and HIST are from the IBM programmer's manual (ref. 12), as are DGELG and the double precision version of EIGEN.

Listing of INVXTX and TRIANG are given here, as follows:

```fortran
SUBROUTINE TRIANG(A,B,NN,NKOL,FORMAT,II)
  DIMENSION FORMAT(II)
  DIMENSION A(1),B(1)
  DOUBLE PRECISION B
  1 FORMAT (1H1)
  3 FORMAT(1H1/H1/H1 )
  COMMON/SMALL/DUM(15),LIST
  N = NN
  NCOL = NKOL
  KLUMPS = N/NCOL
  KEEPT = 0
  K1 = 1
  K2 = NCOL - 1
  K3 = NCOL
  IF (KLUMPS .EQ. 0) GO TO 120
  DO 90 KLUMP=1,KLUMPS
    ITR1 = KEEPT
    I = -1
    ILO = (KLUMP-1)*NCOL + ITR1 + 1
    DO 30 K=K1,K2
      I = I + 1
      ITR1 = ITR1 + K - 1
      ILO = ILO + K - 1
      IHI = ILO + 1
      GO TO (26,28),11
    26 WRITE(LIST,FORMAT) K,(A(J),J=ILO,IHI)
      GO TO 30
    28 WRITE(LIST,FORMAT) K,(B(J),J=ILO,IHI)
    30 CONTINUE
    KEEPT = ITR1 + K2
    DO 60 K=K3,N
      ITR1 = ITR1 + K - 1
      ILO = ILO + K - 1
      IHI = ILO + NCOL - 1
      GO TO (56,58),II
    56 WRITE(LIST,FORMAT) K,(A(J),J=ILO,IHI)
      GO TO 60
    58 WRITE(LIST,FORMAT) K,(B(J),J=ILO,IHI)
    60 CONTINUE
    K1 = K1 + NCOL
    K2 = K2 + NCOL
    K3 = K3 + NCOL
  90 WRITE(LIST,3)
```

C

91
120  ITR1 = KEEPR  
IF (K1 .GT. N) GO TO 180  
I = -1  
ILO = KLUMPS*NCOL + ITR1 + 1  
DO 150  K=K1,N  
I = I + 1  
ITR1 = ITR1 + K - 1  
ILO = ILO + K - 1  
IHI = ILO + I  
GO TO (146,148,11  
146 WRITE(LIST,FORMAT) K,(A(J),J=ILO,IHI)  
GO TO 150  
148 WRITE(LIST,FORMAT) K,(B(J),J=ILO,IHI)  
150 CONTINUE  
180 RETURN  
END

SIBFTC INVXXX

SUBROJTINE INVXTX(A, NN, D, FACT)  

C ASSUMES THE MATRIX A IS SYMMETRIC AND POSITIVE DEFINITE, AND ONLY  
C THE UPPER TRIANGLE IS STORED AS A ONE-DIMENSIONAL ARRAY IN THE  
C ORDER A(1,1), A(1,2), A(2,2), A(1,3), A(2,3), A(3,3), ..., A(N,N).  
C NN IS THE ORDER N OF THE INPUT MATRIX A.  
C D IS ON EXIT) THE DETERMINANT OF A, DIVIDED BY FACTOR**NN.  
C
DIMENSION A(1)  
DOUBLE PRECISION A,PV,F  
N = NN  
ITR1 = 0  
DO 145 K=1,N  
C ITR1 = ITR1+K-1  
KPI = K+1  
KM1 = K-1  
KK = ITR1+K  
PV = 1.0D0/A(KK)  
C ITR2 = 0  
IF (K-1) 150,80,50  
C REDUCE TOP PART OF TRIANGLE, LEFT OF PIVOTAL COLUMN  
50 DO 60 J=1,KM1  
ITR2 = ITR2+J-1  
KJ = ITR2+J  
F = A(KJ)*PV  
DO 60 I=1,J  
IJ = ITR2+I  
IK = ITR1 + I  
60 A(IJ) = A(IJ) + A(IK)*F  
C IF (K-N) 70,120,150  
C REDUCE REST OF TRIANGLE, RIGHT OF PIVOTAL COLUMN  
70 ITR2 = ITR1
DO 110 J=KPl,N
   ITR3 = ITR1
   ITR2 = ITR2+J-1
   KJ = ITR2+K
   F = A(KJ)*PV
   DO 100 I=1,J
   IF (I-K) 90,100,95
   90 IJ = ITR2+I
       IK = ITR1 + I
       A(IJ) = A(IJ) - A(IK)*F
       GO TO 100
   95 IJ = ITR2 + I
   ITR3 = ITR3 + I - 1
   IK = ITR3 + K
   A(IJ) = A(IJ) - A(IK)*F
100 CONTINUE
110 CONTINUE
C
C  DIVIDE PIVOTAL ROW-COLUMN BY PIVOT, INCLUDING APPROPRIATE SIGNS
120 ITR2 = ITR1
   DO 140 I=1,N
      IF (I-K) 125,130,135
   125 IK = ITR1+I
       A(IK) = -A(IK)*PV
       GO TO 140
C
C  (REPLACE PIVOT BY RECIPROCAL)
130 A(KK) = PV
   GO TO 140
135 ITR2 = ITR2+I-1
   KI = ITR2+K
   A(KI) = A(KI)*PV
140 CONTINUE
C
145 CONTINUE
C
150 RETURN
END
REFERENCES


94 NASA-Langley, 1972 — 8 E-6718
"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— National Aeronautics and Space Act of 1958

**NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS**

**TECHNICAL REPORTS:** Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

**TECHNICAL NOTES:** Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

**TECHNICAL MEMORANDUMS:** Information receiving limited distribution because of preliminary data, security classification, or other reasons.

**CONTRACTOR REPORTS:** Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

**TECHNICAL TRANSLATIONS:** Information published in a foreign language considered to merit NASA distribution in English.

**SPECIAL PUBLICATIONS:** Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

**TECHNOLOGY UTILIZATION PUBLICATIONS:** Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

*Details on the availability of these publications may be obtained from:*

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

Washington, D.C. 20546