JANUARY 1978

VOLUME 8 NUMBER 2

MULTIPLE LINEAR REGRESSION VIEWPOINTS

A publication of the Special Interest Group on Multiple Linear Regression

MULTIPLE LINEAR REGRESSION VIEWPOINTS

Chairman .... Steve Spaner, Behavioral Studies, University of Missouri, St. Louis, MO 63121

Editor .... Isadore Newman, Research and Design Consultant, The University of Akron, Akron, OH 44325

Assistant .... Diane Vukovich The University of Akron, Akron, OH 44325

Executive Secretary .... Steve Spaner, Behavioral Studies University of Missouri, St. Louis, MO 63121

Secretary and Chairman-elect .... Dr. John Williams University of North Dakota, Grand Forks, ND 58201

Cover by .... David G. Barr

TABLE OF CONTENTS

Multiple Linear Regression Viewpoints
Volume 8 Number 2

AN INTERACTIVE APPROACH TO RIDGE REGRESSION

J. F. Marquette
M. M. Dufala
The University of Akron

A recurring problem in practical applications of ordinary least squares is the existence of multi-collinearity in the sample of predictor variables. The consequent ill conditioning of the correlation matrix results in large standard errors for the B_i. While these B_i are best linear unbiased estimators (BLUE) for β_i given a particular estimation sample, the large standard errors reduce the utility of the regression equation for predictive purposes in future samples. In practice, even small changes in the distribution of the predictor variables can result in extremely large fluctuations in the predicted values of the criterion variable. If the researcher has sound theoretical reasons for the original choice of predictor variables he may be extremely reluctant to combat the problem by deleting some variables from the data set.

Recent research on ridge regression suggests that relaxation of the unbiasedness criterion will reduce the standard error of the B_i for an estimation sample and stabilize the predicted values of the criterion variable in future samples.[1,2,3] This approach suggests estimating β^* instead of β, using

$$B = (X'X + KI)^{-1}X'Y$$

as the appropriate estimate of β^*.

In this formulation B is no longer BLUE for β, the error SS of the estimation sample is an increasing function of the ridge value K, as is the distance between β and β^*. Hopefully, however, the variability of the predicted values of the criterion value in future samples will be decreased by the choice of an appropriate K. A major problem of this approach is the lack of any rigorous basis for the choice of K.

The suggested approach to choosing K is to start with a small value of K and explore the effect of increasing incremental values on the resulting B_i. The K value chosen is that at which further increment to K produces only minimal changes in the B_i.

Since at this stage the choice of K is an essentially exploratory process, ridge regression would appear to be a prime candidate for an interactive programming approach. The University of Akron's APLSV based ADEPT system provides such a facility.[4]

The following is a brief examination of an interactive analysis of a ridge problem, using the Hald concrete data as reported in Draper and Smith.[5]

Example 1 is a reproduction of the user's initial interaction with the system, specifying the data set and variables to be used. The later portion of the example shows the user's choice of regression options, indicating that the ridge trace is to be plotted, that the base value of K is to be 0 with additional increments of .02 for 6 iterations. Figure 1 lists the output resulting from the above specifications. The initial output presents, for each value of K, the determinant of the augmented X'X matrix, the maximum variance inflation factor (which is the largest element of the diagonal of $X'X^{-1}$), the multiple correlation, error mean square, intercept and the B values for each variable. Since the base value of K was set to 0, the first column of the output presents the ordinary leasts squares results. The maximum variance inflation factor here is 282.52, indicating very high multicollinearity.

The plot of the ridge trace in Figure 1 indicates that there is a very rapid alteration in the B values in the interval K=0 to K=.04 with subsequent relative stability thereafter. The interval of interest for this particular analysis would therefore appear to be K=0 to .04. Resetting the relevant options results in the output shown in Figure 2. The base value of K is still 0, but the increments are now .005. Examination of the ridge trace of Figure 2 shows that even the small K increment from 0 to .005 results in a large shift in the calculated B values, with the B for variable V3 showing a shift in sign as well as magnitude.

Had any of the B_i gone rapidly to 0, the researcher could have followed the suggested course of deleting that particular variable and respecified the analysis requests. It should be noted that the total real time for this "analysis" was approximately ten minutes, which is indicative of the great utility of an interactive approach to statistical computing.

EXAMPLE 1

```
  :OPTIONS? SB

SB:OPTIONS? RB
SB:RETRIEVE BASE REQUESTED
SB:ENTER THE NAME OF THE SAVED DATA BASE: SHALD
SB:USING DATA BASE: HALD
SB:BASE VARIABLES:
V1        V2        V3        V4        V5

  :OPTIONS? R
R :REGRESSION SPECIFICATION

R :OPTIONS? RR
R :SELECT VARIABLES
R : ALL
R :RR-RIDGE REGRESSION REQUESTED
R :SELECT DEPENDENT VARIABLE(S)
R : 5
R :SELECT INDEPENDENT VARIABLE(S)
R : 1 2 3 4
R :RIDGE PLOT? (Y/N) Y
R :ENTER BASE AND  INCREMENT: 0 .02
R :ENTER NUMBER OF ITERATIONS: 6
```

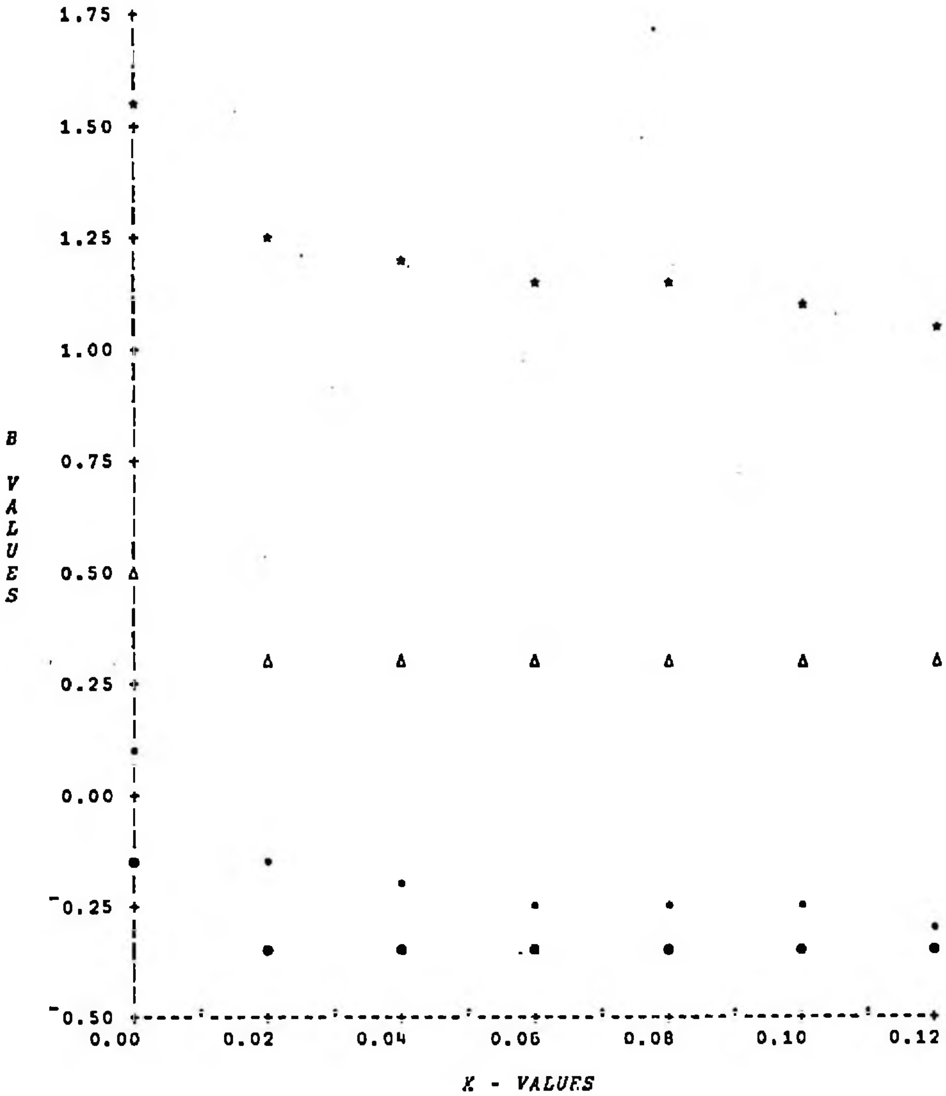
FIGURE 1

RIDGE REGRESSION RESULTS

K	.00000	.02000	.04000	.06000	.08000
DET	.00107	.01608	.03469	.05708	.08345
MAXVIF	282.51286	21.62449	11.42904	7.84179	6.00657
RSQ	.98238	.97194	.96216	.95275	.94366
MSE	5.98295	9.52660	12.84620	16.03903	19.12473
CONST	62.40537	84.35956	85.51446	86.08023	86.46805
*V1	1.55110	1.27227	1.21511	1.17043	1.13260
ΔV2	.51017	.29319	.28868	.28867	.28929
•V3	.10191	-.16305	-.20287	-.23119	-.25354
●V4	-.14406	-.35429	-.35571	-.35234	-.34808

K	.10000	.12000
DET	.11402	.14899
MAXVIF	4.88978	4.13751
RSQ	.93485	.92628
MSE	22.11702	25.02621
CONST	86.77016	87.02083
*V1	1.09962	1.07040
ΔV2	.28985	.29018
•V3	-.27175	-.28684
●V4	-.34370	-.33939

PLOT OF RIDGE TRACE



:OPTIONS?

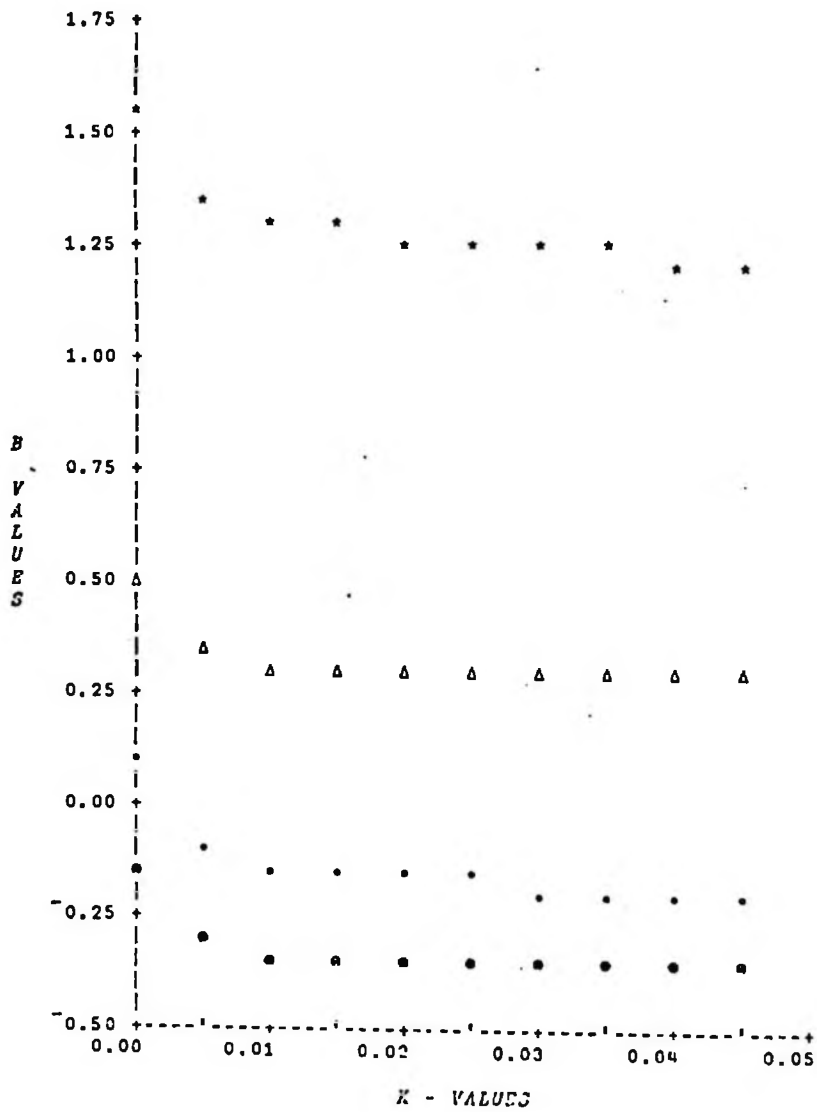
FIGURE 2

RIDGE REGRESSION RESULTS

K	.00000	.00500	.01000	.01500	.02000
DET	.00107	.00450	.00814	.01200	.01608
MAXVIF	282.51286	69.60256	39.85574	27.99987	21.62449
RSQ	.98238	.97959	.97700	.97445	.97194
MSE	5.98295	6.92819	7.80856	8.67330	9.52660
CONST	62.40537	80.12058	82.67556	83.74616	84.35956
*V1	1.55110	1.35513	1.31521	1.29111	1.27227
ΔV2	.51017	.33003	.30612	.29736	.29319
•V3	.10191	-.09330	-.12902	-.14866	-.16305
●V4	-.14406	-.32010	-.34294	-.35088	-.35429

K	.02500	.03000	.03500	.04000	.04500
DET	.02039	.02493	.02969	.03469	.03992
MAXVIF	17.64197	14.91740	12.93563	11.42904	10.24479
RSQ	.96945	.96700	.96456	.96216	.95977
MSE	10.36986	11.20386	12.02915	12.84620	13.65538
CONST	84.77152	85.07618	85.31638	85.51446	85.68323
*V1	1.25601	1.24134	1.22780	1.21511	1.20312
ΔV2	.29096	.28972	.28904	.28868	.28853
•V3	-.17488	-.18518	-.19442	-.20287	-.21068
●V4	-.35577	-.35624	-.35615	-.35571	-.35505

PLOT OF RIDGE TRACE



:OPTIONS?

REFERENCES

1) Hoerl, A. E. and Kennard, R. W. "Ridge Regression: Biased Estimation for Non-Orthogonal Problems." Technometrics 12, 1970, 55-67.

2) ____________________. "Ridge Regression: Applications to Non-Orthogonal Problems." Technometrics 12, 1970, 69-82.

3) Marquardt, D. W. and Snee, R. D. "Ridge Regression in Practice." American Statistician, 29-1, 1975, 3-20.

4) Marquette, J. F. and Dufala, M. M. ADEPT - Advanced Data Enquiry Package Time-Shared. A proprietary program product of The University of Akron, Akron, Ohio.

5) Draper, N. R. and Smith, H. Applied Regression Analysis, John Wiley & Sons, (New York: 1966).

Multiple Linear Regression Viewpoints
Volume 8 Number 2

Ridge Regression: A Panacea?

Joseph M. Walton
Associate Dean of Graduate Studies and Research

Isadore Newman
The University of Akron

John W. Fraas
Director of the Center for Economic
Ashland College

ABSTRACT

The present paper investigates the use of ridge regression and concludes that, while it may be an appropriate technique for some analyses, it may not be useful in instances where shrinkage estimates produce little shrinkage, or where the proportion of subjects to variables is sufficient.

Introduction

The increased use of multiple regression analysis in the social and physical sciences has caused researchers to examine methods of obtaining the most stable regression coefficients, since one of the basic objectives of multiple regression analysis is prediction. The method of ridge regression was introduced as a statistical procedure which may be utilized in regression analyses that are complicated by the problem of multicollinearity, which causes a fluctuation of regression weights from one sample to another (Hoerl, 1962).

The concept of ridge regression was further delineated by Hoerl and his associates in articles dealing with biased estimation for non-orthogonal problems and the application of ridge analysis to non-orthogonal problems (Hoerl and Kennard, 1970b; Hoerl and Kennard, 1970a). Marquardt and Snee later discussed the use of biased estimation and model building, and concluded that "when the predictor variables are highly correlated, ridge regression produces coefficients which predict and extrapolate better than least squares, and is a safe procedure for selecting variables" (Marquardt and Snee, 1975).

The method of ridge analysis requires that a constant be repeatedly added to the diagonal of the $X'X$ matrix (where the X variables are scaled, so that $X'X$ has the form of a correlation matrix) before the matrix is inverted (Newman and Fraas, 1977). In contrast to the standard regression model, ridge regression is an estimation procedure based upon

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

$$I = \text{identity matrix}$$

$$K = 0 \leq K < 1$$

$$\hat{\beta}^* = \text{ridge estimator of } \beta$$

where K is a constant number added to the identity matrix I (Newman and Fraas, 1977). Hoerl and Kennard described two important aspects of ridge regression: (1) the ridge trace, which generally is represented by a two dimensional plot of the coefficient weights vs. the K values, and (2) the determination of a value of K that gives a minimum mean square error [MSE = variance of the coefficient + $(\text{bias})^2$] which produces more stable beta weights (Hoerl and Kennard, 1970b). In using the ridge technique, one accepts some bias in the expected value of the coefficient in return for a lower mean square error (MSE). As Newman and Fraas point out, "the objective of ridge regression is to find a value of K which gives a set of coefficients with a smaller MSE than the one produced by the least squares solution" (Newman and Fraas, 1977). The residual sum of squares will increase as the K value increases. It is important to remember that the purpose of ridge regression is not to obtain "best fit" for the sample, but to develop stable coefficients (Marquardt and Snee, 1975).

Hence, ridge regression presents itself as a method designed to increase the researcher's ability to predict by producing more stable weights from sample to sample, particularly where the independent variables are non-orthogonal. In addition, as reported by Marquardt and Snee, ridge regression has the advantage of producing the ridge trace, which may aid the researcher in identifying the specific coefficients that are sensitive to the data. It is also an easy statistic to compute (Marquardt and Snee, 1975).

The major point of this paper is that the production of more stable weights per se does not cause the prediction to be more accurate. A prediction is only made more accurate by the production of a larger R^2. Therefore, ridge regression may be a misleading technique in the sense that, while it may in fact produce more stable weights, it does not necessarily yield greater accuracy. What the authors believe one must look at are not only the stability weights and stability of the R^2 but the purpose of the model structure. Ridge regression may not in fact be a panacea to the problem of non-orthogonality. The present researchers attempt to document this point of view in the following example.

Data Presentation

In the present study, the researchers attempted to predict counselor practicum ratings among 93 counselor education students, while using as predictors, simple, squared, and interaction variables which included undergraduate grade point average, final graduate grade point average, Miller Analogies Test Scores, and type of undergraduate institution. The data used in the study presented an excellent example of non-orthogonality.

The stability of the multiple linear regression procedure and the ridge procedure for the data were analyzed by two methods: a traditional cross-validation procedure, and ridge analysis. For both analyses, the 93 subjects were randomly divided into two groups of 47 (group A) and 46 (group B) respectively. The results are listed below in Table 1.

Table 1

Cross-Validation Results for the Multiple Regression Analysis and the Ridge Regression Analysis

Group	Multiple Regression R^2 - Value	Ridge Regression R^2 - Value	Cross-Validation with Multiple Regression Coefficient	Cross-Validation with Ridge Regression Coefficient
Group A	.39798	.32800	-	-
Group B	-	-	.29854	.28250
True Population Estimate	.29681	-	-	-

As one can see by inspection of Table 1 the ridge regression shrunk less (traditional shrinkage .09944; ridge shrinkage .04550). This is consistent with what one would expect. The ridge weights are more stable; therefore there should be less shrinkage.

We also know that the traditional multiple R^2 is upward biased. The difference between the sample multiple R^2 and the population value is .10117, while the difference between the ridge sample R^2 and the population is .03119. The ridge R^2 was more similar to the population R^2. However, the shrunken traditional R^2 was more similar to the population R^2 (the difference being .00173), than was either the shrunken or non-shrunken ridge R^2 (.01431 and .03119, respectively).

Obviously this is one sample and no strong generalizations should be made. However, <u>assuming</u> that the data here are representative of what exists, it would seem that the shrunken traditional R would produce the most accurate estimate of the population R and that ridge regression is more likely to produce the more stable weights.

The researcher should keep in mind the purpose of his research when deciding whether ridge regression or multiple linear regression is more appropriate. If the inappropriate statistical tool is used by the researcher, he or she is committing a Type VI Error (Newman, 1976 et al.). That is, there is an inconsistency between the statistical model and the research hypothesis. If the research project requires stable coefficients, as would be the case in making a point prediction over different samples, ridge regression may be the appropriate analytical tool. However, if the purpose of the research project is to test a hypothesis, the use of multiple linear regression would be more appropriate. If ridge regression were used to test a hypothesis, the researcher should remember that a bias is introduced into the analysis processes.

As with most statistical procedures, there is no one best technique and there is no substitute for knowing what research question one is interested in and what technique or models best reflect the question.

Discussion

The R^2 in a traditional regression model tends to be biased upward, that is, the regression equation tends to overestimate the R^2. Consequently, when one completes an analysis using another sample (as was done in the present study), the weights tend to be different. If the same weights are used to predict from another sample, the weights will probably be even more variable and less accurate in the second sample, because they were made to be biased for the first sample. What ridge regression tends to do is make less bias per sample, causing a smaller R^2.

The larger the R^2, the better the predictive ability of the regression equation. For example, if the R^2 is 1.000, then the observed score and the predicted score are the same. When the R^2 is less than one, there is some error between the predicted score and the observed score. The lower the R^2 the more error variance in the prediction. So, the critical point here is that the prediction equation is only as accurate as the R^2 is large.

The interesting quality of ridge regression is that, while it produces more stable regression weights from sample to sample, it also produces a smaller

R^2, which causes more disparity between the predicted and observed scores. Hence, even though there is more stability in the weights, there will tend to be more error variance, and one's ability to predict in subsequent samples is not necessarily improved.

It is thus important in multiple regression analysis to review the results of cross-validation procedures or shrinkage estimates. If these results tend to be relatively stable, then perhaps ridge analysis would be inappropriate.

Suggested shrinkage estimates may include Wherry's original formula (1931), McNemar's modification (1962) or a formula by Lord (1950). These formulas are indicated below:

$$\hat{R}^2 = 1 - (1 - R^2)\,\frac{N-1}{N-K} \quad \text{(Wherry)}$$

$$\hat{R}^2 = 1 - (1 - R^2)\,\frac{N-1}{N-K-1} \quad \text{(McNemar)}$$

$$\hat{R}^2 = 1 - (1 - R^2)\,\frac{N+K+1}{N-K-1} \quad \text{(Lord)}$$

where:

$\hat{R}$ = the corrected estimate of the multiple correlation
R = the actual calculated multiple correlation
K = the number of independent variables
N = the number of independent observations

As Newman pointed out, the formulas developed by Wherry and McNemar attempt to estimate the population R^2 based on the sample, while Lord's formula estimates the R^2 from one sample to another. The cross-validation procedure is more similar to Lord's procedure. Newman concludes that, in deciding upon the shrinkage method to use, one should consider the underlying assumptions of each procedure (Newman, 1975).

Summary and Conclusions

In the present paper, the investigators described the technique of ridge regression, and listed some of its distinct qualities and purported advantages. Data were then presented which tend to support the conclusion that under certain conditions, ridge regression may be an inappropriate technique.

The ridge technique was examined microscopically and the following specific conclusions were drawn:

1. Ridge regression produces less bias R^2 per sample.
2. Ridge regression produces a smaller R^2.
3. Ridge regression produces more stable weights from sample to sample.
4. Ridge regression is not necessarily more accurate than other methods in predicting a specific score.
5. If cross-validation procedures and/or shrinkage estimates produce little shrinkage of the R^2 from one sample to another, it may not be necessary to use ridge regression, depending on the question being asked.
6. If the proportion of subjects to variables is sufficient and a large R^2 is produced, then the regression model will tend to be stable, and ridge regression may again be inappropriate.

The conclusion of this paper is that ridge regression, while useful in some instances, is not in fact a panacea for use in all regression analyses. In some instances it may be better to use more traditional methods. The authors hope that this brief introduction to ridge regression will be useful to the readers.

References

Hoerl, A. E. Application of ridge analysis to regression problems. Chemical Engineering Progress, 1962, 59 (3), 54-59.

Hoerl, A. E. and Kennard, R. W. Ridge regression: applications to non-orthogonal problems. Technometrics, 1970(a), 12(1), 69-82.

Hoerl, A. E. and Kennard, R. W. Ridge regression: biased estimation for non-orthogonal problems. Technometrics, 1970(b), 12(1), 55-67.

Marquardt, D. W. and Snee, R. D. Ridge regression in practice. The American Statistician, 1975, 29(1), 3-20.

Newman, I., Deitchman, R., Burkholder, J., Sanders, R. E., and Ervin, L. Type VI error: inconsistency between the statistical procedure and the research question. Multiple Linear Regression Viewpoints, 1967, 6, 4.

Newman, I. Variations between shrinkage estimation formulas and the appropriateness of their interpretation. Multiple Linear Regression Viewpoints, 1973, 4(2), 45-48.

Newman, I. and Fraas, J. W. Some applied research concerns using multiple linear regression analysis. Unpublished paper presented at the Ohio Academy of Science, Psychology Division; Capital University, Columbus, Ohio, April, 1977.

Multiple Linear Regression Viewpoints
Volume 8 Number 2

A TEACHING EXAMPLE OF A REPLICABLE SUPPRESSOR VARIABLE*

Dennis W. Leitner
Southern Illinois University at Carbondale

ABSTRACT

One of the more elusive concepts in regression analysis is that of a suppressor variable -- a predictor variable that makes a contribution to the multiple R in excess of its zero-order correlation with the criterion due to its correlation with other predictor(s). The concept is elusive for two reasons: (1) it is hard to understand how any variable could behave in such a way and (2) it is hard to find such a variable in a real world data situation, and upon finding such a variable, hard to replicate. One such example that can be demonstrated using data collected from a "typical" graduate class of approximately 20 persons is the prediction of height using weight and age (the suppressor). It makes some intuitive sense in terms of the definition of a suppressor and it has been found to be replicable in several different classes. Typical results from such classes and intuitive justification for its existence are presented.

Introduction

The beginning student of statistics is aided in his understanding of multiple regression by examples with familiar variables. One of the more common examples of multiple regression is the prediction of success in college using a measure of high school performance (for

Presented at the American Statistical Association, Chicago, IL, August, 1977. The author acknowledges the assistance of John T. Mouw in the development of this paper.

example, grade point average or class rank) and a standardized achievement test result (ACT or SAT). Some variation of the above example is used in the admission policy of most colleges and universities.

To provide an actual numerical example using data on the above variables from the class borders on invasion of privacy, if in fact the information is known. A less threatening example is provided by collecting data on the prediction of height using weight and age.

One of the more obvious principles in the use of multiple regression is that the researcher should use predictors that have moderate to high validity but low (or preferably zero) intercorrelations. Students understand the basic problem of redundance among predictors, and that the multiple coefficient of determination will tend to be <u>less</u> than the sum of the coefficients of determination of the individual predictors. But when the author attempted to illustrate this principle with actual data collected on height, weight, and age (each with some possible "measurement" error), the expected relationship between zero-order and multiple correlations did not hold. This led to a discussion of suppressor variables.

<u>Some literature on suppressor variables</u>

Horst (1941) defined suppressor variables as a variable correlating little or none with the criterion, but nonetheless increasing the prediction when added to the regression with another variable. Conger (1974) provided an expanded definition of suppressor variables that included Horst's.

> A suppressor variable is defined to be a variable which increases the predictive validity of another variable (or set of variables) by its inclusion in a regression equation. This variable is a suppressos only for those variables whose regression weights are increased. (p. 36)

Conger presents three types of suppresion effects (traditional, negative, and reciprocal) and the relationships among the variables.

Cohen and Cohen (1975) discuss examples of the same three types of suppression effects, called classical, net, and cooperative, respectively. But while theoretically the properties necessary for the effect of a suppressor variable has been determined, in practice, the occurrence of suppressor variables is rare, and if present, difficult to replicate. Cohen and Cohen (1975) comment:

> Finally, it is important to note that all three kinds of suppression--classical, net, and cooperative--are not frequently found in behavioral science studies. The detailed presentation here is in the interest of enabling the researcher to recognize them when they do occur, and for their value as quasiparadoxical curiosities. (p. 91)

In an effort to provide data sets illustrating a suppressor variable to students in the classroom, Dayton (1972) showed a method for constructing such a variable: The residual of regression X on Y acts as a suppressor in the regression of Y on X. Small random deviations need to be added to the residuals "for realism" in order to avoid perfect multiple correlation. But while these variables act as suppressors, they lack meaning. It is for this reason that the author was encouraged with the unexpected occurrence of the suppressor variable in the classroom example.

Method and results

In a multiple regression class of 30 graduate students, a sheet of paper was passed around on which each student recorder his/her age (in years), height (in inches), and weight (in pounds). These data were to be analyzed by each student to "see" what happens with multiple regression. As it turned out, something strange happened:

Coefficient of determination between height and weight	= .05
Coefficient of determination between height and age	= .09
Multiple coefficient of determination between height and age, weight	= .23

($r_{age,weight} = .23$)

The whole was more than the sum of the parts and had been expected to be less. This was a situation of cooperative suppression using Cohen's terminology, or reciprocal suppression using Conger's, but what was most intriguing was the possibility of the occurrence of age as a suppressor in other graduate classes. The author and colleagues tried this same experiment with graduate classes ranging in size from 13 to 30, and more often than not, the same suppressor effect was noted. Table 1 presents a summary of the results from the nine classes.

Table 1

Incremental Prediction When the Suppressor Variable (Age)

Is Added to the Regression of Height on Weight

Group Number	Correlation between Age and Height	Increase in R When Age Is Added
1	-.30	.42
2*	-.21	.03
3	-.17	.18
4	-.44	.47
5	-.01	.10
6	-.17	.20
7*	.19	.06
8*	-.30	.26
9	-.02	.10

* In these groups, age did not act as a suppressor variable.

Discussion and conclusion

The above example is intuitively appealing since it follows the traditional conceptions of the requirements of a suppressor variable: weight is a moderate predictor of height, age is negligibly correlated with height, and slightly correlated with weight. It is easy for students to understand how age might increase prediction of height (although unrelated to it) by suppressing some of the irrelevant variation of weight. This irrelevant variation of weight is referred to affectionately as the middle-age spread. Although the magnitude of the suppressor effect is small in most cases, it does tend to replicate in other graduate classes.

This serendipitous example has been effective in classes in explaining suppressor variables in multiple regression using familiar variables.

REFERENCES

Cohen, J. and Cohen, P. Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences. Hilsdale, N.J.: Lawrence Erlbaum Associates, 1975.

Conger, A.J. A revised definition for suppressor variables: A guide to their identification and interpretation. Educational and Psychological Measurement, 1974, 34, 35-46.

Dayton, C.M. A method for constructing data which illustrates a suppressor variable. The American Statistician. 1972, 26, 36.

Horst, P. The prediction of personal adjustment. Social Science Research Council Bulletin, No. 48, New York, N.Y.

Multiple Linear Regression Viewpoints
Volume 8 Number 2

An Interactive Version of MULR04 With Enhanced Graphics Capability

Joel H. Burkholder

National Institute of Mental Health
Unit for Research on Behavioral Systems
Laboratory of Brain Evolution and Behavior

Abstract: A version of MULR04 employing random access Read/Write to simulate core memory for RT11 configured minicomputers is discussed. In addition, an optional capability for obtaining high quality graphics on either Tektronix (4010 family) or Calcomp peripheral devices has been integrated into the system.

While not belaboring the point, multiple linear regression is a very powerful analytical tool. MULR04 in particular, is but one example of such an analysis package which allows large numbers of independent variables to operate on a given dependent variable. In exchange for this, a rather large amount of computer memory must be devoted to the task. Because of this size requirement it is commonly assumed that such complex analysis packages are destined to operate only within the domain of large machines. However, a user may not have either physical or budgetary access to a large machine and thereby be limited in their ability to explore their data in great detail.

With the growing use of minicomputers as a fundamental component of the laboratory setting, the user's problem can be easily rectified with the appropriate software. Unfortunately, most minicomputer software packages are not designed to handle large multiple variable data arrays in their regression analyses. This version of MULR04 couples the flexibility of complex multiple regression with the interactive capability of the minicomputer. This interactive program provides the user with the opportunity to enter data and regression models online. It allows examination of results plus high quality graphics when desired. This defines the scope and purpose for developing an interactive minicomputer version of MULR04.

Program LINGRF (Linear + Graphics)

In terms of actual printout this version is very similar to its original precursor, however, this is where the similarity ends. Rather than submitting a card deck containing the parameter card, format card and data cards, all of these functions are handled through a series of prompts given to the user via the system console device. The user is queried for the name of the file, resident on a peripheral device accessible by the computer, to be read into the computer. Further prompts for the number of observations and variables are made. Based on this input a random access file of a calculated size is allocated. As the specified input file is read into memory, the sums, sum of squares and crossproducts are calculated and written onto the system disk. A linear file simulating a matrix with the sums of squares along the diagonal and the crossproducts situated in the off diagonal is perhaps the easiest way to visualize this file's structure. Following this procedure, control is passed to the correlation subroutine (DFCRLB). This routine, using the necessary random access start points entered through the call statement calculates the appropriate locations of model elements, reads them off the disk, computes the correlation coefficients and writes these values onto the disk in an adjacent block of space appropriated for them. Means and standard deviations are then calculated and written onto the disk. This particular subroutine is by far the slowest of them all, largely owing to the time spent in reading and writing. For every variable added, the number of necessary reads and writes becomes multiplied. However, even at its worst (i.e., ± 10 minutes @ + 50 variables), this is far less than a possible overnight wait when submitting a batch run to a large system.

Control is then passed to the printing subroutine. Similar to the intercorrelation routine the necessary positions are calculated and these values are then read from the disk. For printing the correlation and sum of square-crossproduct matrices, an index array containing the calculated positions of the diagonal elements is used to coordinate the reading and printing of correlations in the proper format.

As control is passed back to the main program, the user is again prompted whether or not a file containing the models is to be entered at that time, or read from an already existing file. If a file already exists, the user responds to the query by entering the appropriate existing file name. Control then passes to the regression routine. If the file does not exist, control is passed via a similar sequence to subroutine IMODEL. IMODEL contains a series of prompts to the user requesting information needed to construct a file of regression models and F tests. A call is then made to the multiple regression subroutine which calculates the squared multiple R using the iterative procedure first proposed by Greenberger and Ward (1956). Examples of this algorithm can be found in Veldman's (1967) text and in Kelly, Beggs and McNeil (1969) from whence this version is based.

Rather than calculating address locations for access in active memory, this version reads these values from the random access device. One additional routine (COLUMN) is repeatedly called upon by the multiple regression routine to calculate the position and select elements from the simulated matrix on disk to assist in calculating the multiple R and later in calculating the standard regression weights.

As the printing of the raw regression weights is completed, a test is made to determine whether or not a save file and graphics are desired. If the user has placed a minus sign before the number of predictor + criterion variables when queried in routine IMODEL, the graphics option becomes enabled. Presence of the minus sign during construction of a particular model assumes the user wants to output a file of the predictor variables, raw scores, predicted scores and residuals or utilize the graphics option or both. The user is prompted whether or not a file is wanted, followed by a prompt asking for a file name to be entered. If no file is wanted, no file name request is made and no file is written. However, an unformatted scratch file is generated in anticipation that the user may desire graphics. If no request for graphics is made, the program will read another model record into memory and repeat the aforementioned sequence of operations.

Graphics

When graphics are requested, a call is made to an administrative routine which queries the user for one of three types of graphs or any combination of the three. When the user decides which graph or series of graphs are wanted, a call is made to the appropriate graphic subroutine. Each of these graphic routines require that the user know what values constitute the particular model being analyzed.

Subroutine MEAN. Upon entering this routine a call is made to a routine (ABEL) which prompts the user for the titles of the ordinate and abscissa. This convention is also followed by the other graphics routines. The length of each character string is calculated with the actual character strings occupying a common array.

Routine MEAN assumes that the elements in the model have been based on dichotomously coded data, as would be the case with an analysis of variance problem. The user is prompted for the number of separate levels and treatment groups within each level. If for example, the user was examining the full model of a 2x2 factorial design experiment, the user would enter a '2' for the number of levels and then another '2' for the number of treatment groups. The user is then prompted to enter the elements of the given model that represent a particular level of analysis. A note of warning: these values are not the actual variable numbers, but the rank position of that variable moving from left to right given by the model. Thus if we have a set of consecutive (or nonconsecutive, it makes no difference) predictor variables of the form N...M their rank position of 1...N would be entered. This provision is made on behalf of the user who enters values relevant to a particular model without regard to their order. Upon entry of these values, their position on the random access device is calculated and the raw standard weights are read and summed with the unit vector which was passed via the call statement. The minimum and maximum values are obtained for the array of means and their rank positions and a scaling factor for the abscissa and ordinate are computed. These scaling constants insure that all the elements in the data array will fit within the graphics window. Following this, a series of repeated calls are made to routines PLOTS, PLOT, SYMBOL and AXIS, which are responsible for blanking the video screen (if using a tektronix storage scope) or advancing the plotting pen along the paper scroll (if using a calcomp plotter), positioning the pen, drawing the data points, (and or lines) the ordinate and abscissa scales, and scribing the axis labels. The top two plots of figure 1 are examples of what is generated from this graphics routine. In both cases, they demonstrate a single factor experiment with four treatment groups in the left-hand side plot and three treatment groups for the right-hand side.

Subroutines SCATT and RESID. Except for some particulars, the logic of these two routines are quite similar. SCATT (SCATT = scatter plot) is a general purpose bivariate plot routine in which the user can make any combination of plots of given continuous predicted variable scores against the raw scores, predicted scores, or each other. At the completion of a plot, a timing loop is called (WAITE) which allows time for the user to study the output and obtain a hard copy (if using a Tektronix terminal with hard copy capability). If a Calcomp plotter is the graphics device being used, this timing loop is not necessary.

In both of these graphics routines, elements from the unformatted scratch file written by the regression routine are read. As was the case in routine MEAN, the minimum and maximum values are obtained, and scaling factors calculated.

The residual plotting routine requires no decision on the part of the user to select the appropriate variables. However, the bivariate plot routine does require that the user be knowledgeable of what the rank order is of the variables entered into the particular model in question. Similar to routine MEAN, the rank position of the two variables being plotted are entered by the user when prompted. In addition, the position of the criterion variable and predicted variable are also accessible by the user. One need only remember that the predictor variables are followed by the criterion and predictor variable, respectively. This is of the form: predictor values; 1...N + criterion value + predicted value. The middle and bottom plots of figure 1 provide examples of these two plot routines.

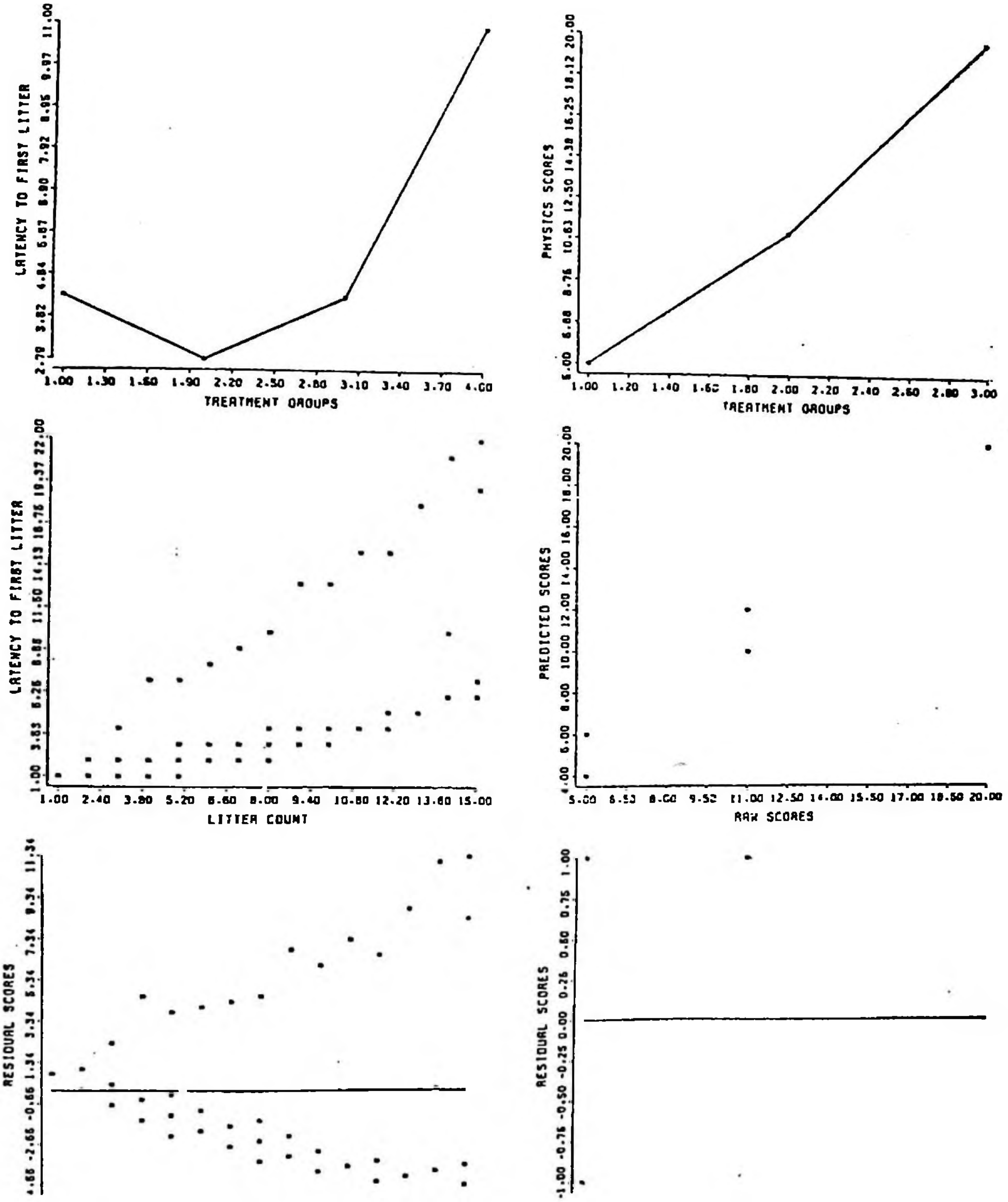
Tektronix and Calcomp graphics routines. The graphics routines called upon by this system (LINGRF), as mentioned earlier, employ calls to routines, SYMBOL, PLOT and AXIS. These three utility routines were originally Calcomp routines which have been modified to interface with RT11 systems. These higher level routines call upon an extensive host of programming which has been integrated into two systems TPLT1 and CPLT1 appended for this system from the Caudar system (Radna & Vaughn, 1977).

LINGRF: Limitations and Applications

One soon notices when comparing this particular version against its parent version that a call to subroutine DFTRAN (or DATRAN) has been omitted. The reason for this is that the monitor system which is resident in most smaller systems is not as sophisticated as what one finds in larger ones. Therefore, data transformations must be generated prior to running the analysis program. A small program (LIND) has been provided as an example of how one may elect to structure their data matrix. If a user has some knowledge of programming (which is necessarily the case when working with a small machine) inserting the transformations, compiling, linking and running this program will not account for much time (+ 5 minutes).

In contrast to this deficit, experience in using this package reveals itself as a formidable laboratory tool, especially when a user is doing work requiring data snooping. The effect of getting an immediate graphic feedback is perhaps the singularly most benificial component of the system. At an intuitive level, the graphics do provide a useful means with which one can understand the nature of his/her data even more.

FIGURE 1


LATENCY TO FIRST LITTER
TREATMENT GROUPS
PHYSICS SCORES
TREATMENT GROUPS
LATENCY TO FIRST LITTER
LITTER COUNT
PREDICTED SCORES
RAW SCORES
RESIDUAL SCORES
RESIDUAL SCORES

Reference Notes

Radna, R. J. and Vaughn, W., CAUDAR (Computer Assisted Unit Acquisition/Reduction), is available through: United States Department of Commerce, National Technical Information Service, Springfield, VA 22161, Accession No. PB 2070745 (in paper copy) and No. PB 2070744 (in 9 track tape). A nominal fee is charged for this service.

Thanks to Dr. James L. Hill for permission to use his data in demonstrating this system's graphics (see figure 1, left-hand side).

This program was developed on the Laboratory of Brain Evolution and Behavior's PDP 11/40 minicomputer system.

References

Greenberger, M.H., and J.H. Ward, Jr., An Iterative Technique for multiple Correlation Analysis, IBM Newsletter, 1956, 85-97.

Kelly, F.J., Beggs, D.L., McNeil, K.A., Research Design in the Behavioral Sciences: Multiple Regression Approach, Carbondale: Southern Illinois University Press, 1969. Data from figure 5-3, page 89 was used for graphic purposes for figure 1 of this paper (right-hand side).

Radna, R.J. and Vaughn, W., CAUDAR, Computer Assisted Unit Acquisition/ Reduction, Electroenceph. Clin. Neurophysiol. In Press, 1977

Tektronix Plot-10, Terminal Control System User's Manual, Document No. 062-1474-00, Tektronix Inc. Beaverton, OR. Copyright 1974.

Veldman, D.J., Fortran Programming for the Behavioral Sciences, New York: Holt Rinehart and Winston, 1967.

```
C       Program Lingrf.For [Linear + Graphics].
C
C
C       An interactive version of Mulro4 adapted for use
C       on RT11 configured minicomputers having a highspeed random
C       access device by J. H. B, Laboratory of Brain Evolution
C       and Behavior, Unit for Research on Behavioral Systems.
C
C       This program in addition to calculating the means,standard
C       deviations,correlations,squared multiple correlation and
C       standardized and raw score regression weights also provides
C       graphics subroutines for output on either calcomp or
C       tektronix graphics terminals.  These graphics subroutines
C       require calls to both calcomp and tektronix system
C       subroutines.
C
C       .............................................................
C
C       Mulro4          Division of Educational Research
C                           University of Alberta
C
        DIMENSION X(100)
        IOUT=6
        COMMON NREC,IOUT
        LOGICAL*1 FMT(80)
        CALL ASSIGN(11,'RK:PLOX.DAT',11,'SCR')
C       Input dialogue
        CALL OUTSTR ('$INPUT DATA FILE NAME')
        CALL ASSIGN (1,0,-1,'RDO')
        CALL ASSIGN (2,'DATA.DAT',8,'SCR')
        CALL ASK ('NUMBER OF OBSERVATIONS',NOB)
        MTXL=0
        DO 200 I=1,NOB
        DO 200 J=I,NOB
200     MTXL=MTXL+1
        CALL ASK (' NUMBER OF VARIABLES',NVARIN)
        ISIZE=(6*NVARIN)+(2*MTXL)+1
        TYPE 400,ISIZE
400     FORMAT(' RANDOM ACCESS FILE LENGTH=',I10)
        DEFINE FILE 4 (ISIZE,2,U,NREC)
        TYPE 1
1       FORMAT(' INPUT FORMAT FOR THIS DATA SET',/)
        ACCEPT 2, FMT
2       FORMAT (80A1)
C       Initialize
        NPER=0
        A=0.0
        DO 9 J=1,ISIZE
        WRITE(4'J) A
9       CONTINUE
        JOBS=0
10      JOBS=JOBS+1
        IF(NOB.EQ.0)GO TO 11
        IF (JOBS.GT.NOB) GO TO 160
11      READ (1,FMT) (X(J),J=1,NVARIN)
14      NPER=NPER+1
        JJ=0
        KK=(2*NVARIN)+1
C       Compute sums of squares and crossproducts and write onto
C       random access device.
        DO 15 J=1,NVARIN
        READ(4'J) A
        KK=KK+JJ
        A=A+X(J)
        WRITE(4'J) A
        JJ=0
        DO 16 K=J,NVARIN
        LLL=KK-1+J+(K-J)
        READ(4'LLL) A2
        A2=A2+(X(J)*X(K))
        WRITE(4'LLL) A2
        JJ=JJ+1
16      CONTINUE
        JJ=JJ-2
15      CONTINUE
        WRITE (2) (X(J),J=1,NVARIN)
        GO TO 10
160     WRITE(6,50)
50      FORMAT(1H1)
```

```
        WRITE(6,51) NOB
51      FORMAT(5X,'NUMBER OF OBSERVATIONS=',I5)
        WRITE(6,52) NPER
52      FORMAT(5X,'NUMBER OF RECORDS READ=',I5)
        WRITE(6,53) NVARIN
53      FORMAT(5X,'NUMBER OF VARIABLES INPUT=',I5)
        WRITE(6,54) FMT
54      FORMAT(,5X,'INPUT FORMAT=',90A1)
        CALL DFCRLB(NPER,NVARIN,MTXL,1,NVARIN+1,2*NVARIN+1)
        NV1=(2*NVARIN)+MTXL+1
        NV2=(2*NVARIN)+(2*MTXL)+1
        CALL CLOSE (1)
        REWIND 2
        TYPE 99
99      FORMAT(' ENTER MODEL SET FROM THE KEYBOARD? = 1',/
        1,' OR READ MODEL SET FROM ALREADY EXISTING FILE? = 0')
        ACCEPT 98,KMOD
98      FORMAT (I1)
        IF(KMOD.EQ.1) GO TO 170
        CALL OUTSTR('$MODEL SET FILE NAME?')
        CALL ASSIGN(3,0,-1,'RDO')
        GO TO 180
170     CALL OUTSTR('$MODEL SET FILE NAME?')
        CALL ASSIGN(3,0,-1,'NEW')
        CALL IMODEL
        REWIND 3
180     CALL GFREGR(NPER,1,NVARIN+1,NV1,NV2,NV2+NVARIN,NV2+2*NVARIN,
        1NVARIN)
        CALL CLOSE (3)
        CALL ASK(' SOMEMORE MODELS? YES=1',IOVER)
        IF (IOVER.EQ.1) GO TO 170
        STOP
        END


C       Imodel.for used to construct regression model file.
        SUBROUTINE IMODEL
        DIMENSION MFLD(56)
        CALL ASK(' ENTER THE NUMBER OF MODELS TO BE BUILT',JMOD)
        ICOUNT=1
        DO 500 I=1,JMOD
        DO 300 J=1,56
300     MFLD(J)=0
        CALL ASK(' ENTER NUMBER OF PREDICTORS+CRITERION VARIABLES',NFLDS)
        TYPE 3
3       FORMAT(' ENTER THE PREDICTOR VARIABLES',/)
        JCOUNT=1
        DO 400 J=1,56
        ACCEPT 4,MFLD(J)
4       FORMAT(I2)
        IF(MFLD(J).EQ.0) GO TO 401
        JCOUNT=JCOUNT+1
400     CONTINUE
401     TYPE 800
800     FORMAT('  CRITERION VARIABLE?')
        ACCEPT 4,ICRIT
        MFLD(JCOUNT)=ICRIT
        WRITE(3,600) ICOUNT,NFLDS,(MFLD(J),J=1,JCOUNT)
600     FORMAT(I2,I3,56I2)
        ICOUNT=ICOUNT+1
500     CONTINUE
        CALL ASK(' ENTER THE NUMBER OF F-TEST COMPARISONS',IFR)
        NFLDS=6
        ICOUNT=1
        DO 700 I=1,IFR
        DO 502 J=1,56
502     MFLD(J)=0
        TYPE 6,ICOUNT
6       FORMAT(' FULL MODEL FOR COMPARISON #',I2,'=',$)
        ACCEPT 2,IFULL
2       FORMAT(I2)
        TYPE 7,ICOUNT
7       FORMAT(' RESTRICTED MODEL FOR COMPARISON#',I2,'=',$)
        ACCEPT 2,IREST
        CALL ASK(' NUMERATOR DEGREES OF FREEDOM=',KNUM)
        ANUM=FLOAT(KNUM)
```

```
      WRITE(4'IM) AIM
      IS=LSIGMA-1+J
      II=LCORR
      READ(4'II) AII
      AIS=SQRT((AII-((AIM2*AIM2)/FN))/FN)
      WRITE(4'IS)AIS
      LCORR=LCORR+NVAR-(I-1)
23    CONTINUE
      CALL DFPRNT (1,NVAR,1,KMEAN,8,8HMEANS     )
      CALL DFPRNT (1,NVAR,1,KSIGMA,20,20HSTANDARD DEVIATIONS )
      CALL DFPRNT (NVAR,NVAR,1,MBCOR,12,12HCORRELATIONS)
      CALL DFPRNT (NVAR,NVAR,1,KKKK,29,29HSUM-OF-SQUARES-CROSSPRODUCTS )
      RETURN
      END


      SUBROUTINE DFPRNT (NR,NC,NRMAX,LNUM,NUMHOL,TITLE)
C     Prints arrays of means and standard deviations and prints the
C     off diagonal portion of the r matrix accessed from the disk
      DIMENSION TITLE(20),X(10),J1(10),LDIAG(100)
      COMMON NREC,IOUT
      KNUM=LNUM
      N=(NUMHOL+3)/4
      WRITE(IOUT,1) (TITLE(J),J=1,N)
1     FORMAT(1H0/1H0,20A4)
      N=((NC-1)/10)+1
      IF(NR.NE.1) GO TO 66
      JA=0
      JB=0
      DO 9 I=1,N
      IF(N.NE.1) GO TO 10
      JA=1
      JB=NC
      IR=NC
      GO TO 19
10    IF(N.EQ.I) GO TO 11
      IR=10
      JA=JB+1
      JB=JA+9
      GO TO 19
11    IR=(NC/10)
      IR=NC-(IR*10)
      JA=JB+1
      JB=JA+IR-1
19    DO 8 K=1,IR
      READ(4'KNUM) A
      X(K)= A
8     KNUM=KNUM+J
      WRITE(IOUT,120) (J,J=JA,JB)
      WRITE(IOUT,3)
3     FORMAT(1H )
9     WRITE(IOUT,110) I,(X(J),J=1,IR)
      RETURN
66    INUM=1-(NC+1)
      DO 100 J=1,NC
      INUM=INUM+(NC-(J-1))+1
100   LDIAG(J)=INUM
      IFLG=1
      IC=NC/10
      IC=(NC-(IC*10))
      IF(N.EQ.1)IC=NC
      DO 1000 NS=1,N
      IF((NS*10).EQ.NC) GO TO 450
      IF(NS.EQ.N) GO TO 410
      IF(N.EQ.1) GO TO 410
450   N10=10
      GO TO 430
410   N10=IC
430   CONTINUE
      IP10=10*NS
      IP=IP10-9
      J=1
      DO 22 I=IP,IP10
      J1(J)=LDIAG(I)
22    J=J+1
      KKK=IP-1
```

```
      CNUM=AINT(ANUM/100.)
      INUM=IFIX(CNUM)
      JNUM=IFIX(ANUM-(CNUM*100.))
      CALL ASK(' DENOMINATOR DEGREES OF FREEDOM=',KDEN)
      ADEN=FLOAT(KDEN)
      CDEN=AINT(ANUM/100.)
      IDEN=IFIX(CDEN)
      JDEN=IFIX(ADEN-(CDEN*100.))
      MFLD(1)=IFULL
      MFLD(2)=IREST
      MFLD(3)=INUM
      MFLD(4)=JNUM
      MFLD(5)=IDEN
      MFLD(6)=JDEN
      WRITE(3,600) ICOUNT,NFLDS,(MFLD(J),J=1,6)
      ICOUNT=ICOUNT+1
700   CONTINUE
      RETURN
      END



      SUBROUTINE DFCRLB (NUM,NVAR,MTXL,LMEAN,LSIGMA,LCORR)
C     Calculate means,standard deviations,and correlations.
C     This version employs random access read/write to simulate
C     core memory.
C     Portion of corrlb 8 Feb/65 modified May/66 from U of A
      COMMON NREC,IOUT
      KKKK=LCORR
      FN=NUM
      JJ=MTXL
      KMEAN=LMEAN
      KSIGMA=LSIGMA
      MCOR=(LCORR+JJ)-1
      MBCOR=MCOR+1
      JCORR=LCORR
C     Compute diagonal and off diagonal elements of r matrix
      LLLL=0
      DO 16 I=1,NVAR
      JJ=0
      JCORR=LCORR
      DO 17 J=I,NVAR
      LLLL=LLLL+1
      ISI=LCORR
      ISJ=JCORR
      ISJI=LCORR+JJ
      ISIJ=MCOR+LLLL
      IMI=LMEAN-1+I
      IMJ=LMEAN-1+J
      READ(4'ISJI) AISJI
      READ(4'ISI) AISI
      READ(4'IMI) AIMI
      READ(4'ISJ) AISJ
      READ(4'IMJ) AIMJ
      IF(ISI-ISJ) 190,18,190
18    XR=1.0
      GO TO 19
190   XD=SQRT(((FN*AISI)-(AIMI*AIMI))*((FN*AISJ)-(AIMJ*AIMJ)))
      IF(XD) 32,33,32
33    XR=0.0
      GO TO 19
32    XR=((FN*AISJI)-(AIMI*AIMJ))/XD
19    WRITE(4'ISIJ) XR
      JCORR=JCORR+NVAR-(J-1)
      JJ=JJ+1
17    CONTINUE
      LCORR=LCORR+NVAR-(I-1)
16    CONTINUE
C     Compute means and sigmas
      LCORR=KKKK
      LMEAN=KMEAN
      LSIGMA=KSIGMA
      DO 23 I=1,NVAR
      IM=LMEAN-1+I
      READ(4'IM) AIM
      AIM2=AIM
      AIM=AIM/FN
```

```
      IF(NS.EQ.N))IP10=IP+(N10-1)
69    WRITE(IOUT,120) (L,L=IP,IP10)
C     Read/write the diagonal set
      DO 330 I=1,N10
      DO 320 J=1,I
      IVAR=J1(J)+KNUM-1
      J1(J)=J1(J)+1
      READ(4'IVAR) A
320   X(J)=A
      KKK=KKK+1
330   WRITE(IOUT,110) KKK,(X(L),L=1,I)
      IF(NS.EQ.N) RETURN
500   DO 530 I=1,10
      IVAR=J1(I)+KNUM-1
      J1(I)=J1(I)+1
      READ(4'IVAR) A
530   X(I)=A
      KKK=KKK+1
540   WRITE(IOUT,110) KKK,(X(L),L=1,10)
      IF(KKK.NE.NC) GO TO 500
1000  CONTINUE
120   FORMAT(1X,//,8X,10I12)
110   FORMAT(1X,I4,5X,10F12.4)
      RETURN
      END



      SUBROUTINE GFREGR (NPER,LMEAN,LSIGMA,LCORR,LSTDWT,LWTS,LRSQ
     1,NVAR)
C     Regred from 8 Feb modified version by Flathman,U of A.
C     Iterative regression
C     Modifed for use on RT11 configured minicomputers.
C     This version provides the user the option of
C     writing predictor variables ,criterion ,predicted
C      and residual scores onto a user specified file.
C     In addition, user has the option of either calcomp or
C     tecktronix graphics display of;  1-the means of groups,
C     2-scatter plot of variables specified in model set, 3-plot
C     of the residuals
      DIMENSION X(100),Z(100)
      INTEGER*2 MFLD(56),MFLDL(27)
      LOGICAL*1 BMT(80)
      COMMON NREC,IOUT
      STOPC=.00001
      MSI=0
      K6=0
2     READ (3,3,END=18)IPROB,NFLDS,(MFLD(I),I=1,56)
3     FORMAT (I2,I3,56I2)
12    IRESID=0
      IF(NFLDS)13,18,19
13    IRESID=1
      NFLDS=IABS(NFLDS)
      GO TO 19
18    RETURN
19    IF(NFLDS-NFLDS/2*2)20,52,20
20    K5=NFLDS-1
      IDC=MFLD(NFLDS)
      WRITE(6,500)
500   FORMAT(1H0)
      WRITE(6,21)IPROB,STOPC,IDC,(MFLD(I),I=1,K5)
21    FORMAT (/,10X,'....','MODEL',I2,E20.8,/,10X,'CRITERION',I5,/,10X
     1,'PREDICTORS',5X,56(I2,1X))
      CALL COLUMN(NVAR,LCORR,IDC,Z)
      NFLD1=NFLDS-1
      DO 22 I=2,NFLD1,2
      M=I/2
      MFLDL(M)=MFLD(I)
22    MFLD(M)=MFLD(I-1)
      DO 23 I=1,NVAR
      J=I+LWTS-1
      A=0.0
      WRITE (4'J) A
      J=I+LSTDWT-1
23    WRITE(4'J) A
      S=0.0
      SIG2=0.0
```

```
      RSQ=0.0
      DEL=0.0
      ITER=0
      ID=1
      CALL COLUMN(NVAR,LCORR,ID,X)
      NGRP=NFLDS/2
24    RSQL=0.0
      DO 29 I=1,NGRP
      KSTAR=MFLD(I)
      KSTOP=MFLDL(I)
      DO 29 J=KSTAR,KSTOP
      IA=(LWTS-1)+J
      READ(4'IA) AA
      AA=AA+(DEL*X(J))
      WRITE(4'IA) AA
      DEN=S-(AA*Z(J))
      IF (DEN) 26,25,26
25    DELT= Z(J)
      STEST=DELT*DELT
      SIG2T=STEST
      RSQT=STEST
      GO TO 27
26    DELT=((SIG2*Z(J))-(S*AA))/DEN
      STEST=S+(DELT*Z(J))
      SIG2T=SIG2+(2.0*AA*DELT)+(DELT*DELT)
      RSQT=(STEST*STEST)/SIG2T
27    IF (RSQL-RSQT)28,29,29
28    SLAR=STEST
      SIG2L=SIG2T
      RSQL=RSQT
      DELTL=DELT
      IDLAR=J
29    CONTINUE
      IF(RSQL-RSQ.LE.STOPC)GO TO 33
      S=SLAR
      SIG2=SIG2L
      RSQ=RSQL
      DEL=DELTL
      ITER=ITER+1
      IZ=(LSTDWT-1)+IDLAR
      ID=IDLAR
      CALL COLUMN(NVAR,LCORR,ID,X)
      READ(4'IZ) AY
      AZ=AY+DEL
      WRITE(4'IZ)AZ
      IF(RSQ-1.)24,33,31
31    WRITE (6,32)
32    FORMAT (///,' RSQ IS GREATER THAN 1.0,CHECK THIS MODEL AND'
     1,/,' AVOID LATER INTERPRETATIONS INVOLVING THIS MODEL')
      RSQL=9999.9
      GO TO 45
33    SDS2=S/SIG2
      WRITE(6,34)RSQL,ITER
34    FORMAT (/,14X,5HRSQ =F11.8,30X,I5,1X,'ITERATIONS')
      DO 35 I=1,NGRP
      KSTAR=MFLD(I)
      KSTOP=MFLDL(I)
      DO 35 J=KSTAR,KSTOP
      IA=LSTDWT-1+J
      READ(4'IA) AA
      AA=AA*SDS2
35    WRITE(4'IA) AA
      WRITE (6,36)
36    FORMAT (/10X' VAR. NUMBER      STD. WT.        ERROR'/)
      DO 38 I=1,NGRP
      KSTAR=MFLD(I)
      KSTOP=MFLDL(I)
      DO 38 J=KSTAR,KSTOP
      IA=LWTS-1+J
      A=0.0
      WRITE(4'IA) A
      DO 37 IL=1,NGRP
      LSTAR=MFLD(IL)
      LSTOP=MFLDL(IL)
      DO 37 L=LSTAR,LSTOP
      CALL COLUMN(NVAR,LCORR,L,X)
      IB=LSTDWT-1+L
      READ(4'IA) AA
      READ(4'IB) AB
      A=AA+(AB*X(J))
```

```
37      WRITE(4'IA) A
        READ(4'IA) AA
        AX=AA-Z(J)
        WRITE(4'IA) AX
        IB=LSTDWT-1+J
        READ(4'IB) AB
38      WRITE(6,39) J,AB,AX
39      FORMAT(1X,I13,F18.8,F15.8)
        WRITE (6,40)
40      FORMAT(/,9X,'  VAR.  NUMBER          WEIGHT'/)
        FK1=0.0
        DO 43 I=1,NGRP
        KSTAR=MFLD(I)
        KSTOP=MFLDL(I)
        DO 43 J=KSTAR,KSTOP
        IA=LSIGMA-1+J
        IB=LSTDWT-1+J
        IC=LSIGMA-1+IDC
        IE=LWTS-1+J
        READ(4'IA) AA
        IF (AA)42,41,42
41      A=0.0
        WRITE(4'IE) A
        GO TO 43
42      READ(4'IB) AB
        READ(4'IC)AC
        A=AB*(AC/AA)
        AX=A
        WRITE(4'IE) A
        READ(4'ID)AD
        FK1=FK1+(AB*AD)/AA
43      WRITE (6,39) J,AX
        ID=LMEAN-1+IDC
        READ(4'ID) AD
        READ(4'IC) AC
        REGCO=AD-(AC*FK1)
        WRITE (6,44) REGCO
44      FORMAT (10X,'CONSTANT='F18.8,//)
45      K5=(LRSQ-1)+IPROB
        WRITE(4'K5) RSQL
        JOUT=0
        IF(IRESID.NE.1) GO TO 50
        TYPE 550,IPROB
550     FORMAT(' MODEL',I2,'-RESIDUAL/PREDICTOR OUTPUT?,1=YES,0=NO')
        ACCEPT 551,JOUT
551     FORMAT(I1)
        IF(JOUT.NE.1) GO TO 552
        CALL OUTSTR('$RESIDUAL/PREDICTORS FILENAME?')
        CALL ASSIGN(10,0,-1,'NEW')
        CALL OUTSTR(' OUTPUT FORMAT FOR SAVE FILE: ')
        ACCEPT 666,BMT
666     FORMAT(80A1)
552     DO 48 IIOB=1,NPER
        READ(2) (X(I),I=1,NVAR)
        PREY=0
        KNUM=1
        DO 47 I=1,NGRP
        KSTAR=MFLD(I)
        KSTOP=MFLDL(I)
        DO 47 J=KSTAR,KSTOP
        Z(KNUM)=X(J)
        KNUM=KNUM+1
        IE=LWTS-1+J
        READ(4'IE) AE
47      PREY=PREY+AE*X(J)
        PREY=PREY+REGCO
        DIF=X(IDC)-PREY
        IF(JOUT.NE.1)GO TO 48
        WRITE(10,BMT)IIOB,(Z(K),K=1,KNUM-1),X(IDC),PREY,DIF
48      WRITE (11) IIOB,(Z(K),K=1,KNUM-1),X(IDC),PREY,DIF
        CALL CLOSE (10)
        CALL ASK('  ANY GRAPHICS ?,YES=1,NO=0',IGRAF)
        IF(IGRAF.NE.1) GO TO 553
        CALL DFGRAF(NPER,Z,KNUM-1,LWTS,REGCO)
553     REWIND 2
50      WRITE(6,51)
51      FORMAT (/,1X,80('*'))
        GO TO 2
52      DF1=MFLD(3)*100+MFLD(4)
        DF2=MFLD(5)*100+MFLD(6)
```

```
      K8=MFLD(1)-1+LRSQ
      READ(4'K8) AK
      IF(MFLD(2).EQ.99) GO TO 777
      K9=MFLD(2)-1+LRSQ
      READ(4'K9) AKK
      GO TO 778
777   AKK=0.0
778   IF (NFLDS.NE.3) GO TO 53
      IF (AK.GT.1.0.OR.AKK.GT.1.0) GO TO 55
      FNPER=NPER
      DF1=DF1-DF2
      DF2=FNPER-DF1-DF2
53    F=((AK-AKK)/DF1) / ((1.0-AK)/DF2)
C     PRBF from Veldman page 131.
      P=PRBF(DF1,DF2,F)
      I=DF1
      J=DF2
      WRITE (6,54) IPROB,F,I,J,MFLD(1),AK,MFLD(2),AKK,P
54    FORMAT (1X,'F-RATIO ',I2,4H F =,F10.4,8X,'D.F.  NUM.='I4,2X,'D.F.'
     1,'DEN.=',I5,8X,I4,F8.5,I4,F8.5,8X,'PROB =',F8.5)
      GO TO 2
55    WRITE (6,56)
56    FORMAT (10X,'THE RECORD IN THIS LOCATION CANNOT BE'
     1,'INTERPRETED! ONE OF THE MODELS INVOLVED WAS IN ERROR.'//)
      GO TO 2
      END


      SUBROUTINE COLUMN (NVAR,LCORR,IDC,Z)
      DIMENSION Z(100) ,LDIAG(100)
      COMMON NREC,IOUT
      DO 100 J=1,100
100   Z(J)=0.0
      INUM=1-(NVAR+1)
      DO 200 J=1,NVAR
      INUM=INUM+(NVAR-(J-1))+1
200   LDIAG(J)=INUM
C     Select column elements above IDC
      NUP=IDC-1
      IF(NUP.EQ.0) GO TO 350
      DO 300 J=1,NUP
      IVAR=LDIAG(IDC-J)+J
      IVAR=IVAR+LCORR-1
      READ(4'IVAR) A
300   Z(IDC-J)=A
C     Select column elements below IDC
350   IVAR=LDIAG(IDC)-1
      DO 400 J=IDC,NVAR
      IVAR=IVAR+1
      JVAR=IVAR+LCORR-1
      READ(4'JVAR) A
400   Z(J)=A
      RETURN
      END


      SUBROUTINE DFGRAF(NPER,Z,KNUM,LWTS,REGCO)
      LOGICAL*1 XLAB(82),YLAB(82)
      COMMON/ESWCH/III
      COMMON NREC,IOUT
      COMMON/LABEL/XLAB,YLAB
      COMMON/STUFF/ITME,XLEN,YLEN,XDIST
      ITME=20
      XLEN=10.0
      YLEN=8.0
      XDIST=1.0
      REWIND 11
      CALL OUTSTR(' PLOT OF TREATMENT GROUP MEANS=1')
      CALL OUTSTR(' SCATTER PLOT=2')
      CALL OUTSTR(' RESIDUAL PLOT=3')
      CALL OUTSTR(' MEANS + SCATTERGRAM=4')
      CALL OUTSTR(' MEANS + SCATTERGRAM + RESIDUALS=5')
```

```
      CALL OUTSTR(' SCATTERGRAM + RESIDUALS=6')
      ACCEPT 3,KGRAF
3     FORMAT(I5)
      IF(KGRAF.EQ.6) GO TO 200
      IF(KGRAF.GE.4) GO TO 100
      GO TO (100,200,300) KGRAF
100   CALL MEAN(LWTS,REGCO)
      IF(KGRAF-4) 400,200,200
200   CALL SCATT(NPER,Z,KNUM)
      IF(KGRAF-5) 400,300,300
300   CALL RESID(NPER,Z,KNUM)
400   RETURN
      END


      SUBROUTINE MEAN(LWTS,REGCO)
      DIMENSION IDFLD(10,10),YMEAN(10,10)
      LOGICAL*1 XLAB(82),YLAB(82)
      COMMON /ESWCH/III
      COMMON NREC,IOUT
      COMMON/LABEL/XLAB,YLAB
      COMMON/STUFF/ITME,XLEN,YLEN,XDIST
      NXCHAR=0
      NYCHAR=0
      CALL ABEL(NXCHAR,NYCHAR,0)
      CALL ASK(' ENTER NUMBER OF SEPERATE LEVELS',NGRPS)
      CALL ASK(' ENTER NUMBER OF TREATMENT GROUPS',IGRPS)
      DO 500 I=1,NGRPS
      TYPE 11,I
11    FORMAT(' BEGIN PROMPT SEQUENCE OF MODEL GROUPS FOR LEVEL',I5)
      DO 500 J=1,IGRPS
      TYPE 22,J
22    FORMAT('$ENTER MODEL VARIABLE FOR TREATMENT GROUP',I5,
      1'=')
      ACCEPT 3,IFIND
3     FORMAT(I5)
      IDFLD(I,J)=J
      JFIND=(LWTS-1)+IFIND
      READ(4'JFIND) XWT
      YMEAN(I,J)=XWT+REGCO
500   CONTINUE
      YMIN=YMEAN(1,1)
      YMAX=YMIN
      IMIN=IDFLD(1,1)
      IMAX=IMIN
      DO 501 I=1,NGRPS
      DO 501 J=1,IGRPS
      IF(YMEAN(I,J).LT.YMIN) YMIN=YMEAN(I,J)
      IF(YMEAN(I,J).GT.YMAX)YMAX=YMEAN(I,J)
      IF(IDFLD(I,J).LT.IMIN)IMIN=IDFLD(I,J)
      IF(IDFLD(I,J).GT.IMAX)IMAX=IDFLD(I,J)
501   CONTINUE
      XMIN=FLOAT(IMIN)
      XMAX=FLOAT(IMAX)
      YSIZE=(YMAX-YMIN)/YLEN
      XSIZE=(XMAX-XMIN)/XLEN
      III=1
      CALL PLOTS(0,0)
      III=0
      CALL PLOT(2.0,1.5,-3)
      DO 90 I=1,NGRPS
      AY=(YMEAN(I,1)-YMIN)/YSIZE
      AX=(FLOAT(IDFLD(I,1))-XMIN)/XSIZE
      CALL PLOT(AX,AY,3)
      DO 90 J=1,IGRPS
      AY=(YMEAN(I,J)-YMIN)/YSIZE
      AX=(FLOAT(IDFLD(I,J))-XMIN)/XSIZE
      CALL PLOT(AX,AY,2)
      CALL SYMBOL(AX,AY,0.07,I,0.0,-1)
90    CONTINUE
C     DRAW LABELS AND AXES
      DELTA=XSIZE*XDIST
      CALL AXIS(0.0,0.0,XLAB,NXCHAR,XLEN,0.0,XMIN,DELTA,XDIST)
      DELTA=YSIZE*XDIST
      CALL AXIS(0.0,0.0,YLAB,NYCHAR,YLEN,90.0,YMIN,DELTA,XDIST)
      CALL WAITE(ITME)
```

```
      III=1
      CALL PLOTS(0,0)
      RETURN
      END


      SUBROUTINE SCATT(NPER,Z,KNUM)
      LOGICAL*1 XLAB(82),YLAB(82)
C     Scatter plot of variables entered by the user
      DIMENSION Z(100)
      COMMON/ESWCH/III
      COMMON NREC,IOUT
      COMMON /LABEL/XLAB,YLAB
      COMMON/STUFF/ITME,XLEN,YLEN,XDIST
      NXCHAR=0
      NYCHAR=0
      CALL OUTSTR(' THIS IS FOR PLOTTING CONTINUOUS VARIALBES')
1000  CALL ABEL(NXCHAR,NYCHAR,0)
      CALL ASK(' ABSCISSA CONTINUOUS VARIABLE=',IX)
      CALL ASK(' ORDINATE CONTINUOUS VARIABLE=',IY)
      READ (11) IIOB,(Z(K),K=1,KNUM+2),DIF
      YSTART=Z(IY)
      XSTART=Z(IX)
      YMIN=YSTART
      XMIN=XSTART
      YMAX=YMIN
      XMAX=XMIN
      DO 500 I=1,NPER-1
      READ (11) IIOB,(Z(K),K=1,KNUM+2),DIF
      IF(Z(IY).LT.YMIN)YMIN=Z(IY)
      IF(Z(IY).GT.YMAX)YMAX=Z(IY)
      IF(Z(IX).LT.XMIN)XMIN=Z(IX)
      IF(Z(IX).GT.XMAX)XMAX=Z(IX)
500   CONTINUE
      REWIND 11
      YSIZE=(YMAX-YMIN)/YLEN
      XSIZE=(XMAX-XMIN)/XLEN
      III=1
      CALL PLOTS(0,0)
      III=0
      CALL PLOT(2.0,1.5,-3)
      AY=(YSTART-YMIN)
      IF(YSTART.EQ.0.0) GO TO 50
      AY=AY/YSIZE
      GO TO 51
50    AY=0.0
51    AX=(XSTART-XMIN)
      IF(AX.EQ.0.0) GO TO 52
      AX=AX/XSIZE
      GO TO 53
52    AX=0.0
53    CALL PLOT(AX,AY,3)
      DO 600 I=1,NPER
      READ (11) IIOB,(Z(K),K=1,KNUM+2),DIF
      AY=(Z(IY)-YMIN)
      IF(AY.EQ.0.0) GO TO 60
      AY=AY/YSIZE
      GO TO 61
60    AY=0.0
61    AX=(Z(IX)-XMIN)
      IF(AX.EQ.0.0) GO TO 62
      AX=AX/XSIZE
      GO TO 63
62    AX=0.0
63    CALL SYMBOL (AX,AY,0.07,1,0.0,-1)
600   CONTINUE
      REWIND 11
      DELTA=XSIZE*XDIST
      CALL AXIS(0.0,0.0,XLAB,NXCHAR,XLEN,0.0,XMIN,DELTA,XDIST)
      DELTA=YSIZE*XDIST
      CALL AXIS(0.0,0.0,YLAB,NYCHAR,YLEN,90.0,YMIN,DELTA,XDIST)
      CALL WAITE(ITME)
      III=1
      CALL PLOTS(0,0)
      CALL ASK(' ANYMORE SCATTER PLOTS,1=YES',IPLOT)
      IF(IPLOT.EQ.1) GO TO 1000
```

```
      RETURN
      END

      SUBROUTINE RESID(NPER,Z,KNUM)
C     Plot residuals
      LOGICAL*1 XLAB(82),YLAB(82)
      DIMENSION Z(100)
      COMMON/ESWCH/III
      COMMON NREC,IOUT
      COMMON /LABEL/XLAB,YLAB
      COMMON/STUFF/ITME,XLEN,YLEN,XDIST
      NXCHAR=0
      NYCHAR=0
1000  CALL ABEL(NXCHAR,NYCHAR,1)
      READ (11) IIOB,(Z(K),K=1,KNUM),RAW,PREY,DIF
      YSTART=DIF
      XSTART=PREY
      YMIN=YSTART
      XMIN=XSTART
      YMAX=YMIN
      XMAX=XMIN
      DO 500 I=1,NPER-1
      READ (11) IIOB,(Z(K),K=1,KNUM),RAW,PREY,DIF
      IF(DIF.LT.YMIN)YMIN=DIF
      IF(DIF.GT.YMAX)YMAX=DIF
      IF(PREY.LT.XMIN)XMIN=PREY
      IF(PREY.GT.XMAX)XMAX=PREY
500   CONTINUE
      REWIND 11
      YSIZE=(YMAX-YMIN)/YLEN
      XSIZE=(XMAX-XMIN)/XLEN
      III=1
      CALL PLOTS(0,0)
      III=0
      CALL PLOT(2.0,1.5,-3)
      AY=(YSTART-YMIN)
      IF(AY.EQ.0.0) GO TO 50
      AY=AY/YSIZE
      GO TO 51
50    AY=0.0
51    AX=(XSTART-XMIN)
      IF(AX.EQ.0.0)GO TO 52
      AX=AX/XSIZE
      GO TO 53
52    AX=0.0
53    CALL PLOT(AX,AY,3)
      DO 600 I=1,NPER
      READ (11) IIOB,(Z(K),K=1,KNUM),RAW,PREY,DIF
      AY=(DIF-YMIN)
      IF(AY.EQ.0.0) GO TO 60
      AY=AY/YSIZE
      GO TO 61
60    AY=0.0
61    AX=(PREY-XMIN)
      IF(AX.EQ.0.0) GO TO 62
      AX=AX/XSIZE
      GO TO 63
62    AX=0.0
63    CALL SYMBOL (AX,AY,0.07,1,0.0,-1)
600   CONTINUE
      REWIND 11
      IF(YMAX-ABS(YMIN))200,201,200
201   AY=YLEN/2
      GO TO 202
200   RANGE=(YMAX-YMIN)
      AA=YLEN/RANGE
      AA=YMAX*AA
      AY=YLEN-AA
202   AX=XLEN
      CALL PLOT(AX,AY,3)
      CALL PLOT(AX,AY,2)
      AX=0.0
      CALL PLOT(AX,AY,2)
      DELTA=YSIZE*XDIST
      CALL AXIS(0.0,0.0,YLAB,NYCHAR,YLEN,90.0,YMIN,DELTA,XDIST)
```

```
      CALL WAITE(ITME)
      III=1
      CALL PLOTS(0,0)
      RETURN
      END


      SUBROUTINE ABEL(NXCHAR,NYCHAR,JUMP)
      LOGICAL*1 XLAB(82),YLAB(82),TLAB(2)
      COMMON/LABEL/XLAB,YLAB
      CALL OUTSTR(' ORDINATE:')
      CALL OUTSTR(' TYPE LABEL WITHIN DELIMITERS')
      ACCEPT 1,YLAB
1     FORMAT(82A1)
      IF(JUMP.EQ.1) GO TO 80
      CALL OUTSTR(' ABSCISSA:')
      CALL OUTSTR(' TYPE LABEL WITHIN DELIMITERS')
      ACCEPT 1,XLAB
      TLAB(1)=XLAB(1)
      DO 40 I=2,82
      IF(XLAB(I).EQ.TLAB(1)) GO TO 50
      XLAB(I-1)=XLAB(I)
40    CONTINUE
50    NXCHAR=I-2
      NXCHAR=NXCHAR*-1
80    TLAB(1)=YLAB(1)
      DO 60 I=2,82
      IF(YLAB(I).EQ.TLAB(1)) GO TO 70
      YLAB(I-1)=YLAB(I)
60    CONTINUE
70    NYCHAR=I-2
      RETURN
      END


      SUBROUTINE WAITE (ITME)
      ISEC=25
      DO 1000 I=1,ITME
      DO 1000 J=1,ISEC
      DO 1000 K=1,ISEC
      DO 1000 L=1,ISEC
1000  CONTINUE
      RETURN
      END


C     SUBROUTINE OUTSTR--OUTPUTS A CHARACTER STRING TO CONSOLE
C               TERMINAL
C     ROUTINE WRITTEN BY WAYNE RASBAND,TECHNICAL DEVELOPMENT,
C     INTERMURAL RESEARCH, NATIONAL INSTITUTE OF MENTAL HEALTH
C
C
C     FORM:     CALL OUTSTR(S)
C
C               WHERE:    S=QUOTED ALPHANUMERIC LITERAL OR THE NAME
C                         OF AN ARRAY CONTAINING A CHARACTER
C                         STRING TERMINATED BY A NULL BYTE
C
C     NOTE:     THE FIRST CHARACTER OF THE STRING IS USED AS A
C               CARRIAGE CONTROL CHARACTER AS FOLLOWS:
C
C                         SPACE     ADVANCE ONE LINE
C                         0         ADVANCE TWO LINES
C                         1         ADVANCE TO TOP OF NEXT PAGE
C                         +         OVERPRINT
C                         $         SUPPRESS CARRIAGE RETURN AT END
C                                   OF LINE
C
```

```
C
        SUBROUTINE OUTSTR(S)
        LOGICAL*1 S(133)
C
C                 SEARCH STRING FOR NULL BYTE TERMINATOR
C
        DO 100 I=1,133
100     IF(S(I).EQ.0)GOTO 200
        RETURN
C
C                 OUTPUT STRING
C
200     TYPE 1,(S(J),J=1,I-1)
1       FORMAT(132A1)
        RETURN
        END


C       ASK SUBROUTINE--- PROMPTS OPERATOR FOR INTEGER VALUE.
C
C       ROUTINE WRITTEN BY BILL VAUGHN, TECHNICAL DEVELOPMENT,
C       INTERMURAL RESEARCH, NATIONAL INSTITUTE OF MENTAL HEALTH.
C
C       FORM: CALL ASK('STRING',NUM)
C
C
C                 WHERE:   'STRING' = PROMPTING MESSAGE
C                               NUM = INTEGER VARIABLE THAT WILL CONTAIN TYPED RESPO
C
C
C
                  SUBROUTINE ASK(STR,NUM)
                  LOGICAL*1 STR(133)
C
C       SEARCH STRING FOR THE NULL BYTE TERMINATOR
                  DO 100 I=1,133
100               IF(STR(I).EQ.0)GOTO 200
C
C       RETURN IF TERMINATOR NOT FOUND
                  RETURN
C
C       OUTPUT PROMPT
C
200     TYPE 2,(STR(J),J=1,I-1)
2       FORMAT('$',132A1)
C
C       ACCEPT RESPONSE
                  ACCEPT 1,NUM
1                 FORMAT(I7)
                  RETURN
                  END


C       Program Lind.for, replaces subroutine dftran or datran.
C       This program to be used in conjunction with program lingrf.for
        DIMENSION X(250)
        LOGICAL*1 AMT(80),BMT(80)
        CALL OUTSTR('$INPUT FILE NAME?')
        CALL ASSIGN(2,0,-1,'RDO')
        CALL OUTSTR('$OUTPUT FILE NAME?')
        CALL ASSIGN(3,0,-1,'NEW')
        TYPE 1
1       FORMAT('  INPUT FORMAT?')
        ACCEPT 2,AMT
2       FORMAT(80A1)
        TYPE 3
3       FORMAT('  OUTPUT FORMAT?')
        ACCEPT 2,BMT
        CALL ASK(' INPUT NUMBER OF VARIABLES?',IIN)
        CALL ASK('  NUMBER OF NEW VARIABLES?',IOUT)
        IOUT=IOUT+IIN
10      DO 500 J=1,250
```

```
      X(J)=0.0
500   READ(2,AMT,END=100) (X(J),J=1,IIN)
C
C
C
C
C
C     enter data transformation set here
C
C
C
C
      WRITE(3,BMT) (X(J),J=1,IOUT)
      GO TO 10
100   STOP
      END
```

If you are submitting a research article other than notes or comments, I would like to suggest that you use the following format, as much as possible:

Title
Author and affiliation
Indented abstract (entire manuscript should be single spaced)
Introduction (purpose—short review of literature, etc.)
Method
Results
Discussion (conclusion)
References

All manuscripts should be sent to the editor at the above address. (All manuscripts should be camera-ready copy.)

It is the policy of the sig=multiple linear regression and of *Viewpoints* to consider for publication articles dealing with the theory and the application of multiple linear regression. Manuscripts should be submitted to the editor as an original, single-spaced typed copy. A cost of $1 per page should be sent with the submitted paper. Reprints are available to the authors from the editor. Reprints should be ordered at the time the paper is submitted and 20 reprints will cost $.50 per page of manuscript. Prices may be adjusted as necessary in the future.

"A publication of the *Multiple Linear Regression Special Interest Group* of the American Educational Research Association, published primarily to facilitate communication, authorship, creativity, and exchange of ideas among the members of the group and others in the field. As such it is not sponsored by the American Educational Research Association nor necessarily bound by the Association's regulations.

"Membership in the *Multiple Linear Regression Special Interest Group* is renewed yearly at the time of the American Educational Research Association Convention. Membership dues pay for a subscription to the *Viewpoints* and are divided into two categories: individual=$3.00; and institutional (libraries and other agencies)=$12.50. Membership dues and subscription requests should be sent to the Executive Secretary of the MLRSIG."

THE UNIVERSITY OF AKRON
AKRON, OHIO 44325

157PD 78
SPANER STEVEN D.
BEHAVIORAL STUDIES
UNIV OF MO-ST LOUIS
ST LOUIS, MISSOURI 63121

BOOKS - SPECIAL 4th CLASS RATE

TABLE OF CONTENTS