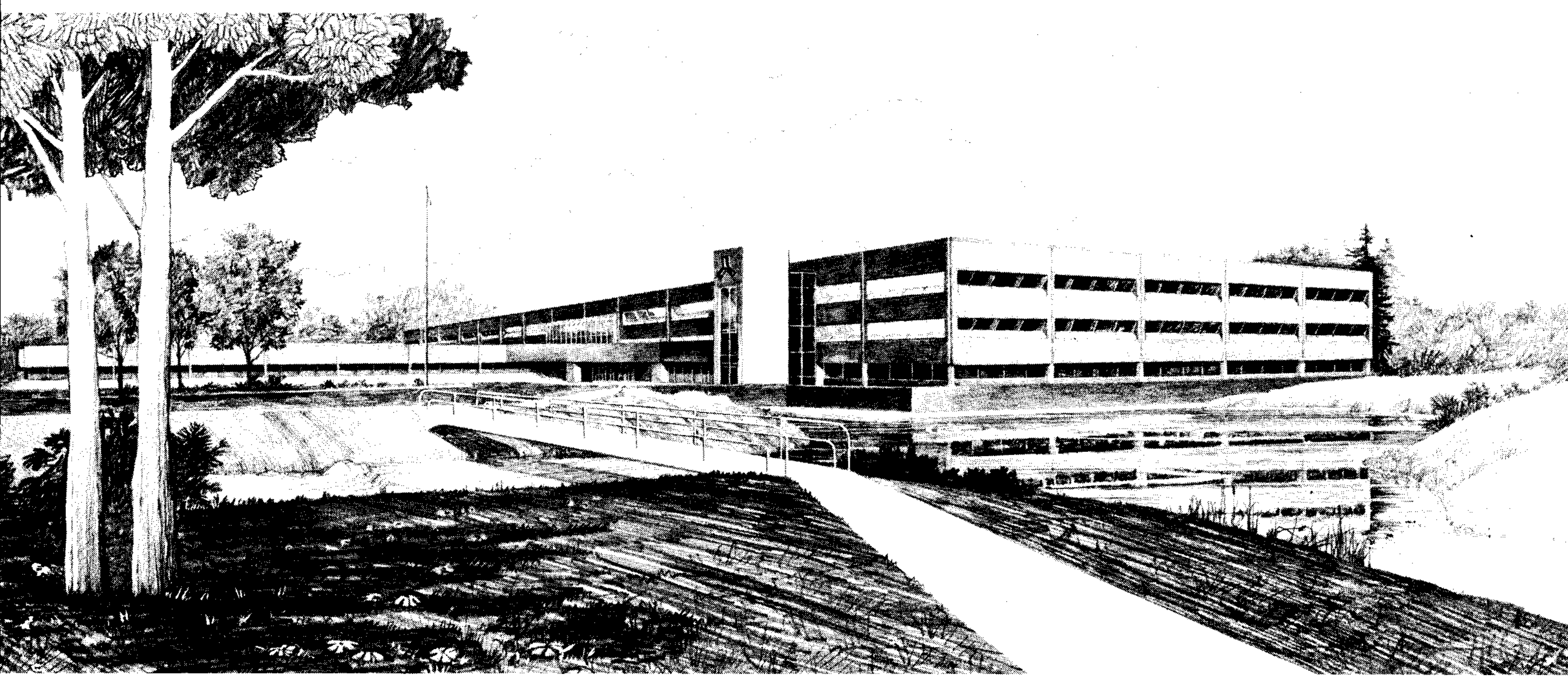
EGG-EA-5455
May 1981

USER'S GUIDE TO ANYOLS:
A FLEXIBLE STEPWISE REGRESSION PROGRAM

MASTER

C. L. Atwood

U.S. Department of Energy
Idaho Operations Office • Idaho National Engineering Laboratory



This is an informal report intended for use as a preliminary or working document

Prepared for the
U.S. Department of Energy
Idaho Operations Office
Under DOE Contract No. DE-AC07-76ID01570

EG&G Idaho

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

EGG-EA-5455

USER'S GUIDE TO ANYOLS:

A FLEXIBLE STEPWISE REGRESSION PROGRAM

C. L. Atwood

Published May 1981

EG&G Idaho, Inc.
Idaho Falls, Idaho 83415

Prepared for the
U.S. Department of Energy
Idaho Operations Office
Under DOE Contract No. DE-AC07-76ID01570

CONTENTS

USER'S GUIDE TO ANYOLS: A FLEXIBLE STEPWISE REGRESSION PROGRAM

1. INTRODUCTION

1.1 Summary

ANYOLS is a stepwise regression program. It fits data using ordinary or weighted least squares. It selects the variables to put into the model in a stepwise way, based on F, s^2, MSE (based on C_p), PRESS, or a user-written subroutine. The user can define forcing priorities to influence the order of entering variables. Instead of stepwise selection, the program can also try all possible combinations of any desired subset of the variables. Automatic output for the final model in a stepwise search includes plots of the residuals, studentized residuals, and leverages; if the model is not too large, the output also includes partial regression and partial leverage plots. A data set may be re-used so that several selection criteria can be tried. Flexibility is increased by allowing the substitution of user-written FORTRAN subroutines for several default subroutines. The program is written in CDC FORTRAN 4.

1.2 If You Have Never Used ANYOLS

If you are working on the INEL Cyber, and if you want to do a least squares fit of some data using the constant term as one of the terms in the fitting equation, the following job will do it. Your output will consist of a printout plus some plots on microfilm.

```
Job card
Account Card
ATTACH,ANYOLS,ID=CLA.
ANYOLS,ADD=0.
*EOR
 $SETUP NX=<your number of predictor variables>,$
```

Data deck. Each line consists of values of the predictor variables, followed by the corresponding value of the response variable. Any line may be continued onto the next line. Free format.

*EOR
Names of the variables, one per line. Format is (A10).

Use *EOR when working from a terminal. Replace this by 7/8/9 in column 1 if using punch cards. The word $SETUP must be in columns 2 through 7, and the final $ is necessary. Using a comma before the final $ prevents the printing of a nuisance message.

For example, suppose that I wanted to fit the equation

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2$$

based on the five data cases

X	Y
4	31
-2	12
3.1	19
7	82
4	28.5

Suppose that X corresponds to length and Y to temperature. Then, the job would consist of

```
CLA,P1,T37.
ACCOUNT,E230,DONTUDARE,TM4.
ATTACH,ANYOLS,ID=CLA.
ANYOLS,ADD=0.
*EOR
 $SETUP NX=2,$
 4      16      31
-2       4      12
 3.1     9.61   19
 7      49      82
 4      16      28.5
```

```
*EOR
CONSTANT
LENGTH
LENGTH SQ
TEMP
```

ANYOLS allows much more flexibility than such a simple example illustrates. For example, you may: enter a title for the problem; fit without the constant term; make the program calculate some of the data (e.g., X^2) from other parts of the data (e.g., X); fit several responses simultaneously; select the variables to be used in the fit according to forcing priorities; use various criteria for selecting the variables to enter; use weighted least squares; build your own output routine; etc. For this, you must read more of this manual.

2. TERMINOLOGY

ANYOLS allows many ways to select a model and many kinds of diagnostic plots. The terminology is defined here. It may be best to skip this section on the first reading and to refer back to it as necessary.

2.1 Model

Suppose some numbers X_{ij} and Y_i have been observed, and they are put into the following equation:

$$\begin{pmatrix} Y_1 \\ \vdots \\ Y_{NO} \end{pmatrix} = \begin{pmatrix} X_{1,1} & \cdots & X_{1,NX} \\ \vdots & & \vdots \\ X_{NO,1} & \cdots & X_{NO,NX} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_{NX} \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ e_{NO} \end{pmatrix} .$$

The X_{ij}'s are referred to as predictor or independent variables, and the Y_i's as response or dependent variables. The β_j's are unknown constants, called regression coefficients in statistical contexts. The letter e stands for error, but may reflect a random component in the process that produced the Y_i's (measurement uncertainty, unpredictable physical variability, etc.) or lack of fit in the model, or both. The above model will be called the full model. Each column of X_{ij}'s contains observed values of a variable such as age, conductivity, etc. The full model tries to explain the variation in Y by using all the X variables that are under consideration.

A stepwise regression program such as ANYOLS tries to find a submodel that fits the data well. For notational simplicity, suppose that the submodel consists of the first P columns of the matrix of X_{ij}'s:

$$\begin{pmatrix} Y_1 \\ \vdots \\ Y_{NO} \end{pmatrix} = \begin{pmatrix} X_{1,1} & \cdots & X_{1,P} \\ \vdots & & \vdots \\ X_{NO,1} & \cdots & X_{NO,P} \end{pmatrix} \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_P \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ e_{NO} \end{pmatrix} \qquad (1)$$

abbreviated in matrix notation as

$$Y = X\beta + e.$$

The <u>least squares</u> estimate of β is

$$\hat{\beta} = (X'X)^{-1} X'Y$$

and the corresponding <u>fitted value</u> of Y is

$$\hat{Y} = X\hat{\beta} = X(X'X)^{-1} X'Y$$
$$= HY$$

defining H. The letter H stands for <u>hat matrix</u>, the matrix relating Y to $\hat{Y}$ (Hoaglin and Welsch[1]). The <u>residual vector</u> is defined as

$$R = Y - \hat{Y}.$$

The individual terms of R, $Y_i - \hat{Y}_i$, are called the <u>residuals</u>. The term <u>least squares</u> comes from the fact that, among linear functions of Y, $\hat{\beta}$ minimizes the <u>residual sum of squares</u>

$$RSS = R'R = \sum_{i=1}^{NO} (Y_i - \hat{Y}_i)^2 .$$

(This assertion is proved in books such as Draper and Smith.[2])

If the first column of X is a column of 1's, Model (1) becomes

$$\begin{pmatrix} Y_1 \\ \vdots \\ \vdots \\ Y_{NO} \end{pmatrix} = \beta_1 + \begin{pmatrix} X_{1,2} & \cdots & X_{1,P} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ X_{NO,2} & \cdots & X_{NO,P} \end{pmatrix} \begin{pmatrix} \beta_2 \\ \vdots \\ \beta_P \end{pmatrix} + \begin{pmatrix} e_1 \\ \vdots \\ \vdots \\ e_{NO} \end{pmatrix} .$$

We then say that "the constant is included in the model."

If the data satisfy Model (1) with random e_i's (so that the Y_i's are observed values of random variables), and if

the e_i's are independent normal$(0,\sigma^2)$, (2)

the following can be shown:

$\hat{\beta}$ is multivariate normal$[\beta,\sigma^2(X'X)^{-1}]$
R is multivariate normal$[0,\sigma^2(I-H)]$
s^2 = RSS/(NO-P) is an unbiased estimator of σ^2.

These facts form the basis for the statistical examination of a proposed model.

A generalization of the above is to get the weighted least squares estimate, i.e., find $\hat{\beta}$ to minimize

$$\Sigma w_i^2 (Y_i - \hat{Y}_i)^2$$

for weights w_i. To use the earlier results, just multiply the ith row of X and the ith element of Y by w_i, to get $\tilde{X}$ and $\tilde{Y}$. Then do everything as before, using $\tilde{X}$ and $\tilde{Y}$ instead of X and Y.

A different generalization allows several simultaneous responses. If Y, β, and e are all matrices with NY columns, the definitions for $\hat{\beta}$, $\hat{Y}$, and R can all be used. The quantity corresponding to RSS is R'R, an NY x NY matrix rather than a sum of squares. If NY = 1, this model reduces to Model (1). If NY > 1 and ANYOLS needs to look at a one-dimensional quantity such as RSS or s^2, it will base it on the rightmost column of Y, i.e., the response variable with the highest index.

2.2 Ways to Select a Model

The purpose of fitting a model is to explain the variation in Y. If, based on our understanding of the data, we think that the constant should be included in any reasonable model, the total variation to be explained is $\Sigma (Y_i - \bar{Y})^2$, where $\bar{Y}$ is the mean of the Y_i's. If we think that the constant should not be included, i.e., when the predictor variables are all 0 then Y should be close to 0, the total variation to be explained is ΣY_i^2. The first case is called regression with a constant, and the total variation is the corrected sum of squares. The second case is called regression through the origin, and the total variation is called the uncorrected sum of squares.

For a model to fit the data well, the residuals must be small. A model is underfit if not enough variables have been included, so that the variation in Y has not been explained as completely as possible. It is also possible, in fact dangerously easy, to overfit a model by adding too many columns to X. One way to do this is by introducing new predictor variables. Another way is to introduce high powers of the original predictor variables. If X has so many columns that it is square (the extreme case), and if X is nonsingular, the equation

$$Y = X\beta$$

has an exact solution and the residuals are zero. But this is not a good model. It is fitting itself to random variation, and will be very poor for predicting new values or for explaining any underlying relation between the predictors and the response. A good model selection criterion must guard against both underfit and overfit.

A crude way to assess the adequacy of a fit is to look at

$$R^2 = 1 - \text{RSS/total variation}.$$

R^2 is the portion of the total variation that is explained by the model. A small R^2 indicates a poorly fitting model. A large R^2 indicates very little, since R^2 can be brought up to 1 by overfitting. In the same spirit, suppose that Assumption (2) is accepted for Model (1), and let us test whether all the β_i's (except perhaps the constant) are zero. The test statistic is F for regression, called FREG in ANYOLS. A small value for FREG indicates that the model is not clearly better than no model at all. A large value for FREG could result from good fit or from overfit.

Here is a more refined way to select a model. Suppose that Model (1) is being tentatively considered, and that Assumption (2) is accepted. For each variable in the model, test whether the corresponding β_i is 0, either with an F statistic (called FVAR in ANYOLS) or with a t statistic. If the test statistic is large, we have confidence that β_i is not zero and that the corresponding variable should be included in the model. One way to select a model is to include variables corresponding to large values of FVAR and to remove variables with small values. (This will be referred to as basing the model selection on F.)

Another criterion for deciding on a model is to prefer a model with small s^2. Increasing P, the number of variables in the model, decreases both the numerator and the denominator in the definition of s^2, so it may increase or decrease s^2. In usual practice, the minimum s^2 is attained for an intermediate P, neither too small nor too large.

Another criterion can be used when a good prior estimate of σ^2 is available. The method is known as the C_p method. It takes the observed RSS, attributes part of it to variance (σ^2) and part to lack of fit (or bias). It puts the pieces together to get an estimated mean squared error (MSE) for the fitting equation. A criterion for selecting a model is to try to minimize MSE. See Daniel and Wood,[3] p. 86, for details.

Still another criterion is based on PRESS, the PRediction Error Sum of Squares. For this, one data point is left out, the other data points are fitted to the model, and the error in predicting Y at the left-out point is measured. The sum of squares of these prediction errors, summed over all NO points, is PRESS, and PRESS/NO estimates the squared prediction error at a randomly chosen new point. The selection criterion is to try to minimize PRESS. See Allen[4] for further discussion of PRESS.

Using s^2, MSE, or PRESS automatically guards against overfitting as well as underfitting. Using F guards against overfitting only because an arbitrary value is chosen, and variables with FVAR less than this value are not included in the model.

These and related selection criteria are discussed in a review article by Hocking.[5]

2.3 Diagnostic Checks on the Assumptions

In this section, the goal is not to make assumptions and then select a good model, but rather to check the assumptions. In particular, we try to find data points that do not seem to fit (outliers), points that are extremely influential, or points that seem unusual in other ways. Plots are indispensible.

It is useful to look at the residuals for large values or for nonrandom patterns. See Daniel and Wood[2] or Draper and Smith[5] for instruction in the art of examining residuals. Under Assumption (2), the ith residual R_i has variance $\sigma^2(1 - h_{ii})$, where h_{ii} is the ith diagonal element of H. The standardized residual is $R_i/(s\sqrt{1 - h_{ii}})$. The studentized residual (Reference 1) replaces s by s_{-i}, the estimate of σ based on fitting the model with the ith data point left out. Large standardized or studentized residuals (say magnitude > 3) indicate outliers, lack of fit, or a heavy-tailed distribution for e.

The ith partial residual leaving out variable j (Larsen and McCleary[6]) is defined as

$$Y_i - \sum_{k \neq j} X_{ik} \hat{\beta}_k .$$

That is, fit the current model (which includes variable j), and then calculate the residual with variable j left out. Reference 6 advocates plotting X_j versus the partial residual leaving out X_j.

Closely related in spirit is the partial regression residual leaving out variable j (Bellesly, Kuh, and Welsch[7] or Henderson and Velleman[8]). It is simply the residual resulting from leaving variable j out of the current model and fitting the resulting model. Denote the ith such fitted value by $\hat{Y}_{i,-j}$ and the ith partial regression residual by $Y_i - \hat{Y}_{i,-j}$. This residual shows the unexplained variation in Y after accounting for the predictor variables in the model other than variable j. One could treat variable j as a response variable, fit it to the model that leaves out variable j, and look at

$$X_{ij} - \hat{X}_{ij,-j} .$$

This residual shows what is left of variable j after explaining as much of variable j as possible by the other variables. Reference 8 advocates a partial regression plot, plotting $X_{ij} - \hat{X}_{ij,-j}$ versus $Y_i - \hat{Y}_{i,-j}$ for i = 1,..., NO.

The component effect of variable j on Y_i is

$$\hat{\beta}_j (X_{ij} - \bar{X}_{\cdot j})$$

where

$$\bar{X}_{\cdot j} = \frac{1}{NO} \sum_{i=1}^{NO} X_{ij}$$

and $\hat{\beta}_j$ results from fitting the current model, including variable j. See Reference 3, p. 122, or Wood.[9] In Reference 9, Wood advocates a plot of the component effect-plus-residual.

The leverage of the ith data point (Reference 1) is defined as h_{ii}, the ith diagonal element of the hat matrix. It measures how influential the ith data point is. It can be shown that

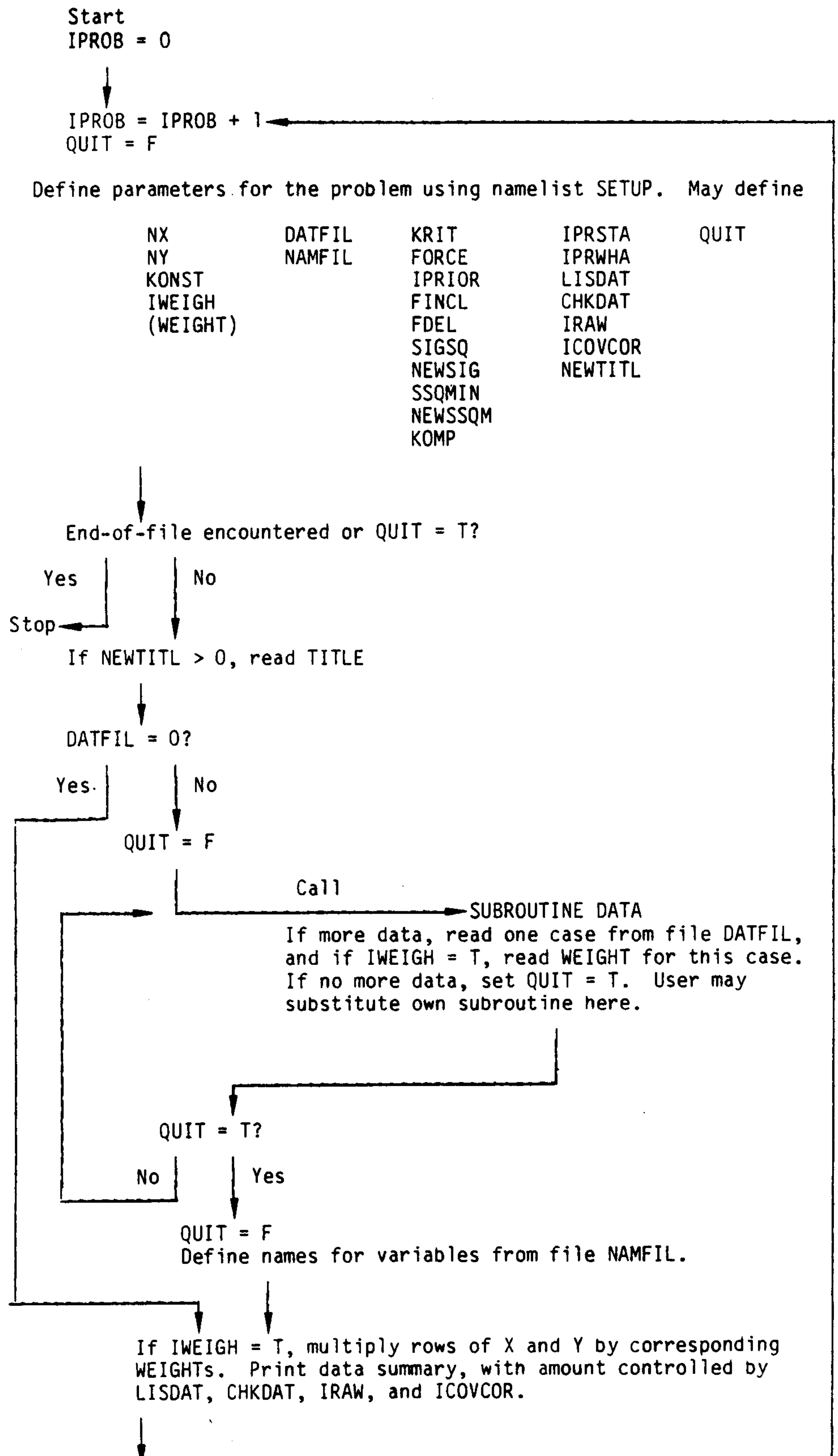
$$\sum_{i=1}^{NO} h_{ii} = P$$

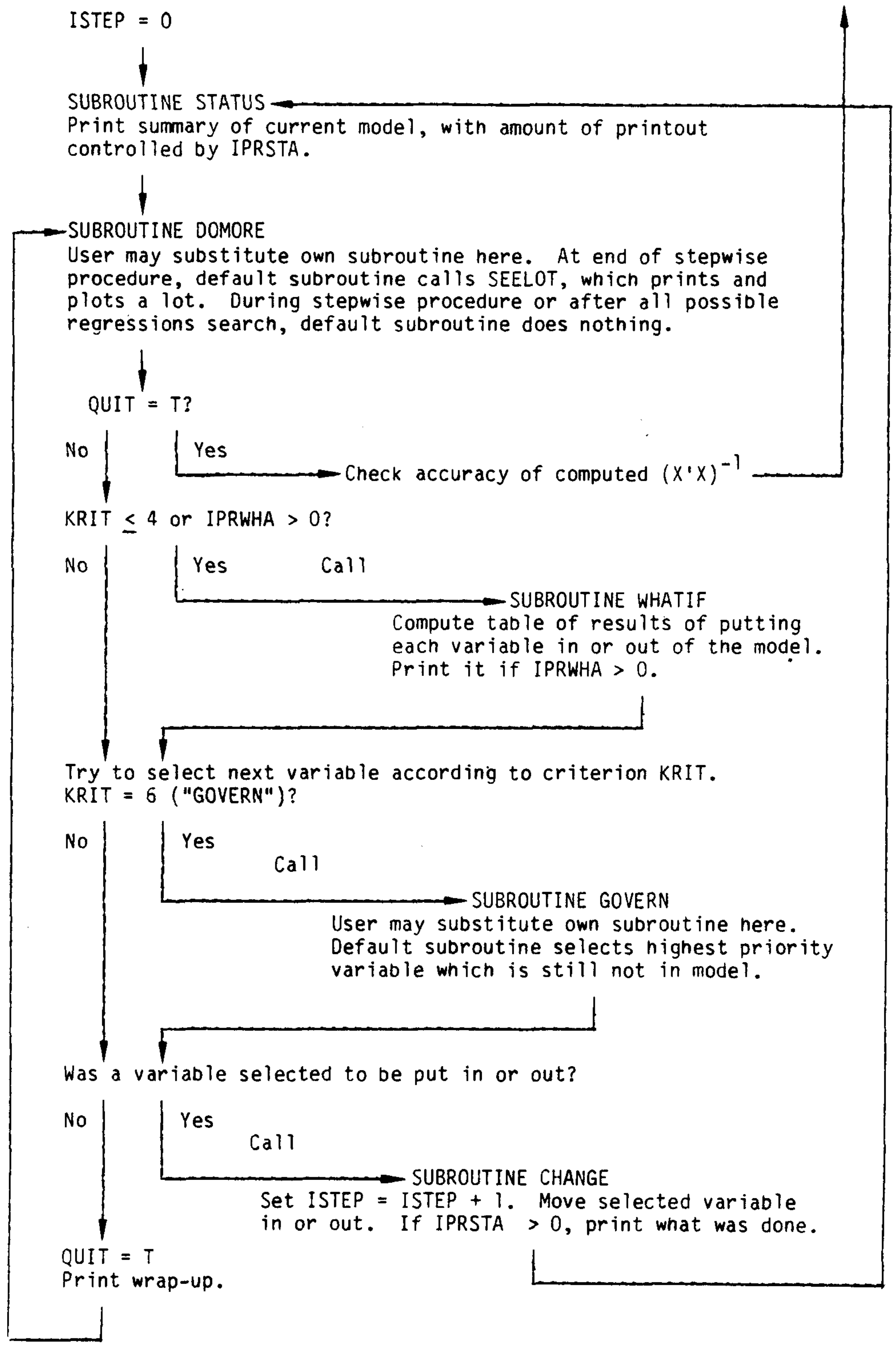
and $0 \leq h_{ii} \leq 1$. If $h_{ii} = 0$, then changing Y_i would not change $\hat{\beta}$ at all. If $h_{ii} = 1$, then the model is devoting an entire parameter to fitting the ith data point alone. For any data point, the partial leverage due to variable j (Reference 8) is the leverage with variable j in the model minus the leverage when variable j is left out.

If any points stand out in plots of any of the above, they deserve further investigation.

3. STRUCTURE OF ANYOLS

The flow diagram given here is explained in detail in the next several sections.





Subroutines which may be called by user-written subroutines

BBIAS	LIST
CHANGE	PLIST
COVB	PLOT
CORB	SEELOT

4. INPUT PARAMETERS

The user defines parameter values by entering them in CDC NAMELIST form. The form is

$SETUP, <parameter name> = <value>,..., $

Column 1 must be blank. Parameter definitions may continue on successive lines. THe final dollar sign is necessary. Examples of such input are given in Section 1.2, in Appendix B, and in CDC FORTRAN manuals.

Parameters keep their values until redefined by the user. The parameters are described here.

4.1 Parameters Defining the Model

NX is the number of predictor variables in the full model. It does not have a default value, but must be defined by the user. When a user tries ANYOLS for the first time, it may be good to define NX and to let everything else take its default value.

NY is the number of response variables, with default 1.

KONST is a logical variable with default value .TRUE. If KONST = .TRUE., the constant is forced into the model. If KONST = .FALSE., the constant is not forced in.

NX must agree with the number of predictor variables to be read later by the subroutine DATA. For example, suppose we wanted to fit the model $Y_i = \beta_1 + \beta_2 X_i$ to the points (4.4, 1.1), (4.6, 1.6), (5.5, -0.6), and (2.8, 2.0). This could be done in three ways.

Parameters

NX = 2	NX = 2	NX = 1
NY = 1	NY = 1	NY = 1
KONST = .TRUE.	KONST = .FALSE.	KONST = .TRUE.

Data Cards

1	4.4	1.1	1	4.4	1.1	4.4	1.1
1	4.6	1.6	1	4.6	1.6	4.6	1.6
1	5.5	-0.6	1	5.5	-0.6	5.5	-0.6
1	2.8	2.0	1	2.8	2.0	2.8	2.0

In the way shown on the left, the program would read three variables on each card and think of them as CONSTANT, X, and Y. Because KONST = .TRUE., it would force the constant into the model on the first step, and would calculate R^2 and FREG based on the corrected sum of squares. In the way shown in the middle, the program would read three variables and think of them as X1, X2, and Y. The fact that X1 happens to be constant would be irrelevant. X1, X2, or both might eventually be put into the model, and either might be entered on the first step. The program would calculate R^2 based on the uncorrected sum of squares. In the way shown on the right, the program would read two numbers, X and Y. Since KONST = .TRUE. but X is not constant, the program would insert 1 before X and change NX from 1 to 2 during this problem. Then, it would proceed as in the first example. At the end of this problem, NX would revert back to 1.

IWEIGH controls whether the method of weighted least squares is used. If IWEIGH = .TRUE., weighted least squares is to be used. The default is IWEIGH = .FALSE. If IWEIGH = .TRUE., the weights must be entered in one of two ways. When new data is being entered, WEIGHT(I) must be read by subroutine DATA along with data case I. When data from a previous problem is to be used, the weights must be entered using namelist SETUP, along with the other input parameters. If data case I has WEIGHT(I), the corresponding term in the sum of squares is multiplied by the square of WEIGHT(I). WEIGHT does not have a default value. If IWEIGH = .TRUE., the user must define WEIGHT.

To use only a portion of the data cases, set IWEIGH = .TRUE. and give each data case weight 0 or 1.

4.2 Parameters Defining the Input Files

Files 5, 11, 12, and 13 are available for input.

DATFIL is the integer number of the file where subroutine DATA will read the data. The default is DATFIL = 5, indicating that the data is on the file INPUT. If DATFIL = 0, the program does not read any data, but uses the data from the preceding problem.

NAMFIL is the number of the file where the program will read the names of the variables. The default is NAMFIL = 5. Each name may be up to ten characters long, with no restrictions on the characters. The program expects to read one name per card image, with format (A10), NX + NY names in all. If NAMFIL = 0, the program does not read the names, but makes up default names like CONSTANT, X1, X2,..., Y.

If DATFIL = 0, then NAMFIL is irrelevant. The program only looks for new names when it has been given new data.

Warning: if the model includes a constant term, there must be a corresponding name. If the program is reading names from the file NAMFIL, you must supply this name. In each of the three related examples of Section 4.1, there must be three names, whether NX is initially defined as 1 or 2. By the time the names are defined, the program is treating NX as 2.

4.3 Parameters Defining the Selection Procedure

The program selects variables to put in or out of the model according to a user-chosen criterion. A variable can be entered into the model only if it satisfies the tolerance condition. Roughly, a variable violates the tolerance condition if it is so nearly dependent on the variables already in the model that entering it would require calculations that cannot be performed with numerical accuracy. A number, the tolerance, is computed for a variable under consideration. On the CDC 176, the variable is not entered if the tolerance is less than 10^{-6}. For details, see Berk.[10] This technicality will not be emphasized below, but will always be implicit.

The order in which variables are entered can be influenced by forcing priorities. If FORCE = .FALSE., no variable (except the constant if KONST = .TRUE.) is forced in or out of the model. The default is FORCE = .FALSE. If FORCE = .TRUE., the NX-dimensional array IPRIOR defines the forcing priorities for the independent variables. IPRIOR does not have a default value, because ANYOLS uses dynamic storage and NX, the dimension of IPRIOR, can change from problem to problem. If IFORCE = .FALSE., the program behaves just as if IPRIOR(I) = 1 for all I. Overriding all other considerations, if KONST = .TRUE., the program behaves as if IPRIOR(1) = 10. (Technically, the program makes a working copy of the user's definition of IPRIOR and "behaves as if" IPRIOR had certain values by making changes in the working copy.) The basic code for IPRIOR is

IPRIOR(I)	Meaning
9, 8, 7, 6	Force in variable I if possible, and never remove it.
5, 4, 3, 2	Force in variable I if possible, but then set IPRIOR(I) = 1.
1	After all forcing-in has been done, put variable I in or out according to the selection criterion.
0	Keep variable I out of the model.
Negative	Keep variable I out, and never mention it in the WHATIF table.

The exact meaning of IPRIOR varies somewhat according to the selection criterion KRIT described below. If the same variables are used in several problems, it might be desirable in some problems to give some variables priority $\leq$ 0. During a problem, the values of IPRIOR may change, because if KONST = .TRUE., then IPRIOR(1) is set to 10, and if IPRIOR(I) $\leq$ 5, then IPRIOR(I) is set to 1 when the variable is put into the model. However, at the end of a problem, IPRIOR reverts back to the values which the user last defined using namelist SETUP. (Technically, the program's working copy of IPRIOR is thrown away at the end of a problem.)

These is one other situation in which IPRIOR gets changed. Suppose that namelist SETUP defines KONST = .TRUE., FORCE = .TRUE., and defines values for IPRIOR(1) through IPRIOR(NX). Suppose also that the first variable read by subroutine DATA is not constant. Then, as mentioned before, the program inserts a 1 at the beginning of each data case, increases NX by 1, and sets IPRIOR(1) = 10. The remaining values of IPRIOR, namely, IPRIOR(2) through IPRIOR(new NX) are given the values that the user defined as IPRIOR(1) through IPRIOR(old NX). IPRIOR reverts to its original values at the end of the problem.

KRIT defines the selection criterion. In CDC NAMELIST input, a variable cannot be entered as a character string, so KRIT is entered as an integer. However, it is immediately converted internally to a mnemonic character string. In future versions of ANYOLS, it is planned that KRIT will be entered directly as a character string. Therefore, in the description below, KRIT will be described as an integer followed by the corresponding character string in parentheses. The default value is KRIT = 2 ("SSQ").

KRIT = 1 ("F") says to select the next variable according to its F statistic. Here F means F for the variable (FVAR), in the model consisting of the variable under consideration and the other variables that are in the model at the time.

Two parameters used are FINCL, "F to include," and FDEL, "F to delete." They have default values 0.1 and 0.05. At each step, the program first looks to see if there are variables which are not in the model and which have IPRIOR > 1. If so, it considers those with the highest value of IPRIOR. Among these variables, it tries the one with the highest value of FVAR. If FVAR > FINCL and the tolerance condition is satisfied, the variable is entered into the model. Otherwise, the variable with the next highest FVAR is tried, and so forth. If no variables with the highest priority can be entered, variables with the next highest priority are tried, and so forth. (If a variable is entered having 2 $\leq$ IPRIOR(I) $\leq$ 5, IPRIOR(I) is changed to 1 when the variable is entered.) If no variable with priority > 1 is entered, the program checks all variables with IPRIOR(I) = 1. First it looks at those that are currently in the model. If it finds any with FVAR < FDEL, it removes from the model the one with the smallest FVAR. Otherwise, it tries to enter the variable with the highest value of FVAR just as it did with the higher priority variables. The procedure terminates when no variable can be moved.

KRIT = 2 ("SSQ"), KRIT = 3 ("MSE"), and KRIT = 4 ("PRESS") say to select the next variable to most reduce s^2, MSE, or PRESS. At each step, the program first looks to see if there are variables which are not in the model and which have IPRIOR > 1. If so, it considers those with the highest value of IPRIOR and tries the variable which most reduces the quantity of interest, s^2, MSE, or PRESS. If the tolerance condition is satisfied and if entering the variable actually does reduce the quantity of interest, the variable is entered into the model. Otherwise, the program tries the variable with the next largest reduction of s^2, MSE, or PRESS, and so forth. If no variable has been entered, those with the next highest priority are tried, and so forth. (If 2 $\leq$ IPRIOR(I) $\leq$ 5, IPRIOR(I) is changed to 1 when variable I is entered.) If no variable with priority > 1 is entered in this way, the program checks all variables with IPRIOR(I) = 1, and enters or removes the one that most reduces the quantity of interest. The procedure terminates when no further reduction can be made in the quantity of interest, s^2, MSE, or PRESS.

As discussed in Section 2.2, MSE cannot be defined without a "good" estimate of σ^2. The estimate is called SIGSQ and may be defined using namelist SETUP. It has no default value. Another way to define SIGSQ is as follows. SSQMIN is initially undefined but, as the stepwise procedure continues, SSQMIN is set to the smallest s^2 value found so far. On problems after the first, defining NEWSIG = .TRUE. causes the program to set SIGSQ equal to the current value of SSQMIN. Setting NEWSSQM = .TRUE. resets SSQMIN as undefined. The defaults are NEWSIG = .TRUE., NEWSSQM = .TRUE. Setting both true causes both SIGSQ and SSQMIN to be set undefined at the beginning of a problem.

An informative message is printed if, at the end of a problem, SSQMIN < SIGSQ. This warns the user that either SIGSQ is too large an estimate of σ^2, or the model for which s^2 = SSQMIN is overfit.

Computation of PRESS requires both the residuals and the diagonal elements of the hat matrix. These values are updated at each step. However, in problems with very many data cases (in the thousands), it may be necessary to suppress these computations. This can be done by setting KOMP = .FALSE. The default is KOMP = .TRUE.

KRIT = 5 ("ALL") says that under this option, all variables with IPRIOR(I) > 1 are first put into the model and left there. All variables with IPRIOR(I) < 1 are kept out. Then, all possible combinations of variables with IPRIOR(I) = 1 are tried. The variables are put in and out in an efficient way so that only $2^k - 1$ one-variable changes are necessary to examine all possible models involving k variables. The method is discussed by Schatzoff, Tsao, and Fienberg.[11]

When KRIT = 5 ("ALL"), a summary is printed at the end showing the five models with smallest s^2, the five with smallest MSE, and the five with smallest PRESS. The procedure terminates before completion if the tolerance condition is violated for a variable that is due to be entered.

Most users will want to set IPRSTA and IPRWHA (defined in Section 4.4) to 0 when using KRIT = 5 ("ALL").

KRIT = 6 ("GOVERN") allows the user to write a subroutine GOVERN to select the variables in any desired way. How to write such a subroutine is described in Section 5.4. The default subroutine enters all the variables which have priority ≥ 1 and which satisfy the tolerance condition, beginning with the highest priority variables and working down.

4.4 Parameters Controlling the Output

IPRSTA (for "I PRint STAtus") controls how complete a summary of the current model is printed at each step. If IPRSTA = 0, 1, or 2, subroutine STATUS prints respectively (0) nothing, (1) a brief summary, or (2) a brief summary, a list of possible outlying points, and a table of the regression coefficients. For details, see the description of subroutine STATUS in Section 6.9. The default for IPRSTA is 2.

When the program is running automatically, IPRSTA has a second effect. When the program enters or removes a variable from the model, if IPRSTA > 0 then the step number and a line of asterisks are printed, and a comment is printed telling which variable was entered or deleted. For details, see the description of subroutine CHANGE in Section 6.2.

IPRWHA (for "I PRint WHAtif") controls the printing of subroutine WHATIF. If IPRWHA = 0, nothing is printed. If IPRWHA > 0, a table is printed showing what would result if each variable were moved in or out of the model. The default is IPRWHA = 1. For details, see the description of WHATIF in Section 6.11.

If CHKDAT = .TRUE., the program prints the minimum, maximum, range, mean, and standard deviation of each variable as input. If LISDAT = .TRUE., the program prints all the data cases that have non-zero weight. It is presumed that, if this information is desired, it will be desired when the data are first read. Therefore, if this same data set is used in a later problem (by setting DATFIL = 0) and if on that problem, every data case has non-zero weight, CHKDAT and LISDAT are ignored. This will help avoid redundant output. The default values are CHKDAT = .TRUE. and LISDAT = .TRUE.

IRAW and ICOVCOR control the printing of the sums of squares and products of the data. The code for IRAW is:

IRAW	Print
1	Sum of squares and cross terms for X's
2	Sum of squares and cross terms for Y's
3	Sum of squares and cross terms for X's and Y's

If IRAW = 0, none of these are printed. The default is IRAW = 3. The code for ICOVCOR is:

	Print		
For	Covariance Matrix	Correlation Matrix	Both Matrices
X's	1	2	3
Y's	4	5	6
X's and Y's	7	8	9

If ICOVCOR = 0, none of these are printed. The default is ICOVCOR = 9.

NEWTITL controls user entry of the title for a problem. The default is NEWTITL = 0, which leaves the title unchanged. (The default for TITLE is blank.) If NEWTITL = 1, immediately after reading the parameter definitions via namelist SETUP, the program will read the next card of the imput file as a title. If NEWTITL = 2, the program will read the next two cards as a title. So, in all, the title may be up to 160 characters long.

4.5 QUIT

The logical variable QUIT is used in various places to terminate execution of one portion of the program and to go on to the next portion, if there is a next portion. If namelist SETUP reads an end-of-file or if it reads QUIT = T, the ANYOLS session terminates.

5. USER-WRITTEN SUBROUTINES

The INEL procedure ANYOLS, described in Section 8, allows easy substitution of user-written subroutines for the default subroutines. The procedure parameter ADD must be set to 0 or 1 in order to tell the procedure whether user-written subroutines are to be used. See Section 8 for full details. This section tells how to rewrite the subroutines DATA, GOVERN, and DOMORE.

5.1 Variables in COMMON Blocks

The computer calculates several variables that may be useful in subroutines. IPROB is the number of the current problem. ISTEP is the number of the current step; it starts at zero and is incremented whenever subroutine CHANGE moves a variable in or out of the model. NV = NX + NY, the number of variables under consideration. During calls to DATA, NO is the number of the current data case being read. After all the data have been read, NO is the number of data cases with non-zero weights. Setting BOMB = .TRUE. tells the program that a serious error has occurred, so stop working. The quantities are stored in COMMON blocks as follows:

```
      COMMON IPROB, NX, NY, NV, NO, KRIT, FORCE, ISTEP,
     &        IPRWHA, IPRSTA
      COMMON/FINEPT/ KONST, IWEIGH, KOMP, SIGSQ, SSQMIN,
     &        NAMFIL, DATFIL, FINCL, FDEL, BOMB, TITLE(16)
      LOGICAL FORCE, QUIT, KONST, IWEIGH, KOMP, BOMB
      INTEGER DATFIL, TITLE
```

Another common block stores quantities which are computed for the model at each step. They may be of interest in subroutines GOVERN and DOMORE. The block is:

```
      COMMON/RESLTS/ RSS, RSQ, SSQ, MSE, MSB, PRESS, FREG,
     &        FVAR, P, DF, DFNUM, K
      REAL MSE, MSB
      INTEGER P, DF, DFNUM
```

Here, RSS is the residual sum of squares, RSQ is R^2, SSQ is s^2, and MSE and MSB are the mean squared error and mean squared bias based on C_p. The quantity stored as PRESS is PRESS/NO, the mean squared prediction error. FREG is F for regression, with DFNUM and DF degrees of freedom in the numerator and denominator. FVAR is F for the variable most recently put in or out of the model, and K is the number indexing that variable. P is the number of variables now in the model, counting the constant if included.

User-written subroutines can communicate with each other through user-defined labeled COMMON blocks

5.2 Files

Files 11, 12, and 13 are available for use in user-written subroutines, as are file 5 (INPUT) and file 6 (OUTPUT).

5.3 Subroutine DATA

The form is

SUBROUTINE DATA(U,WEIGHT,QUIT).

A complete listing of the default subroutine is given in Appendix A. The quantities IPROB, NX, NY, NO, IWEIGH, and DATFIL are available in common blocks. NX and NY are as defined by the user, and NV = NX + NY.

On each call, the default subroutine reads the array U(1),..., U(NV) in free format from the file DATFIL. The first NX U's are the predictor variables, and the next NY are the response variables. If IWEIGH = .TRUE., it also reads WEIGHT for this case, right after U(NV). It sets QUIT = .TRUE., signalling the end of the data, when it encounters an end-of-file on file DATFIL. During calls to DATA, the quantity NO is the number of the current call in this problem.

The user may substitute another subroutine to do any of the following: use a formatted read statement; transform some of the variables, e.g., by taking logs; read fewer than NX predictor variables, and construct the others by calculating powers, products, etc.; flag the end of the data by something other than an end-of-file, e.g., by counting the data cases (but do not count wrong!); and so forth.

5.4 Subroutine GOVERN

The form is

```
  SUBROUTINE GOVERN (KNEXT, DATA, X, Y, BETA, ID, IPRIOR, Q, TABLE, NAM,
&                    RESID, SCRTCH, NX1, NY1, NO1, NVREAD, QUIT)
```

A complete listing of the default subroutine is given in Appendix A. Many of the arguments are arrays, with dimensions given in the listing. The arguments NX1, NY1, and NO1 are equal to NX, NY, and NO. They are included as arguments because they are used in DIMENSION statements, and variables in COMMON, such as NX, are not allowed in FORTRAN DIMENSION statements. NVREAD is the number of variables (X's and Y's) that were passed in by subroutine DATA on each call. The meaning of the other arguments are:

KNEXT — is the index of the next variable that GOVERN selects to be put in or out.

DATA — is the array of data (X's and Y's) as input by subroutine DATA, stored one case at a time.

X and Y — are the arrays of independent and dependent variables for the full model. They are constructed as follows. If KONST = .TRUE. and the first data entry is not the same in all cases, an initial 1 is inserted before each data case. (Then NX + NY becomes equal to NVREAD + 1.) If IWEIGH = .TRUE., each data case is multiplied by the corresponding WEIGHT and cases with weight zero are dropped. Storage is by columns.

BETA(I)	is $\hat{\beta}_i$, the estimated Ith regression coefficient, if variable I is in the model.
ID(I)	is 0 or 1, according as variable I is out of the model or in the model.
IPRIOR(I)	is the forcing priority of variable I.
Q(I)	is $1 - h_{II}$, where h_{II} is the Ith diagonal element of the hat matrix.
TABLE	is computed by subroutine WHATIF and shows the resulting values of various quantities if each independent variable were moved in or out of the model. If WHATIF has not been called on this step, GOVERN may call it. The Ith row of TABLE corresponds to moving variable I in or out of the model and shows the resulting TOL, RSS, SSQ, RSQ, FVAR, MSE, MSB, and PRESS/NO. The quantity TOL is the tolerance of the variable, and the other quantities are as explained in Section 5.1.
NAM	is the array of names of the variables.
RESID	is the vector of residuals.
SCRTCH	is an array that the user may use as desired.

A user-written subroutine may be substituted for the default GOVERN. At each step, it may select KNEXT in any way desired, using any of the above quantities. The stepwise procedure terminates when GOVERN sets QUIT = .TRUE.

5.5 Subroutine DOMORE

The form is

```
  SUBROUTINE DOMORE (DATA, X, Y, BETA, ID, IPRIOR, Q, TABLE, NAM,
&                    RESID, SCRTCH, NX1, NY1, NO1, NVREAD, QUIT)
```

A complete listing of the default subroutine is given in Appendix C. The meaning of the arguments is as explained in the last subsection. The default subroutine does not use any of these arguments, but a user-written subroutine may. At the end of a stepwise process, the default subroutine calls the subroutine SEELOT. Otherwise, the default subroutine simply returns.

A user-written subroutine could, for example, print or plot output of interest at each step. Or, it could let a problem run to completion and then redefine KRIT and set QUIT = .FALSE. Then, the program would continue running based on the new value of KRIT. Or, DOMORE could decide to remove a variable, and call CHANGE directly to do it. Or, DOMORE could call other user-written subroutines. The flexibility of ANYOLS is limited primarily by the user's ingenuity in rewriting GOVERN and DOMORE.

6. OTHER SUBROUTINES

There are various subroutines available which the user may wish to call. They are described below.

6.1 BBIAS

If the full model is correct and the current model is used, $\hat{\beta}$ is biased. BBIAS prints the biases. The biases, of necessity, are in terms of the (unknown) β_i's corresponding to the variables that are not in the current model.

6.2 CHANGE (N, IPRINT, QUIT)

This subroutine moves the requested variable in or out of the model and increments ISTEP by 1. N can be either the name or the number of the selected variable; execution is negligibly faster if it is the number. If the variable cannot be entered because it does not satisfy the tolerance condition, QUIT is set to .TRUE.

If IPRINT = 0, the subroutine prints nothing. If IPRINT > 0, it sets off the new portion of the output by printing ISTEP and a line of asterisks. It then prints a statement telling which variable was moved in or out of the model. When the program is running automatically, IPRSTA is used for IPRINT in calls to CHANGE.

6.3 COVB, CORB, and COVCRB

These print the estimated covariance matrix of $\hat{\beta}$, the correlation matrix of $\hat{\beta}$, and both, respectively. Assumption (2) is used and s^2 is used to estimate σ^2 in the covariance of $\hat{\beta}$.

The meaning of these matrices is this: the ith diagonal term of the estimated covariance matrix is the estimated variance of $\hat{\beta}_i$. The (i,j)th term is the estimated covariance of $\hat{\beta}_i$ and $\hat{\beta}_j$. The (i,j)th term of the correlation matrix is the statistical correlation between $\hat{\beta}_i$ and $\hat{\beta}_j$. The covariance and correlation matrices are both symmetric.

6.4 GETF(FIT,N), GETR(RES,N), and GETS(SRES,N)

These get the fitted values, the residuals, and the studentized residuals, respectively. N can be either the name or the number of the variable of interest. For example, if variable NX + 1 is named "Y", the calls

```
CALL GETF(SCRTCH,"Y")
```

and

```
CALL GETS(SCRTCH(NO + 1), NX + 1)
```

would get the fitted values and studentized residuals of Y. The fitted values and studentized residuals would each be stored as NO values in the array SCRTCH, the fitted values starting at position 1, and the studentized residuals starting at position NO + 1.

The variable identified by N may be either a predictor variable or a response variable, but it must not be in the model.

If KOMP = .TRUE., the usual case, the residuals for Y are in the array RESID, so GETR is not needed.

6.5 GETQ

This calculates the array Q, defined under the subroutine GOVERN. If KOMP = .TRUE., Q is automatically calculated, so GETQ is not needed.

6.6 GETTOL(N,TOL,OKAY)

This calculates the tolerance, TOL, for the variable with name or number N. The logical variable OKAY is set .TRUE. if TOL $\geq 10^{-6}$ and set .FALSE. otherwise. If OKAY = .TRUE., the variable identified by N can be entered without undue numerical inaccuracy.

6.7 PLOT, LIST, AND PLIST

These produce a plot, a listing, and both, respectively, for two user-specified quantities. The call for PLOT is

```
CALL PLOT(CODE1,N1,CODE2,N2,LTITLE)
```

and the calls for LIST and PLIST are analogous. Here, LTITLE is a six-computer-word (60-character) array containing the plot title. N1 and N2 are the names or numbers of the variables to be plotted on the horizontal and vertical axes, respectively. The codes are mnemonic character strings, defined below.

"D"	Observed Data value
"F"	Fitted value
"R"	Residual
"SR"	Studentized Residual
"PRS"	Partial ReSidual
"C"	Component effect
"C+R"	Component effect plus Residual
"N"	Sequence Number, in order of data input
"L"	Leverage

"PL" Partial Leverage

"PRG" Partial ReGression residual

For example

```
CALL PLIST ("D", 3, "SR", "HEAT FLUX", LT)
```

would produce a plot and a listing of the raw data values of variable 3 versus the studentized residuals of the variable named Heat Flux. The title would be the contents of the array LT.

There are several details which must be mentioned for some of the codes: if the code is "N" or "L," the corresponding name is ignored. For example,

```
CALL PLOT("N", JUNK, "L", 87, LT)
```

would produce a plot of the leverages of the points in the order of the data input. The entries JUNK and 87 are ignored.

Partial residuals and partial regression residuals look at the residuals for one variable while another variable is left out of the model. The convention used in PLOT is that, if CODEi is "PRS" or "PRG", the residuals for the corresponding Ni are looked at while the variable N1 is left out. For example,

```
CALL PLOT("PRG", "X3", "PRG", "Y", LT)
```

would produce the partial regression plot described in Section 2.3, plotting the residual of X3 when X3 is out of the model versus the residual of Y when X3 is out of the model.

All the plots use the IGS graphics system. There is one subroutine in ANYOLS that makes all the calls to IGS. It can be replaced by a subroutine that uses DISSPLA. The INEL procedure described in Section 8 makes it easy to use either IGS or DISSPLA.

6.8 SEELOT(Y,ID,NAM,RESID,Q,SCRTCH,NX1,NY1,NO1)

This subroutine prints and plots lots of things. The arguments have the same meanings as described under subroutine GOVERN. A listing is given in Appendix A, as a guide that the user may wish to modify in writing an output subroutine. SEELOT does the following:

If NY > 1, call SUMSQ.

Call COVCRB.

Print a table showing Y, $\hat{Y}$, residuals, studentized residuals, and leverages, for the NO observations. If NY > 1, only do this for the highest indexed Y.

plot $\hat{Y}$ versus residuals
$\hat{Y}$ versus studentized residuals
$\hat{Y}$ versus Y
residuals in order of data input
leverages in order of data input.

If the model involves at most five independent variables (not counting the constant), then for each X_i in the model:
plot X_i versus R
do partial regression plot of Y with X_i removed
plot X_i versus partial leverage for X_i.

6.9 STATUS(IPRINT)

If IPRINT = 0, STATUS prints nothing.

If IPRINT = 1, STATUS prints a compact statement of what variables are in the model. It then prints RSS, SSQ, RSQ, FREG, FVAR, DFNUM, DF, MSE, MSB, and PRESS/NO. These quantities have the meanings explained in Section 5.1.

If IPRINT = 2, STATUS also prints points that have high or low leverage [greater than 2*P/NO or less than P/(2*NO)] and points with large standardized residuals (magnitude greater than 3). It also prints the values of $\hat{\beta}$, their estimated standard deviations, and the associated Student's t values, for the variables in the model.

When the program is running automatically, IPRSTA is used for IPRINT in calls to STATUS.

6.10 SUMSQ

If NY > 1, it is of interest to know the NY x NY matrix R'R. The subroutine SUMSQ prints this matrix. The lower right element is equal to RSS.

6.11 WHATIF(IPRINT)

WHATIF computes TABLE, described under subroutine GOVERN. If IPRINT > 0, WHATIF prints TABLE. When the program is running automatically and performing a stepwise regression, IPRWHA is used for IPRINT in calls to WHATIF.

6.12 YBIAS(U)

Here, U is an NX-dimensional vector, the point that we are interested in. If the full model is correct and the current model is not the full model, $\hat{Y}$ is a biased estimate of EY at U. Subroutine YBIAS computes the bias. Of necessity, the answer is in terms of the (unknown) regression coefficients corresponding to the variables that are not in the model.

7. EXAMPLE

The job in Appendix A contains the following.

Job control cards
User-written subroutines DATA and DOMORE
NAMELIST input for problem 1
Data cards
Names of the variables
NAMELIST input for problem 2.

The job control cards use the INEL procedure ANYOLS, described in Section 8.

The user-written subroutine DATA reads three predictor variables and one response variable. It then defines the first predictor variable to be the constant 1, and it calculates the squares and products of the three basic predictor variables, making, in all, ten predictor variables and one response variable. The names of these variables are given in 11 lines later in the job sequence. The subroutine DATA is written to keep reading data until it encounters an end of file. Therefore, the data card images end with a *EOR. (This was input from a terminal; from cards, a 7/8/9 multiply punched in column 1 would be used.)

The user-written subroutine DOMORE just copies the default subroutine on problem 1. On problem 2, it waits until a model has been selected using KRIT = 2 ("SSQ"). Then, it sets SIGSQ = SSQMIN, so now MSE is defined, and it changes the selection criterion to PRESS. It then lets the stepwise procedure resume. When a model has again been selected, it calls BBIAS and YBIAS and uses the default subroutine.

The input for problem 1 sets NX = 10. It sets KRIT = 1, so the selection criterion will be F. It sets NEWTITL = 1, so the program looks to the next card image for a title.

The input for problem 2 sets DATFIL = 0, so the old data and names will be used. It sets IWEIGH = .TRUE., so the data cases will be weighted. The values of WEIGHT are 0 for cases 1, 3, 4, and 21, and 1 for the other cases. The four cases are given weight 0 because they are apparently outliers (see Reference 3, p. 74). Now, FORCE = .TRUE., so the variables are forced, with priorities given by IPRIOR, 2 for the constant and the three linear terms, and 1 for the other six variables. (Since KONST has the default value of .TRUE., the priority for the constant will be overridden and set to 10.) The value of KRIT is 2, so variable selection will be according to s^2. This will be changed by subroutine DOMORE during the problem. Finally, NEWTITL = 0, so no title is read; the old title is used.

There is input for a problem 3, simply setting QUIT = T. This terminates the job. This line is optional. If the line were omitted, the program would encounter an end-of-file here, report it, and terminate.

The output for this job is lengthy, so not all of it is included in this user's guide. Most of the output for problem 1 is given in Appendix C.

When the job was run on February 10, 1981, it used 1.43 CP seconds and 21.96 system seconds, and it cost 34 cents at priority 1.

8. HOW TO ACCESS ANYOLS AT INEL

On the INEL Cyber, a procedure has been prepared to do all the file handling. The procedure is called ANYOLS, and has several associated optional parameters. They are:

I = <file name> This specifies the input file. The default is I = INPUT.

GR = <code> This specifies the graphics package to be used. The code is

I	- IGS
D	- DISSPLA
any other symbol	- no graphics

The default is GR = I.

ADD = <0 or 1> The code is

0 - User-written FORTRAN subroutines are not used.
1 - User-written FORTRAN subroutines are used.

The default is ADD = 1.

PL = <integer> This specifies the output print limit. The default is PL = 5000.

ID = <user ID> This causes the graphics file (TAPE10 with IGS, PLFILE with DISSPLA) to be catalogued as ANYOLPL, with the entered ID, and RP = 3. It can then be viewed on a Tektronix screen before the film arrives. The default is ID = 000.

WA = <integer> This defines the size of the work area to be used. The default is 30000. The maximum that will fit in a CDC 176 is approximately 90000.

Therefore, the two commands

ANYOLS

and

ANYOLS,I=INPUT,GR=I,ADD=1,PL=5000

are equivalent.

The job deck would then look like this.

Job card
Account card
<ATTACH, any input or data files needed.>
ATTACH,ANYOLS,ID=CLA.
ANYOLS <,with parameters if desired>.
*EOR
FORTRAN subroutines } Unless ANYOLS card says ADD = 0.
*EOR
Input deck } Unless ANYOLS card says I = some file other than INPUT.

9. INDEX

Term	References (Definitions Underlined) by section	Default if Applicable. Dimension in Parentheses if Applicable
Influential points	2.3	
IPRIOR	3, 4.3, 5.4, 5.5, 7	(NX)
IPROB	3, 5.1, 5.3	
IPRSTA	3, 4.3, 4.4, 5.1, 6.2, 6.9	2
IPRWHA	3, 4.3, 4.4, 5.1, 6.11	1
IRAW	3, 4.4	3
ISTEP	3, 5.1, 6.2	
IWEIGH	3, 4.1, 5.1, 5.3, 5.4, 7	.FALSE.
K	5.1	
KNEXT	5.4	
KOMP	3, 4.3, 5.1, 6.4, 6.5	.TRUE.
KONST	3, 4.1, 4.3, 5.1, 5.4, 7	.TRUE.
KRIT	3, 4.3, 5.1, 5.5, 7	2 ("SSQ")
Lack of fit	2.2, 2.3	
Least squares	2.1	
Leverage	1.1, 2.3, 6.7, 6.8, 6.9	
LISDAT	3, 4.4	.TRUE.
Model	2.1	
MSB	5.1, 5.4, 6.9	
MSE, "MSE"	1.1, 2.2, 4.3, 5.1, 5.4, 6.9, 7	
NAM	5.4, 5.5, 6.8	(NV)
Namelist SETUP	1.2, 4, Appendix B	
NAMFIL	3, 4.2, 5.1	5

Term	References (Definitions Underlined) by section	Default if Applicable. Dimension in Parentheses if Applicable
NEWSIG	3, 4.3	.TRUE.
NEWSSQM	3, 4.3	.TRUE.
NEWTITL	3, 4.4, 7	0
NO	2.1, 5.1, 5.3, 6.4	
NO1	5.4, 5.5, 6.8	
NV	5.1, 5.3	
NVREAD	5.4, 5.5	
NX	2.1, 3, 4.1, 5.1, 5.3, 7	undefined
NX1	5.4, 5.5, 6.8	
NY	2.1, 3, 4.1, 5.1, 5.3, 6.8, 6.10	1
NY1	5.4, 5.5, 6.8	
OKAY	6.6	
Outliers	2.3	
Overfit	2.2	
P	2.1, 2.2, 5.1	
Partial leverage	1.1, 2.3, 6.7, 6.8	
Partial regression	1.1, 2.3, 6.7, 6.8	
Partial residual	2.3, 6.7	
Plots	2.3, 6.7, 6.8, 8, Appendix C	
Predictor variables	2.1, 2.2, 2.3, 4.1, 5.3, 6.4	
PRESS, "PRESS"	1.1, 2.2, 4.3, 5.1, 5.4, 6.9, 7	
Q	5.4, 5.5, 6.5, 6.8	(NO)

Term	References (Definitions Underlined) by section	Default if Applicable. Dimension in Parentheses if Applicable
QUIT	3, 4.5, 5.3, 5.4, 5.5, 6.2, 7	
R	1.1, 2.1, 2.3, 5.4, 6.4, 6.7, 6.8, 6.10	(NO)
Regression coefficients	2.1	
Regression through the origin	2.2, 4.1	
Regression with a constant	2.2, 4.1	
RESID	5.4, 5.5, 6.4, 6.8	(NO,NX)
Residual (see R)		
Residual sum of squares (see RSS)		
Response variables	2.1, 2.2, 2.3, 4.1, 5.3, 6.4	
Re-using data	1.1, 4.2, 4.4, 7	
RSQ, R^2	2.2, 4.1, 5.1, 5.4, 6.9	
RSS	2.1, 2.2, 5.1, 5.4, 6.9, 6.10	
SCRTCH	5.4, 5.5, 6.4, 6.8	(*)
Sequency number	6.7	
SETUP (see namelist SETUP)		
σ^2	2.1, 2.2, 2.3, 4.3, 6.3	
SIGSQ	3, 4.3, 5.1, 7	
SSQ, s^2, "SSQ"	1.1, 2.1, 2.2, 4.3, 5.1, 5.4, 6.3, 6.9, 7	
SSQMIN	3, 4.3, 5.1, 7	
Standardized residuals	2.3, 6.9	

Term	References (Definitions Underlined) by section	Default if Applicable. Dimension in Parentheses if Applicable
Studentized residuals	1.1, 2.3, 6.4, 6.7, 6.8	
Student's t	2.2, 6.9	
Subroutine BBIAS	3, 6.1, 7	
Subroutine CHANGE	3, 4.4, 5.1, 5.5, 6.2	
Subroutine CORB	3, 6.3	
Subroutine COVB	3, 6.3	
Subroutine COVCRB	3, 6.3, 6.8	
Subroutine DATA	3, 4.1, 5.1, 5.3, 7, Appendix A	
Subroutine DOMORE	3, 5.5, 7, Appendix A	
Subroutine GETF	3, 6.4	
Subroutine GETQ	3, 6.5	
Subroutine GETR	3, 6.4	
Subroutine GETS	3, 6.4	
Subroutine GETTOL	3, 6.6	
Subroutine GOVERN	3, 4.3, 5.1, 5.4, Appendix A	
Subroutine LIST	3, 6.7	
Subroutine PLIST	3, 6.7	
Subroutine PLOT	3, 6.7	
Subroutine SEELOT	3, 5.5, 6.8, Appendix A	
Subroutine STATUS	3, 4.4, 6.9	
Subroutine SUMSQ	3, 6.8, 6.10	
Subroutine WHATIF	3, 4.4, 5.4, 6.11	
Subroutine YBIAS	3, 6.12, 7	

Term	References (Definitions Underlined) by section	Default if Applicable. Dimension in Parentheses if Applicable
TABLE	5.4, 5.5, 6.11	(NX,8)
TITLE	3, 4.4, 5.1	(16)
TOL	5.4, 6.6	
Tolerance	4.3, 5.4	
Total variation	2.1	
Uncorrected sum of squares	2.2, 4.1	
Underfit	2.2	
User-written subroutine	1.1, 5.1, 8, Appendix B	
Variance (see also SSQ, σ^2)	2.1, 6.3	
WEIGHT	3, 4.1, 5,3, 5.4, 7	(NO)
Weighted least squares (see IWEIGH, WEIGHT)	2.1	
X	2.1, 5.4, 5.5	(NO,NX)
Y	2.1, 5.4, 5.5, 6.8	(NO,NY)
Y (see fitted value)	2.1, 6.8, 6.12	

10. REFERENCES

1. D. C. Hoaglin and R. E. Welsch, "The Hat Matrix in Regression and ANOVA," The American Statistician, 32, 1, 1978, pp 17-22.

2. N. R. Draper and H. Smith, Applied Regression Analysis, New York: Wiley, 1966.

3. C. Daniel and F. S. Wood, Fitting Equations to Data, New York: Wiley-Interscience, 1971.

4. D. M. Allen, "The Relationship Between Variable Selection and Data Augmentation and a Method for Prediction," Technometrics, 16, 1974, pp 125-127.

5. R. R. Hocking, "The Analysis and Selection of Variables in Linear Regression," Biometrics, 32, 1976, pp 1-50.

6. W. Larsen and S. McCleary, "The Use of Partial Residual Plots in Regression Analysis," Technometrics, 14, 3, 1972, pp 781-790.

7. D. Bellesly, E. Kuh, and R. E. Welsch, Regression Diagnostics: Identifying Influential Data and Sources of Collinearity, New York: Wiley, 1980.

8. H. V. Henderson and P. F. Velleman, Building Multiple Regression Models Interactively, Economic and Social Statistics Technical Report No. 1279/023, New York State School of Industrial and Labor Relations, Cornell University, 1979.

9. F. S. Wood, "The Use of Individual Effects and Residuals in Fitting Equations to Data," Technometrics, 15, 4, 1973, pp 677-695.

10. K. N. Berk, "Tolerance and Condition in Regression Computations," Journal of the American Statistical Association, 72, 360, 1977, pp 863-866.

11. M. Schatzoff, R. Tsao, and S. Fienberg, "Efficient Calculation of All Possible Regressions," Technometrics, 10, 1968, pp 769-779.

APPENDIX A

LISTINGS OF CERTAIN SUBROUTINES

```
      SUBROUTINE DATA(U,WEIGHT,QUIT)
C     READS CARD IMAGE DATA FROM DATFIL (DEFAULT = TAPE5)
C     STOPS WHEN ENCOUNTERS EOF
      COMMON IPROB,NX,NY,NV,NO
      COMMON /FINEPT/ KONST,IWEIGH,KOMP,
     &                SIGSQ,SSQMIN,NAMFIL,DATFIL,
     &                FINCL,FDEL,BOMB
      INTEGER DATFIL
      LOGICAL QUIT,IWEIGH,BOMB
      DIMENSION U(2)
      READ(DATFIL,*) U(1)
         IF(EOF(DATFIL).EQ.0) GO TO 20
         QUIT = .TRUE.
         RETURN
   20 CONTINUE
      IF(.NOT.IWEIGH) READ(DATFIL,*) (U(I),I=2,NV)
      IF(EOF(DATFIL).NE.0) GO TO 100
      IF(IWEIGH) READ(DATFIL,*) (U(I),I=2,NV),WEIGHT
      IF(EOF(DATFIL).NE.0) GO TO 100
      RETURN
  100    QUIT = .TRUE.
         BOMB = .TRUE.
         IF(.NOT.IWEIGH) WRITE(6,1000) NO,NV,(U(I),I=1,NV)
         IF(IWEIGH) WRITE(6,1001) NO,NV,(U(I),I=1,NV),WEIGHT
         RETURN
 1000 FORMAT(/" ** SUBROUTINE DATA ENCOUNTERED EOF IN THE MIDDLE OF ",
     &       "CASE",I6/" THE",I5," VALUES NOW IN U ARE "/(6E15.6))
 1001 FORMAT(/" ** SUBROUTINE DATA ENCOUNTERED EOF IN THE MIDDLE OF ",
     &       "CASE",I6/" THE",I6," VALUES NOW IN U, AND THE WEIGHT ",
     &       "FOR THIS CASE, ARE"/(6E15.6))
      END
```

```
      SUBROUTINE DOMORE(DATA,X,Y,BETA,ID,IPRIOR,Q,TABLE,NAM,RESID,
     &                  SCRTCH,NX1,NY1,NO1,NVREAD,QUIT)
C     CALLS SEELOT WHEN FINAL MODEL HAS BEEN SELECTED, EXCEPT AFTER
C     ALL POSSIBLE REGRESSIONS SEARCH
C     USER MAY MODIFY THIS SUBROUTINE AS DESIRED
      COMMON IPROB,NX,NY,NV,NO,KRIT,FORCE,ISTEP,IPRWHA,IPRSTA
      LOGICAL QUIT
      DATA IDOALL /0/
      IF(KRIT.EQ.3HALL) IDOALL = 1
      IF(.NOT.QUIT) RETURN
      IF(IDOALL.EQ.0)CALL SEELOT(Y,ID,NAM,RESID,Q,SCRTCH,NX1,
     &                           NY1,NO1)
      IDOALL = 0
      RETURN
      END
```

```
      SUBROUTINE GOVERN(KNEXT,DATA,X,Y,BETA,ID,IPRIOR,Q,TABLE,NAM,RESID,
     &                  SCRTCH,NX1,NY1,NO1,NVREAD,QUIT)
C
C     THE DEFAULT VERSION, GIVEN HERE, PUTS IN VARIABLES ONE AT A TIME,
C     PUTTING IN HIGHEST PRIORITY VARIABLES FIRST.
C     K, IN COMMON /RESLTS/, IS LAST VARIABLE ENTERED.
C     KNEXT, PASSED AS PARAMETER, IS NEXT VARIABLE TO BE ENTERED.
C
C     USER MAY CHANGE PROGRAM TO SELECT KNEXT AS WISHED.
C     PROCEDURE TERMINATES WHEN QUIT = .TRUE.
C
      COMMON IPROB,NX,NY,NV,NO,KRIT,FORCE,ISTEP,IPRWHA,IPRSTA
      COMMON /FINEPT/ KONST,IWEIGH,KOMP,
     &                FDEL,BOMB,TITLE(16)
      COMMON /RESLTS/ RSS,RSQ,SSQ,MSE,MSB,PRESS,FREG,FVAR,P,DF,DFNUM,K
      REAL MSE,MSB
      LOGICAL QUIT
      DIMENSION DATA(NVREAD,NO1),X(NO1,NX1),Y(NO1,NY1),BETA(NX1,NY1),
     &      ID(NX1),IPRIOR(NX1),Q(NO1),TABLE(NX1,8),NAM(1),
     &      RESID(NO1,NY1),SCRTCH(1)
C
C     NX1=NX, NY1=NY, NO1=NO.  THEY ARE ONLY USED BECAUSE QUANTITIES
C     IN COMMON MAY NOT BE USED AS VARIABLE DIMENSIONS.  USE NX, ETC. IN
C     THE PROGRAM.
C
      DIMENSION NOUT(10)
C
      DO 10 J=1,10
   10   NOUT(J)=0
      DO 20 I=1,NX
   20   IF(IPRIOR(I).GT.0 .AND. ID(I).EQ.0)
     &            NOUT(IPRIOR(I)) = NOUT(IPRIOR(I)) + 1
      DO 40 JJ=1,10
        J=11-JJ
        IF(NOUT(J).EQ.0) GO TO 40
        DO 30 I=1,NX
          IF(IPRIOR(I).EQ.J .AND. ID(I).EQ.0) GO TO 50
   30     CONTINUE
   40   CONTINUE
      KNEXT = 1
      QUIT = .TRUE.
      RETURN
   50 KNEXT=I
      RETURN
      END
```

```
      SUBROUTINE SEELOT(Y,ID,NAM,RESID,Q,SCRTCH,NX1,NY1,NO1)
C     PRINTS LOTS OF INFORMATION ABOUT CURRENT MODEL, AND
C     GENERATES LOTS OF PLOTS.
C     THIS SUBROUTINE IS CALLED BY THE DEFAULT VERSION OF DOMORE WHEN
C     THE STEPWISE PROCEDURE HAS DECIDED ON A FINAL MODEL.
C     IF THE USER IS WRITING OWN SUBROUTINES, SEELOT MAY BE CALLED,
C     OR USED AS A MODEL FOR USER'S OUTPUT SUBROUTINE.

      COMMON IPROB,NX,NY,NV,NO,KRIT,FORCE,ISTEP,IPRWHA,IPRSTA
      COMMON /FINEPT/ KONST,IWEIGH,KOMP,
     &                SIGSQ,SSQMIN,NAMFIL,DATFIL,
     &                FINCL,FDEL,BOMB,TITLE(16)
      COMMON /RESLTS/ RSS,RSQ,SSQ,MSE,MSB,PRESS,FREG,FVAR,P
      INTEGER P
      LOGICAL KONST,KOMP
      DIMENSION Y(NO1,NY1),ID(NX1),NAM(1),RESID(NO1,NY1),Q(1),SCRTCH(1)

C     NX1=NX, NY1=NY, NO1=NO.  THEY ARE ONLY USED BECAUSE QUANTITIES
C     IN COMMON MAY NOT BE USED AS VARIABLE DIMENSIONS.  USE NX, ETC. IN
C     THE PROGRAM.

      DIMENSION LT1(6),LT2(6),LT3(6),LT4(6),LT5(6),LT6(6),LT7(6),
     &          LT8(6),FMT(6)
      DATA LT1 /"FITTED VS. RESIDUALS                                  "/
      DATA LT2 /"FITTED VS. STUDENTIZED RESIDUALS                      "/
      DATA LT3 /"FITTED VS. OBSERVED                                   "/
      DATA LT5 /"RESIDUALS IN ORDER OF DATA INPUT                      "/
      DATA LT6 /"LEVERAGE OF DATA CASES, IN ORDER OF DATA INPUT        "/

      IF(NY.GT.1) CALL SUMSQ
      CALL COVCRB
      CALL GETF(SCRTCH(1),NAM(NV))
      CALL GETS(SCRTCH(NO+1),NAM(NV))
      IF(.NOT.KOMP) CALL GETR(SCRTCH(2*NO+1),NAM(NV))
      IF(.NOT.KOMP) CALL GETQ
      WRITE(6,1000) NAM(NV)
      DO 10 I=1,NO
        XLEVER = 1 - Q(I)
        IF(KOMP) WRITE(6,1001)
     &    I,Y(I,NY),SCRTCH(I),RESID(I,NY),SCRTCH(NO+I),XLEVER
        IF(.NOT.KOMP) WRITE(6,1001)
     &    I,Y(I,NY),SCRTCH(I),SCRTCH(2*NO+I),SCRTCH(NO+I),XLEVER
   10   CONTINUE
      CALL PLOT("F",NV,"R",NV,LT1)
      CALL PLOT("F",NV,"SR",NV,LT2)
      CALL PLOT("F",NV,"D",NV,LT3)
      CALL PLOT("N",NV,"R",NV,LT5)
      CALL PLOT("N",NV,"L",NV,LT6)
      IF(.NOT.KONST) K = 0
      IF(KONST)      K = 1
      IF(P.GT.5+K) GO TO 50
      IF(P.LE.K) GO TO 50
        IBOT=1+K
        DO 20 I=IBOT,NX
          IF(ID(I).EQ.0) GO TO 20
          CALL CHARCT(NAM(I),1,NCHAR)
          ENCODE(60,1002,LT4)
          ENCODE(60,1003,FMT) NCHAR
          ENCODE(60,FMT,LT4)  NAM(I)
          CALL PLOT("D",I,"R",NV,LT4)
   20     CONTINUE
        DO 30 I=IBOT,NX
          IF(ID(I).EQ.0) GO TO 30
          ENCODE(60,1004,LT7) NAM(I)
          CALL PLOT("PRG",I,"PRG",NV,LT7)
   30     CONTINUE
        DO 40 I=IBOT,NX
          IF(ID(I).EQ.0) GO TO 40
          CALL CHARCT(NAM(I),1,NCHAR)
          ENCODE(60,1002,LT8)
          ENCODE(60,1006,FMT) NCHAR,NCHAR
          ENCODE(60,FMT,LT8)  NAM(I),NAM(I)
          CALL PLOT("D",I,"PL",I,LT8)
   40     CONTINUE
   50 CONTINUE
      RETURN
 1000 FORMAT(T12,"OBSERVED",T31,"FITTED",T57,"STUDENTIZED"/2X,
     &       "CASE",T12,A10,T31,"VALUE",T43,"RESIDUAL",T58,"RESIDUAL",
     &       T70,"LEVERAGE"/)
 1001 FORMAT(1X,I5,3E15.5,2F12.3)
 1002 FORMAT(60X)
 1003 FORMAT("(A", I2 ,", "" VS. RESIDUALS"")", 38X)
 1004 FORMAT("PARTIAL REGRESSION PLOT, REMOVING ",A10,16X)
 1006 FORMAT("(""OBSERVED "",A", I2 ,", "" VS. PARTIAL LEVERAGE ",
     &       "DUE TO "",A", I2 ,")")
      END
```

APPENDIX B

INPUT FOR SAMPLE JOB

```
CLANY,P1,T37.
ACCOUNT,E230,DCNTUDARE,TM4.
ATTACH,ANYOLS,ID=CLA.
ANYOLS.
*FOR
      SUBROUTINE DATA(U,WEIGHT,QUIT)
      COMMON IPROB,NX,NY,NV,NO
      LOGICAL QUIT
      DIMENSION U(11)

      READ(5,*) U(2),U(3),U(4),U(11)
        IF(EOF(5).NE.0) QUIT = .TRUE.
        IF(QUIT) RETURN
      U(1)=1
      DO 10 I=1,3
   10   U(4+I)=U(1+I)**2
      U(8)=U(2)*U(3)
      U(9)=U(2)*U(4)
      U(10)=U(3)*U(4)

      RETURN
      END
      SUBROUTINE DOMORE(DATA,X,Y,BETA,ID,IPRIOR,Q,TABLE,NAM,RESID,
     &                  SCRTCH,NX1,NY1,NO1,NVREAD,QUIT)
      COMMON IPROB,NX,NY,NV,NO,KRIT,FORCE,ISTEP,IPRWHA,IPRSTA
      COMMON /FINEPT/ KONST,IWEIGH,KOMP,SIGSQ,SSQMIN,NAMFIL,DATFIL,
     &                FINCL,FDEL,BOMB,TITLE(16)
      COMMON /RESLTS/ RSS,RSQ,SSQ,MSE,MSB,PRESS,FREG,FVAR,P,DF,DFNUM,
     &                K,TOL
      INTEGER P, DF, DFNUM
      LOGICAL QUIT
      DIMENSION U(10)
      DATA IDOALL /0/

C     ON PROBLEM 1, JUST USE DEFAULT DOMORE.
C
C     ON PROBLEM 2, DO NOTHING UNTIL STEPWISE PROCEDURE SELECTS A "FINAL"
C     MODEL, BASED ON KRIT = 2("SSQ").
C     THEN SET SIGSQ = SSQMIN, SO MSE IS DEFINED
C          SET KRIT  = 4("PRESS")
C      AND SET QUIT  = .FALSE., SO STEPWISE PROCEDURE RESUMES.
C     WHEN PROCEDURE  AS AGAIN SELECTED A FINAL MODEL, CALL BBIAS AND
C     YBIAS (AT AN ARBITRARY POINT) AND THEN GO TO DEFAULT DOMORE.

      IF(IPROB.EQ.1) GO TO 200

      IF(.NOT.QUIT) RETURN
      IF(KRIT.NE."SSQ") GO TO 100

C     KRIT = "SSQ"  AND  QUIT = T

        KRIT  = "PRESS"
        SIGSQ = SSQMIN
        QUIT  = .FALSE.
        RETURN
```

```
  100 CONTINUE
C       KRIT = "PRESS"  AND  QUIT = T
          CALL BBIAS
          U(1)=1
          U(2)=70
          U(3)=20
          U(4)=80
          DO 150 I=2,4
  150       U(I+3)=U(I)**2
          U(8)=U(2)*U(3)
          U(9)=U(2)*U(4)
          U(10)=U(3)*U(4)
          CALL YBIAS(U)
C       DEFAULT DOMORE STARTS HERE
  200 CONTINUE
      IF(KRIT.EQ."ALL") IDOALL = 1
      IF(.NOT.QUIT) RETURN
      IF(IDOALL.EQ.0) CALL SEELOT(Y,ID,NAM,RESID,Q,SCRTCH,NX1,
     &                            NY1,NO1)
      IDOALL = 0
      RETURN
      END
*EOR
 $SETUP NX=10,NEWTITL=1,NAMFIL=5,KRIT=1,
 $
STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
80,27,89,42
80,27,88,37
75,25,90,37
62,24,87,28
62,22,87,18
62,23,87,18
62,24,93,19
62,24,93,20
58,23,87,15
58,18,80,14
58,18,89,14
58,17,88,13
58,18,82,11
58,19,93,12
50,18,89,8
50,18,86,7
50,19,72,8
50,19,79,8
50,20,80,9
56,20,82,15
70,20,91,15
*EOR
CONST
X1
X2
X3
X1**2
X2**2
X3**2
X1*X2
X1*X3
X2*X3
STACK LOSS
 $SETUP KRIT=2,FORCE=T, IPRIOR=4*2,6*1,
  DATFIL=0, NEWTITL=0,
  IWEIGH=T, WEIGHT=0,1,C,0,16*1,0,
 $
 $SETUP QUIT=T $
```

APPENDIX C

PARTIAL OUTPUT FROM SAMPLE JOB

****
****

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6 ←——— TITLE, ENTERED BY USER

PROBLEM NUMBER 1 ←——— IPROB

NUMBER OF INDEPENDENT VARIABLES IS 10 ←——— NX

NUMBER OF DATA CASES READ = 21 ←——— NO

PRINTED BECAUSE CHKDAT = T

CHECK SUMMARIES OF DATA VALUES READ FOR PLAUSIBILITY

VARIABLE	CASE	MIN. VALUE	CASE	MAX. VALUE	RANGE	MEAN	ST. DEVIATION
CONST	1	1.0000	1	1.0000	0.	1.0000	0.
X1	15	50.000	1	80.000	30.000	60.429	8.9473
X2	12	17.000	1	27.000	10.000	21.095	3.0846
X3	17	72.000	7	93.000	21.000	86.286	5.2294
X1**2	15	2500.0	1	6400.0	3900.0	3731.7	1154.[illegible]
X2**2	12	289.00	1	729.00	440.00	454.52	135.17
X3**2	17	5184.0	7	8649.0	3465.0	7472.6	877.46
X1*X2	15	900.00	1	2160.0	1260.0	1296.3	375.96
X1*X3	17	3600.0	1	7120.0	3520.0	5237.5	962.51
X2*X3	17	1368.0	1	2403.0	1035.0	1826.5	326.97
STACK LOSS	16	7.0000	1	42.000	35.000	17.524	9.9265

DATA CASES ←——— PRINTED BECAUSE LISDAT = T

CASE										
1	1.0000 42.000	80.000	27.000	89.000	6400.0	729.00	7921.0	2160.0	7120.0	2403.0
2	1.0000 37.000	80.000	27.000	88.000	6400.0	729.00	7744.0	2160.0	7040.0	2376.0
3	1.0000 37.000	75.000	25.000	90.000	5625.0	625.00	8100.0	1875.0	6750.0	2250.0
4	1.0000 28.000	62.000	24.000	87.000	3844.0	576.00	7569.0	1488.0	5394.0	2088.0
5	1.0000 18.000	62.000	22.000	87.000	3844.0	484.00	7569.0	1364.0	5394.0	1914.0
6	1.0000 18.000	62.000	23.000	87.000	3844.0	529.00	7569.0	1426.0	5394.0	2001.0
7	1.0000 19.000	62.000	24.000	93.000	3844.0	576.00	8649.0	1488.0	5766.0	2232.0
8	1.0000 20.000	62.000	24.000	93.000	3844.0	576.00	8649.0	1488.0	5766.0	2232.0
9	1.0000 15.000	58.000	23.000	87.000	3364.0	529.00	7559.0	1334.0	5046.0	2001.0
10	1.0000 14.000	58.000	18.000	80.000	3364.0	324.00	6400.0	1044.0	4640.0	1440.0
11	1.0000 14.000	58.000	18.000	89.000	3364.0	324.00	7921.0	1044.0	5162.0	1602.0
12	1.0000 13.000	58.000	17.000	88.000	3364.0	289.00	7744.0	986.00	5104.0	1496.0
13	1.0000 11.000	58.000	18.000	82.000	3364.0	324.00	6724.0	1044.0	4756.0	1476.0
14	1.0000 12.000	58.000	19.000	93.000	3364.0	361.00	8649.0	1102.0	5394.0	1767.0
15	1.0000 8.0000	50.000	18.000	89.000	2500.0	324.00	7921.0	900.00	4450.0	1602.0
16	1.0000 7.0000	50.000	18.000	86.000	2500.0	324.00	7396.0	900.00	4300.0	1548.0
17	1.0000 8.0000	50.000	19.000	72.000	2500.0	361.00	5184.0	950.00	3600.0	1368.0
18	1.0000 8.0000	50.000	19.000	79.000	2500.0	361.00	6241.0	950.00	3950.0	1501.0
19	1.0000 9.0000	50.000	20.000	80.000	2500.0	400.00	6400.0	1000.0	4000.0	1600.0
20	1.0000 15.000	56.000	20.000	82.000	3136.0	400.00	6724.0	1120.0	4592.0	1640.0
21	1.0000 15.000	70.000	20.000	91.000	4900.0	400.00	8281.0	1400.0	6370.0	1820.0

R A W   S U M S   O F   S Q U A R E S   A N D   C R O S S   T E R M S

F O R   I N P U T   V A R I A B L E S   (X ' S   A N D   Y ' S)

	CONST	X1	X2	X3	X1**2	X2**2	X3**2	X1*X2	X1*X3	X2*X3
CONST	21.000	1269.0	443.00	1812.0	78365.	9545.0	.15692E+06	27223.	.10999E+06	39357.
X1	1269.0	78365.	27223.	.10999E+06	.49518E+07	.59712E+06	.95647E+07	.17118E+07	.68213E+07	.23676E+07
X2	443.00	27223.	9545.0	38357.	.17118E+07	.21010E+06	.33326E+07	.59712E+06	.23676E+07	.82943E+06
X3	1812.0	.10999E+06	38357.	.15692E+06	.68213E+07	.82943E+06	.13637E+08	.23676E+07	.95647E+07	.33326E+07
X1**2	78365.	.49518E+07	.17118E+07	.68213E+07	.32044E+09	.38259E+08	.59552E+09	.11025E+09	.43275E+09	.14949E+09
X2**2	9545.0	.59712E+06	.21010E+06	.82943E+06	.38259E+08	.47221E+07	.72305E+08	.13387E+08	.52109E+08	.18322E+08
X3**2	.15692E+06	.95647E+07	.33326E+07	.13637E+08	.59552E+09	.72305E+08	.11888E+10	.20654E+09	.83434E+09	.29048E+09
X1*X2	27223.	.17118E+07	.59712E+06	.23676E+07	.11025E+09	.13387E+08	.20654E+09	.38259E+08	.14949E+09	.52109E+08
X1*X3	.10999E+06	.68213E+07	.23676E+07	.95647E+07	.43275E+09	.52109E+08	.83434E+09	.14949E+09	.59552E+09	.20654E+09
X2*X3	38357.	.23676E+07	.82943E+06	.33326E+07	.14949E+09	.18322E+08	.29048E+09	.52109E+08	.20654E+09	.72305E+08
STACK LOSS	368.00	23953.	8326.0	32189.	.15960E+07	.19244E+06	.28223E+07	.55220E+06	.21022E+07	.73047E+06

PRINTED BECAUSE IRAW = 3

	STACK LOSS
CONST	368.00
X1	23953.
X2	8326.0
X3	32189.
X1**2	.15960E+07
X2**2	.19244E+06
X3**2	.28223E+07
X1*X2	.55220E+06
X1*X3	.21022E+07
X2*X3	.73047E+06
STACK LOSS	8518.0

← THIS IS LAST COLUMN OF THE 11 x 11 MATRIX STARTED ABOVE. (ONLY ROOM FOR 10 COLUMNS ON A PAGE.)

COVARIANCE MATRIX

FOR INPUT VARIABLES (X'S AND Y'S)

	X1	X2	X3	X1**2	X2**2	X3**2	X1*X2	X1*X3	X2*X3	STACK LOSS
X1	80.054	21.578	23.401	10301.	967.82	3906.5	3179.9	8327.1	2367.2	81.680
X2	21.578	9.5147	6.3061	2795.1	416.24	1060.8	1087.6	2254.6	965.47	26.807
X3	23.401	6.3061	27.347	2833.6	277.52	4585.0	886.38	3539.9	1093.5	20.755
X1**2	10301.	2795.1	2833.6	.13335E+07	.12571E+06	.47293E+06	.41248E+06	.10625E+07	.30270E+06	10605.
X2**2	967.82	416.24	277.52	.12571E+06	18272.	46638.	48275.	.10079E+06	42273.	1198.7
X3**2	3906.5	1060.8	4585.0	.47293E+06	46638.	.76994E+05	.14837E+06	.59266E+06	.18378E+06	3447.6
X1*X2	3179.9	1087.6	886.38	.41248E+06	48275.	.14837E+06	.14136E+06	.32910E+06	.11358E+06	3578.5
X1*X3	8327.1	2254.6	3539.9	.10625E+07	.10079E+06	.59266E+06	.32910E+06	.92642E+06	.26885E+06	8323.6
X2*X3	2367.2	965.47	1093.5	.30270E+06	42273.	.18378E+06	.11358E+06	.26885E+06	.10691E+06	2776.4
STACK LOSS	81.680	26.807	20.755	10605.	1198.7	3447.6	3578.5	8323.6	2776.4	98.535

PRINTED BECAUSE ICOVCOR = 9

CORRELATION MATRIX

FOR INPUT VARIABLES (X'S AND Y'S)

	X1	X2	X3	X1**2	X2**2	X3**2	X1*X2	X1*X3	X2*X3	STACK LOSS
X1	1.0000	.7819	.5001	.9970	.8002	.4976	.9453	.9669	.8092	.9197
X2	.7819	1.0000	.3909	.7847	.9983	.3919	.9378	.7594	.9573	.8755
X3	.5001	.3909	1.0000	.4692	.3926	.9992	.4508	.7033	.6395	.3998
X1**2	.9970	.7847	.4692	1.0000	.8053	.4667	.9500	.9560	.8017	.9251
X2**2	.8002	.9983	.3926	.8053	1.0000	.3932	.9499	.7746	.9564	.8934
X3**2	.4976	.3919	.9992	.4667	.3932	1.0000	.4497	.7017	.6406	.3958
X1*X2	.9453	.9378	.4508	.9500	.9499	.4497	1.0000	.9094	.9239	.9588
X1*X3	.9669	.7594	.7033	.9560	.7746	.7017	.9094	1.0000	.8543	.8712
X2*X3	.8092	.9573	.6395	.8017	.9564	.6406	.9239	.8543	1.0000	.8554
STACK LOSS	.9197	.8755	.3998	.9251	.8934	.3958	.9588	.8712	.8554	1.0000

THE CONSTANT IS FORCED IN, BUT NO OTHER VARIABLES ARE FORCED ← *BECAUSE KONST = T AND FORCE = F*

AT EACH STEP, VARIABLE IS ENTERED OR DELETED BASED ON F
F TO INCLUDE = .1000 F TO DELETE = .0500

FROM STEP 1 ON, R SQUARED AND F FOR REGRESSION ARE BASED ON SUM OF SQUARES ABOUT MEAN ("CORRECTED SUM OF SQUARES") ← *BECAUSE KONST = T*

0***

MODEL=00000 00000

RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR REGRESSION	DF1,DF2	PRESS/NO (=M.S.PR.ER.)
8518.00	405.619	0.	0.	0, 21	405.619

EVEN THOUGH IT WILL NOT PRODUCE LARGEST F, THE CONSTANT IS ENTERED FIRST, BECAUSE KONST = T.

EFFECT OF CHANGING EACH VARIABLE (STAR MARKS VARIABLE IN MODEL)

VARIABLE	FORCING PRIORITY	TOLERANCE	RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR THE VARIABLE	PRESS/NO (=M.S.PR.ER.)
CONST	10	1.000000	2069.24	103.462	-	62.3298	108.635
X1	1	1.000000	1196.54	59.8270	.859528	122.377	64.7692
X2	1	1.000000	1255.32	62.7660	.852627	115.710	67.7296
X3	1	1.000000	1915.24	95.7619	.775154	68.9498	100.851
X1**2	1	1.000000	569.243	28.4622	.933172	279.274	31.5270
X2**2	1	1.000000	675.707	33.7853	.920673	232.121	37.3646
X3**2	1	1.000000	1817.59	90.8797	.786617	73.7283	96.1321
X1*X2	1	1.000000	547.928	27.3964	.935674	290.917	30.2807
X1*X3	1	1.000000	1097.15	54.8575	.871196	135.275	59.7181
X2*X3	1	1.000000	1138.45	56.9225	.866348	129.642	61.6974

1***

CONST HAS JUST BEEN ENTERED

MODEL=10000 00000

RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR REGRESSION	DF1,DF2	PRESS/NO (=M.S.PR.ER.)
2069.24	103.462	0.	0.	0, 20	108.635

VARIABLE	BETA	ST DEV	T
CONST	17.5238	2.21963	7.895

EFFECT OF CHANGING EACH VARIABLE (STAR MARKS VARIABLE IN MODEL)

VARIABLE	FORCING PRIORITY	TOLERANCE	RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR THE VARIABLE	PRESS/NO (=M.S.PR.ER.)
* CONST	10	*	* 8518.00	* 405.619	-	* 62.3298	* 405.619
X1	1	1.000000	319.116	16.7956	.845781	104.201	18.9942
X2	1	1.000000	483.151	25.4290	.766508	62.3732	29.4328
X3	1	1.000000	1738.44	91.4969	.159864	3.61538	95.1979
X1**2	1	1.000000	298.218	15.6957	.855880	112.835	17.4679
X2**2	1	1.000000	417.740	21.9863	.798119	75.1149	25.1677
X3**2	1	1.000000	1745.06	91.8452	.156666	3.52964	95.6201
X1*X2	1	1.000000	166.846	8.78135	.919369	216.640	9.84942
X1*X3	1	1.000000	498.763	26.2507	.758963	59.8260	30.6557
X2*X3	1	1.000000	555.116	29.2166	.731729	51.8240	34.4518

2***

X1*X2 HAS JUST BEEN ENTERED

MODEL=10000 00100

RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR REGRESSION	DF1,DF2	PRESS/NO (=M.S.PR.ER.)
166.846	8.78135	.919369	216.640	1, 19	9.84942

2**

X1*X2 HAS JUST BEEN ENTERED

MODEL=10000 00100

RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR REGRESSION	DF1,DF2	PRESS/NO (=M.S.PR.ER.)
166.846	8.78135	.919369	216.640	1, 19	9.84942

CASES WITH HIGH AND LOW LEVERAGE MIN POSSIBLE = 0, AVE. = .09524, MAX POSSIBLE = 1
HIGH 1 .29890 2 .29890

VARIABLE	BETA	ST DEV	T
CONST	-15.2931	2.32149	-6.588
X1*X2	.253152E-01	.171993E-02	14.719

*LARGEST VALUE. THEREFORE X2*X3 GOES IN NEXT.*

EFFECT OF CHANGING EACH VARIABLE (STAR MARKS VARIABLE IN MODEL)

VARIABLE	FORCING PRIORITY	TOLERANCE	RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR THE VARIABLE	PRESS/NO (=M.S.PR.ER.)
* CONST	10	*	* 547.928	* 27.3964	-	* 43.3968	* 30.2807
X1	1	.106440	163.411	9.07842	.921028	.378288	12.4005
X2	1	.120548	157.217	8.73428	.924022	1.10241	11.1056
X3	1	.796758	164.113	9.11740	.920689	.299697	10.4429
X1**2	1	.097414	162.563	9.03128	.921438	.474214	12.4291
X2**2	1	.097709	160.423	8.91241	.922472	.720616	11.7995
X3**2	1	.797729	163.591	9.08840	.920941	.358104	10.4043
* X1*X2	1	*	* 2069.24	* 103.462	*0.	* 216.640	* 108.635
X1*X3	1	.172964	166.838	9.26879	.919372	.818383E-03	11.4278
X2*X3	1	.146369	153.713	8.53962	.925715	1.53785	10.0549

PRINTED BECAUSE IPRWHA = 1

3**

ISTEP = 3

X2*X3 HAS JUST BEEN ENTERED

MODEL=10000 00101 *← MODEL CONTAINS VARIABLES 1, 8 AND 10*

PRINTED BECAUSE IPRSTA = 2

RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR REGRESSION	DF1,DF2	PRESS/NO (=M.S.PR.ER.)
153.713	8.53962	.925715	112.155	2, 18	10.0549

CASES WITH HIGH AND LOW LEVERAGE MIN POSSIBLE = 0, AVE. = .14286, MAX POSSIBLE = 1
HIGH 1 .34090 2 .36242
LOW 5 .05250 6 .06832 11 .07051 20 .06421

VARIABLE	BETA	ST DEV	T
CONST	-10.3311	4.60992	-2.241
X1*X2	.303946E-01	.443328E-02	6.856
X2*X3	-.632166E-02	.509771E-02	-1.240

INCLUDED VARIABLES HAVE F > FDEL. REMAINING VARIABLES HAVE F < FINCL. THEREFORE STOP.

EFFECT OF CHANGING EACH VARIABLE (STAR MARKS VARIABLE IN MODEL)

VARIABLE	FORCING PRIORITY	TOLERANCE	RESID SS	S SQUARED (=RSS/DF)	R SQUARED	F FOR THE VARIABLE	PRESS/NO (=M.S.PR.ER.)
* CONST	10	*	* 196.602	* 10.3475	-	* 5.02232	* 11.7996
X1	1	.033182	153.713	9.04194	.925715	.183946E-04	12.6922
X2	1	.064196	152.979	8.99876	.926070	.815969E-01	11.5237
X3	1	.083952	152.826	8.98974	.926144	.987217E-01	11.4508
X1**2	1	.023702	153.616	9.03626	.925762	.107048E-01	12.8956
X2**2	1	.055261	153.676	9.03974	.925733	.415148E-02	12.2639
X3**2	1	.082883	153.114	9.00669	.926005	.665516E-01	11.5450
* X1*X2	1	*	* 555.116	* 29.2166	* .731729	* 47.0048	* 34.4518
X1*X3	1	.092965	153.658	9.03873	.925742	.605626E-02	12.3828
* X2*X3	1	*	* 166.846	* 8.78135	* .919369	* 1.53785	* 9.84942

THIS IS THE FINAL MODEL

C O V A R I A N C E   M A T R I X

F O R   B E T A S   I N   M O D E L

	CONST	X1*X2	X2*X3
CONST	21.251	.12660E-01	-.20397E-01
X1*X2	.12660E-01	.19654E-04	-.20880E-04
X2*X3	-.20397E-01	-.20880E-04	.25987E-04

← THIS, AND THE PLOTS BELOW, ALL PRODUCED BY SUBROUTINE SEELOT.

C O R R E L A T I O N   M A T R I X

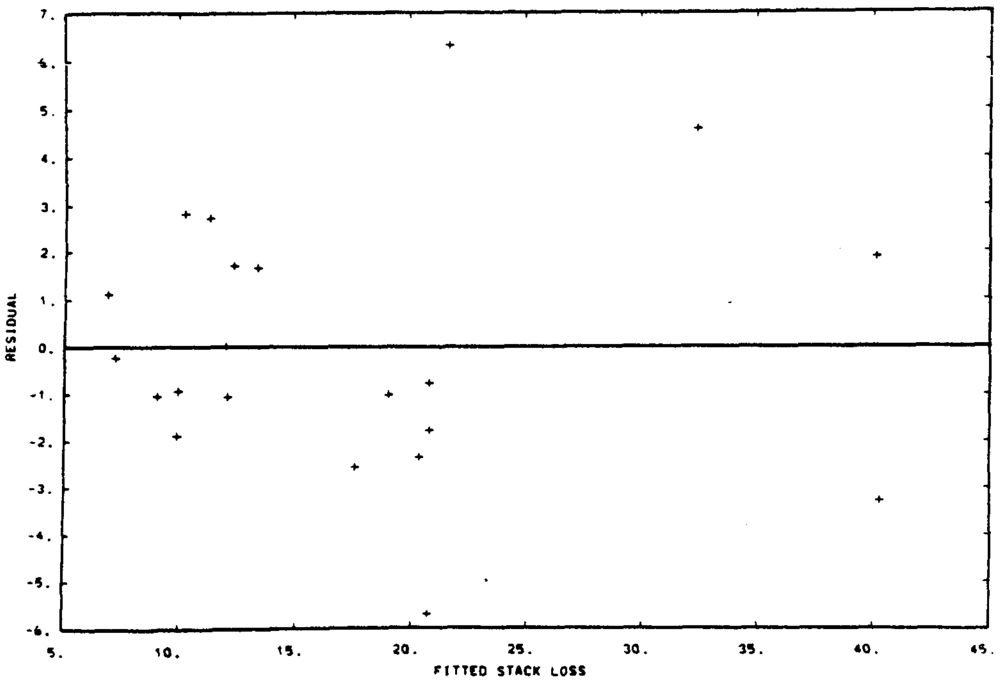
F O R   B E T A S   I N   M O D E L

	CONST	X1*X2	X2*X3
CONST	1.0000	.6195	-.8680
X1*X2	.6195	1.0000	-.9239
X2*X3	-.8680	-.9239	1.0000

CASE	OBSERVED STACK LOSS	FITTED VALUE	RESIDUAL	STUDENTIZED RESIDUAL	LEVERAGE
1	.42000E+02	.40130E+02	.18697E+01	.779	.341
2	.37000E+02	.40301E+02	-.33010E+01	-1.458	.362
3	.37000E+02	.32435E+02	.45649E+01	1.816	.166
4	.28000E+02	.21696E+02	.63035E+01	2.607	.095
5	.18000E+02	.19028E+02	-.10275E+01	-.352	.052
6	.18000E+02	.20362E+02	-.23620E+01	-.830	.068
7	.19000E+02	.20786E+02	-.17862E+01	-.697	.252
8	.20000E+02	.20786E+02	-.78616E+00	-.303	.252
9	.15000E+02	.17566E+02	-.25657E+01	-.928	.111
10	.14000E+02	.12298E+02	.17023E+01	.629	.172
11	.14000E+02	.11274E+02	.27264E+01	.966	.071
12	.13000E+02	.10181E+02	.28192E+01	1.018	.100
13	.11000E+02	.12070E+02	-.10701E+01	-.384	.136
14	.12000E+02	.11993E+02	.65881E-02	.002	.089
15	.80000E+01	.68968E+01	.11032E+01	.395	.127
16	.70000E+01	.72381E+01	-.23814E+00	-.084	.105
17	.80000E+01	.98958E+01	-.18958E+01	-.709	.187
18	.80000E+01	.90550E+01	-.10550E+01	-.370	.095
19	.90000E+01	.99489E+01	-.94888E+00	-.330	.078
20	.15000E+02	.13343E+02	.16566E+01	.575	.064
21	.15000E+02	.20716E+02	-.57160E+01	-2.253	.076

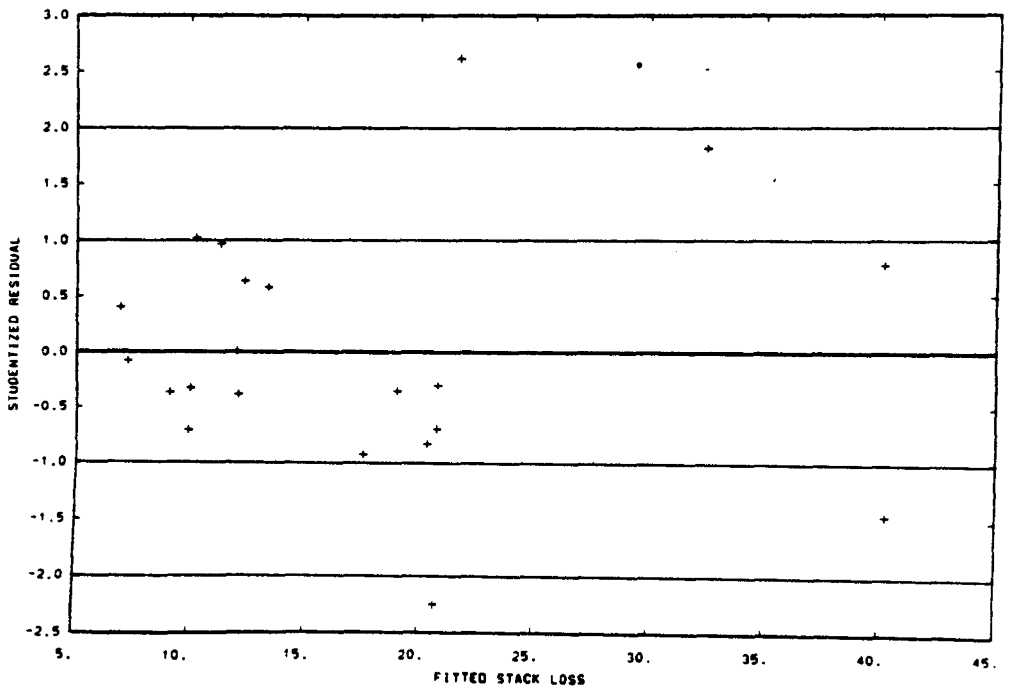
THIS CONCLUDES PROBLEM 1

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3

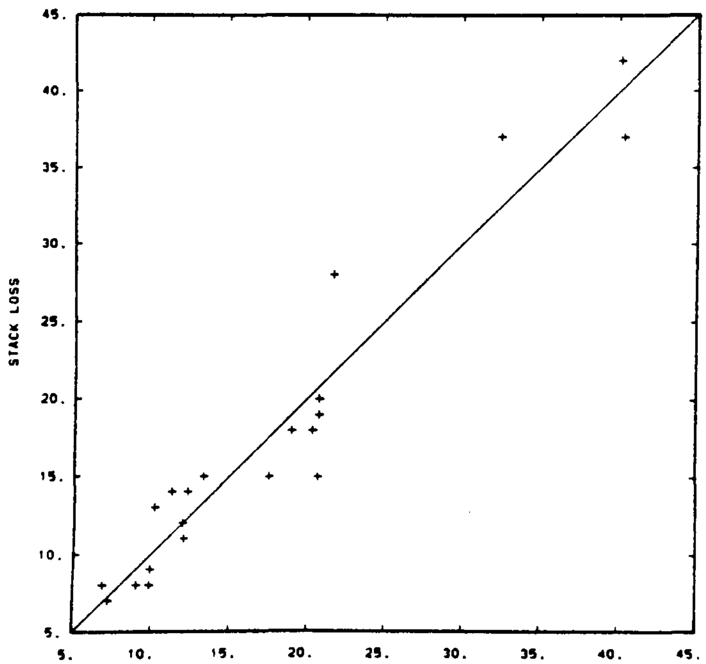


FITTED VS. RESIDUALS

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3

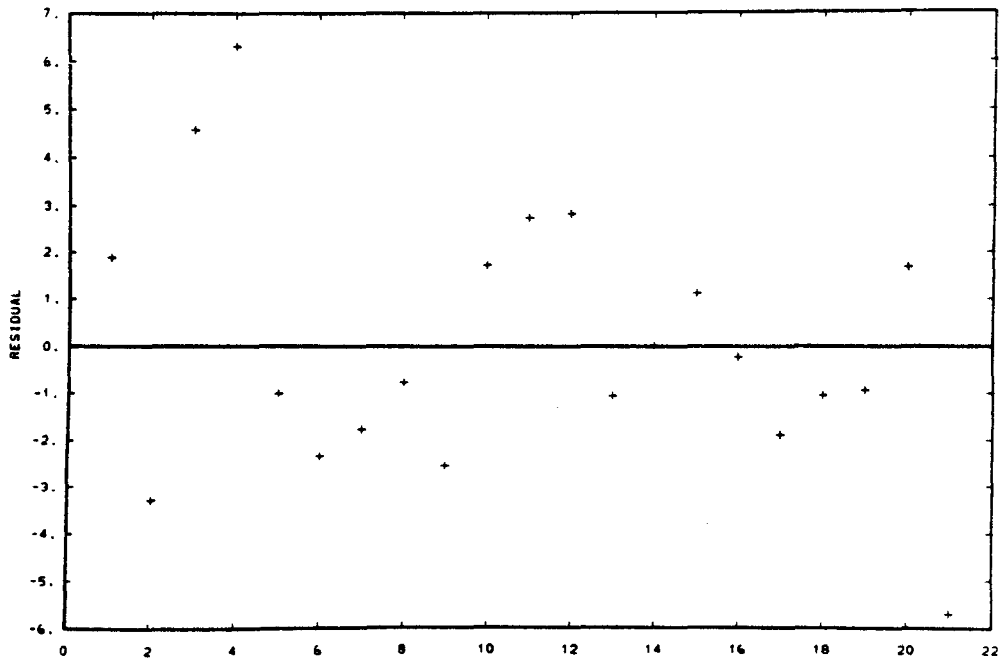


FITTED VS. STUDENTIZED RESIDUALS



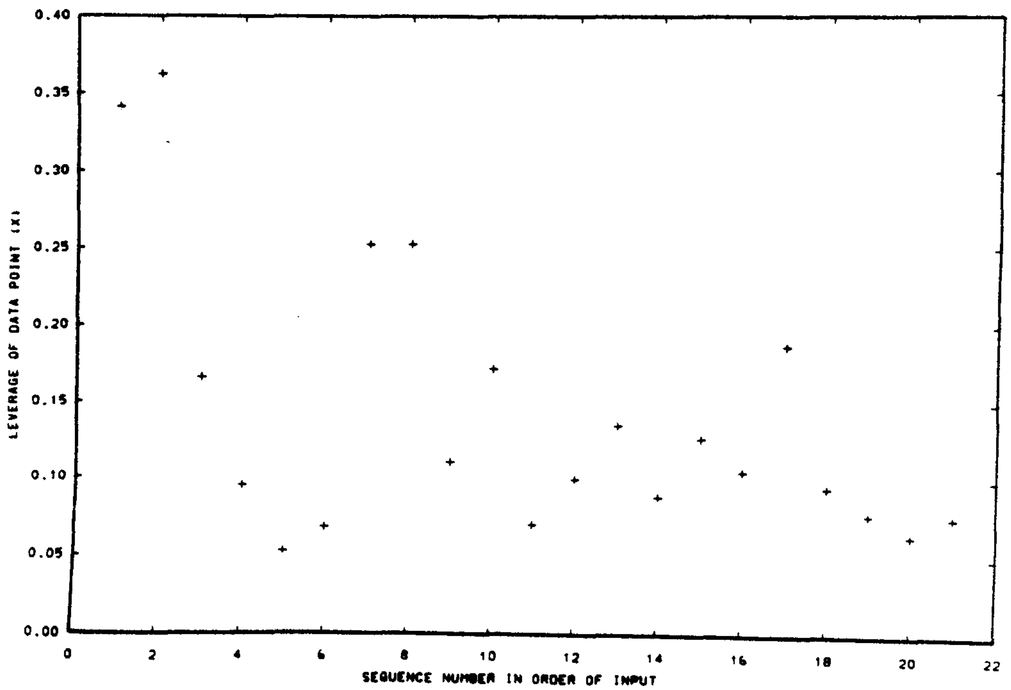
FITTED VS. OBSERVED

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



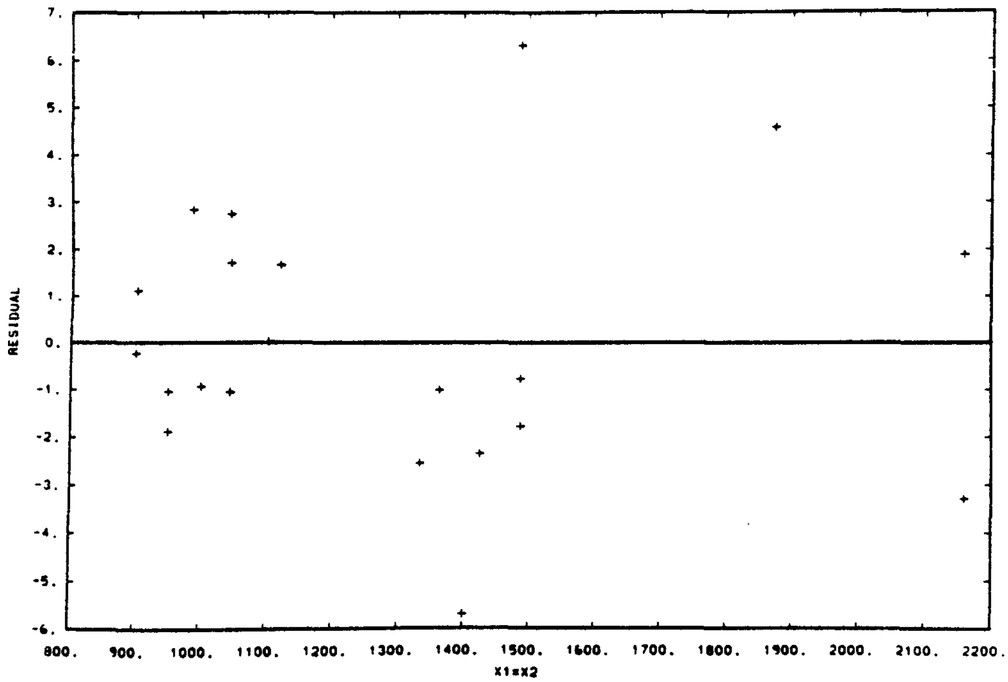
RESIDUALS IN ORDER OF DATA INPUT

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



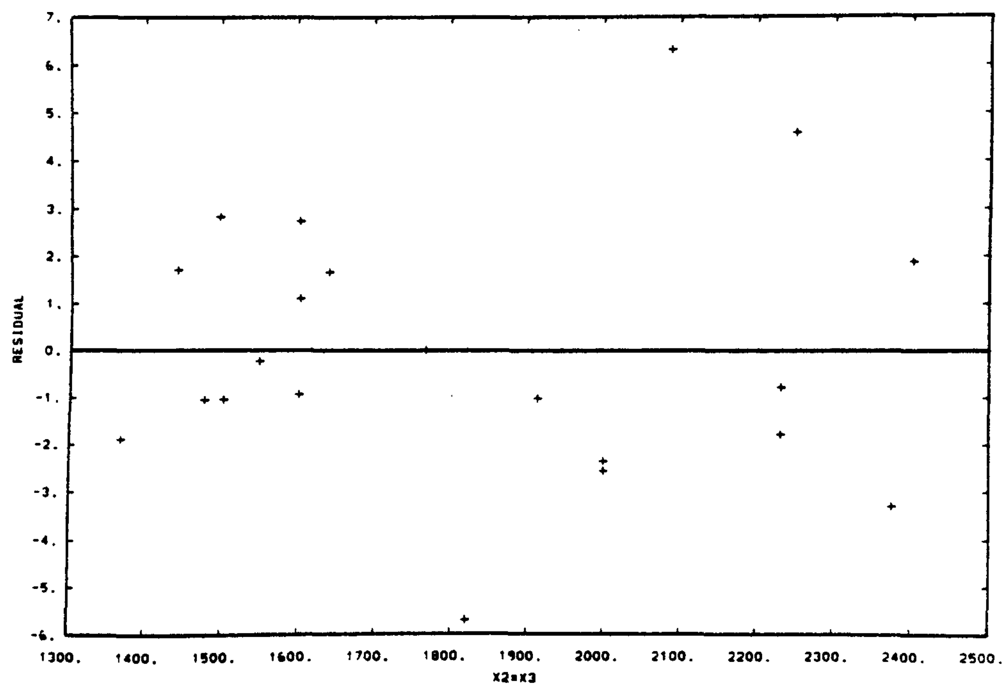
LEVERAGE OF DATA CASES, IN ORDER OF DATA INPUT

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



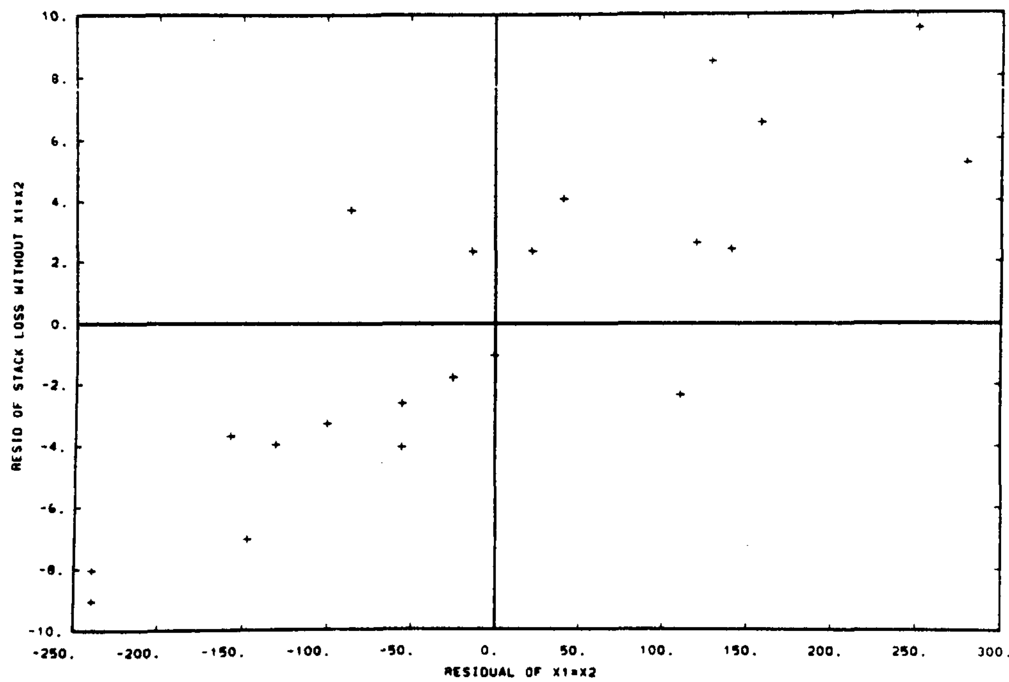
X1*X2 VS. RESIDUALS

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



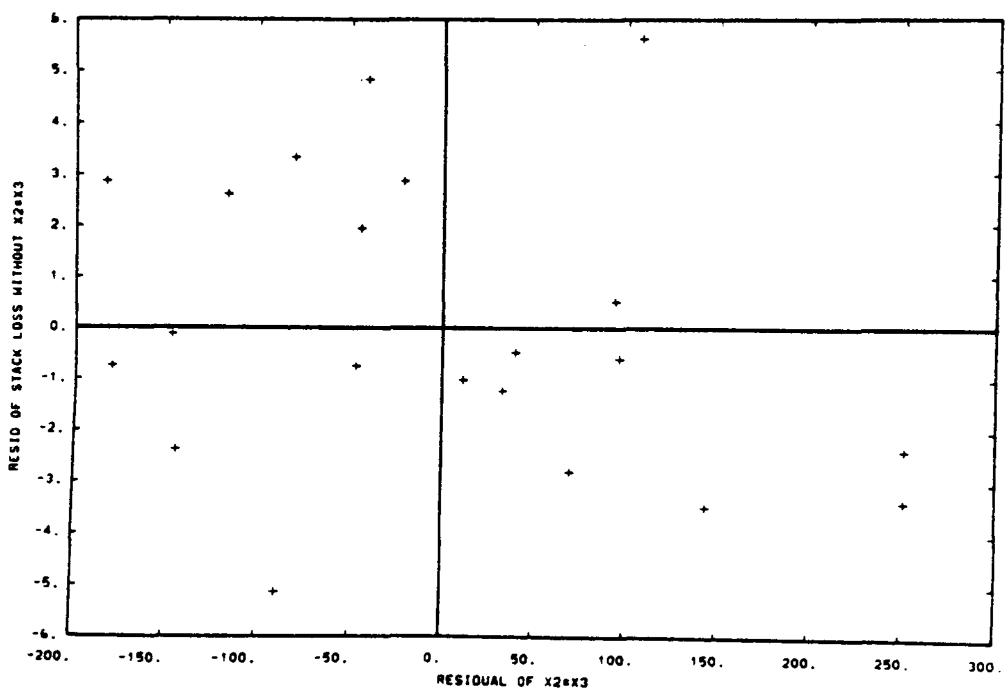
X2*X3 VS. RESIDUALS

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



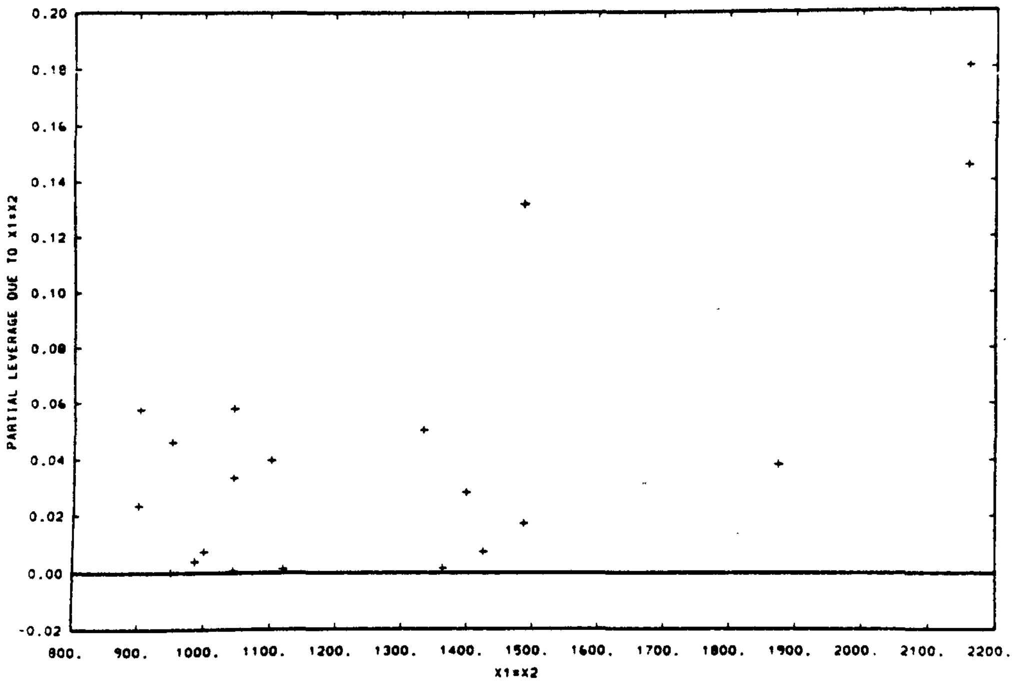
PARTIAL REGRESSION PLOT, REMOVING X1*X2

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



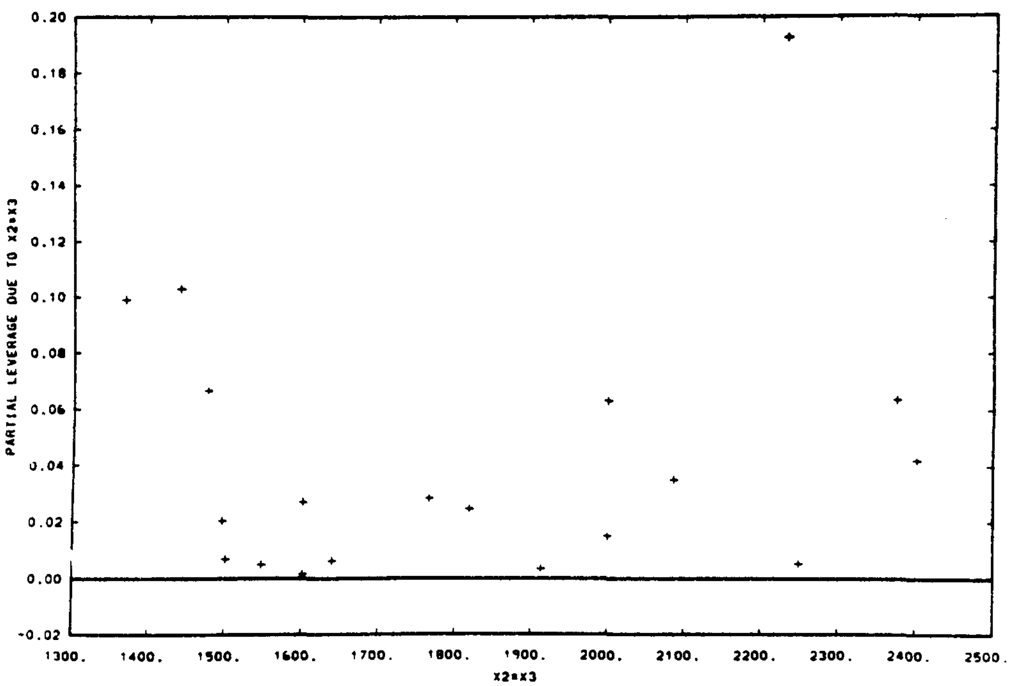
PARTIAL REGRESSION PLOT, REMOVING X2*X3

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



OBSERVED X1*X2 VS. PARTIAL LEVERAGE DUE TO X1*X2

STACK LOSS PROBLEM, FROM DRAPER AND SMITH, CH. 6
PROBLEM 1, STEP 3



OBSERVED X2*X3 VS. PARTIAL LEVERAGE DUE TO X2*X3