United States
Department of
Agriculture

Forest
Service

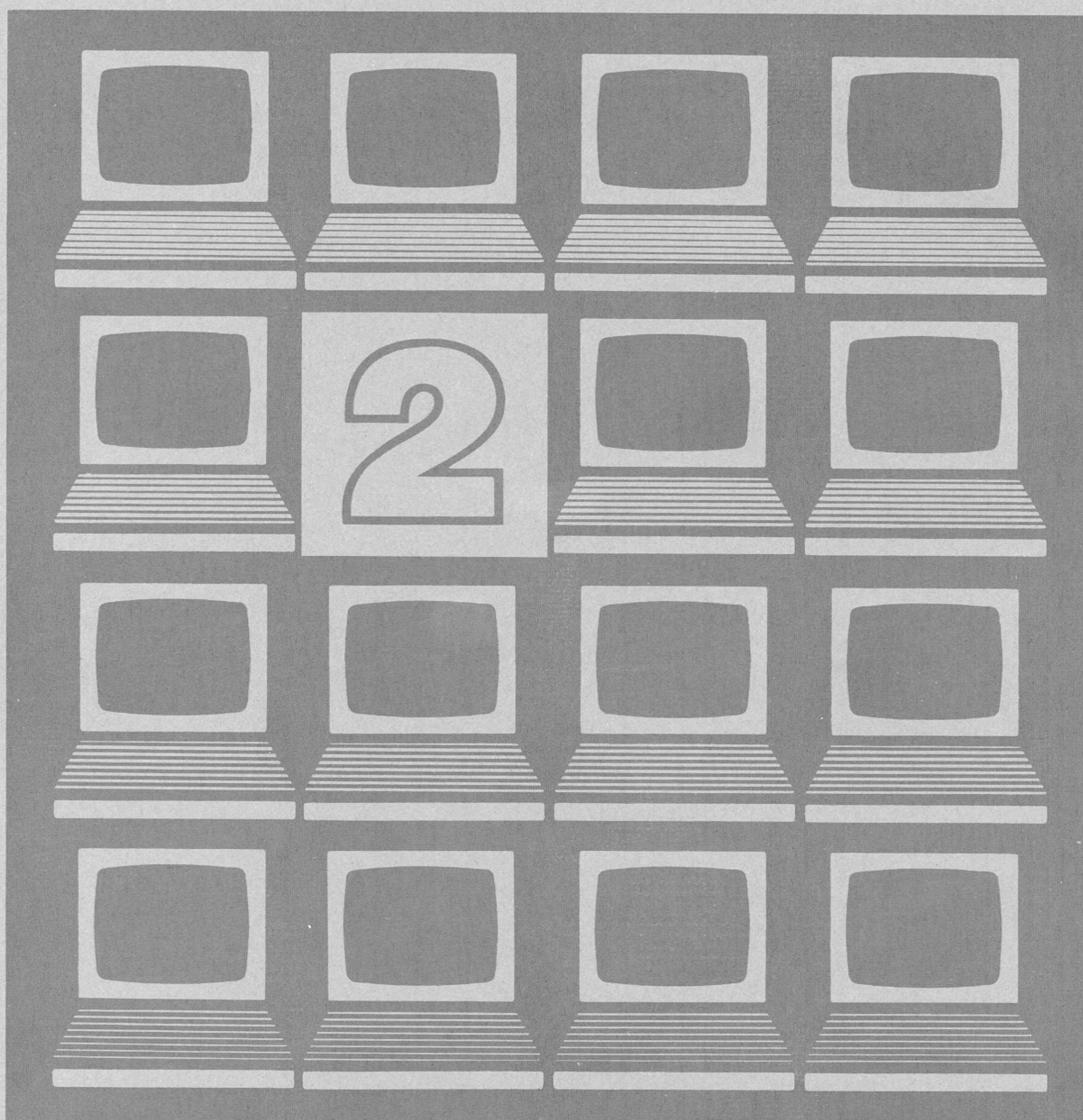
North Central
Forest Experiment
Station

General Technical
Report **NC-85**

The Microcomputer Scientific Software Series 2

General Linear Model - Regression

Harold M. Rauscher



CONTENTS

North Central Forest Experiment Station
Forest Service—U.S. Department of Agriculture
1992 Folwell Avenue
St. Paul, Minnesota 55108
Manuscript approved for publication January 20, 1983
September 1983

THE MICROCOMPUTER SCIENTIFIC SOFTWARE SERIES 2: GENERAL LINEAR MODEL—REGRESSION

Harald M. Rauscher, *Research Forester, Grand Rapids, Minnesota*

GENERAL INFORMATION

Identification

Name:	General Linear Model–Regression (GLMR)
Location	North Central Forest Exp. Station 1831 Highway 169 East Grand Rapids, MN 55744
Language:	Microsoft BASIC
Computer:	Radio Shack TRS-80 Microcomputer[1]

Summary

General

The general linear model regression (GLMR) program provides the microcomputer user with a sophisticated regression analysis capability. GLMR is a member of the SPS family of programs (Buhyoff *et al.* 1980). It provides the utmost generality regarding choice of models. The output provides a regression ANOVA table, estimators of the regression model coefficients, their confidence intervals, confidence intervals around the predicted Y-values, residuals for plotting, a check for multicollinearity, a check for autocorrelation, and the scaled regression coefficients. A plotting routine is part of the regression program to facilitate quick plotting of residuals.

[1]*Mention of trade names does not constitute endorsement of the products by the USDA Forest Service.*

Documentation

This manual will explain the operation of GLMR, detail how the program functions in enough depth to allow a knowledgeable user to implement improvements and extensions, and present the complete source code listings.

Objectives

In this paper I have two objectives:

(a) Provide small computer users with a flexible and general purpose regression analysis package.

(b) Provide user instructions in enough detail to allow implementation of extensions to the basic system.

Assumptions

I assume that the users of GLMR are also programmers. In my opinion, the opportunity cost of owning a small computer system without being able to program it is too high. Consequently, this document is aimed at users who understand the operating environment of their small computer, can program their small computer in its resident BASIC dialect, and have some understanding of the differences between the many dialects of BASIC in use.

Furthermore, I assume that users have enough statistical knowledge to intelligently use regression analysis. This manual does not teach regression analysis. The theoretical basis for regression analysis is presented in some depth to allow the informed reader to judge the mathematical approach used as the basis of this program.

References

Bahr, D. E. Discrete model simulation program. Interface Age 4(8): 94-101; 1979.

Buhyoff, G. J.; Rauscher, H. M.; Hull IV, R. B.; Killeen, K. Microcomputer-resident comprehensive statistical analysis. Behav. Meth. & Instrumentation 12: 551-553; 1980.

Draper, N. R.; Smith, H. Applied regression analysis. New York: John Wiley & Sons; p. 407.

Durbin, J.; Watson, G. S. Testing for serial autocorrelation in regression. Biometrika 37: 409-438; 1950.

Durbin, J.; Watson, G. S. Testing for serial autocorrelation in regression II. Biometrika 38: 159-178; 1951.

Durbin, J.; Watson, G. S. Testing for serial autocorrelation in regression III. Biometrika 58: 1-19; 1971.

Forsythe, G. E.; Malcolm, M. A.; Moler, C. B. Computer methods for mathematical computations. New Jersey: Prentice Hall Inc.: p 259; 1977.

Golub, G. H.; Reinsch, C. Singular value decomposition and least squares solutions. Numer. Math. 14: 403-420; 1970.

Mather, P. M. Computational methods of multivariate analysis in physical geography. New York: John Wiley & Sons, 1976; p. 532.

Nash, J. C. Compact numerical methods for computers. New York: John Wiley & Sons: 1979; 227 p.

Rauscher, H. M. The microcomputer scientific software series 1: Numerical information manipulation system Gen. Tech. Rep. NC-84. St. Paul MN: U.S. Department of Agriculture, Forest Service, North Central Forest Experiment Station; 1983. p.

Searle, S. R. Linear models. New York: John Wiley & Sons: 1971; p. 532.

Snedecor, G. W.; Cochran, W. G. Statistical methods. Ames, IA: Iowa State Univ. Press: 1967. p. 593.

Sokal, R. R.; Rohlf, F. J. Biometry The principles and practice of statistics in biological research. San Francisco, CA: W. H. Freeman and Co.: 1969. p. 776.

Quick Reference Guide

Input requirements

There are two input requirements: an m (number of observations) by 1 matrix containing the observations on the dependent variable and an m by n (number of variables) matrix containing the corresponding independent data. When an intercept term is required in the regression equation, the first variable of the independent data matrix must be a column of 1's. When no intercept term is desired, i.e. the line is forced through the origin, the column of 1's must be absent.

Regression analysis options

The following 10 regression analysis options are available:

(a) ANOVA table with regression coefficients,
(b) test for homogeneity of variances,
(c) confidence intervals for the coefficients,
(d) confidence intervals for the estimates of the dependent variable,
(e) plotting of residuals,
(f) check for multicollinearity,
(g) store the (X'X) inverse matrix on disk,
(h) check for autocorrelation,
(i) scaled regression coefficients, and
(j) test for equality of regression coefficients.

APPLICATION
Introduction

This section of the guide will teach how to properly use GLMR. Each option in the system will be presented, discussed, and illustrated by example.

For the purposes of linear regression, a model is linear in the parameters for any function of the independent variable as long as these functions do not involve the coefficients b_i. The general form of a linear model is (Mather 1976)):

$$Y = b_0 + b_1 A_1 + \ldots + b_k A_k \qquad (1)$$

where: $A_k = f(x_1, x_2, \ldots, x_k)$

This program requires an input Y, the m by 1 matrix of observations on the dependent variable. Due to memory size constraints, the program is dimensioned to accept no more than 200 observations. The X matrix must be an m by n matrix in which the columns of X are the observations on the independent variable(s). Missing values, coded -99.99, are not allowed in matrices Y and X. The program is constrained to accept no more than 10 independent variables when no intercept term (b_0) is in the model. When b_0 is used, only 9 independent variables are permitted. All dimensions are in line number 40000 and may be readily changed to meet changing needs consistent with the available maximum memory of the hardware. For example, a problem with only 50 observations could use up to 40 independent variables in the model equation. If an intercept term (b_0) is in the model, then the first column of X must contain all 1's. How the values of the independent variables are computed is irrelevant to the program. This allows maximum design flexibility for the analyst.

To illustrate the use of this program, I will follow an example from the widely available text on regression analysis by Draper and Smith (1966, p. 362) (table 1). Load the disk with the programs on the disk drive which must have the operating system present (drive 0) and a formatted data diskette with no operating system present on drive 1. Run program "GLR/BAS". The menu of choices wil present itself (fig. 1). Select number 5 to enter matrix Y from the keyboard. You will be asked for the number of rows (m) in the matrix and the number of columns (1) to enter <25,1>. Next you will be asked to give the first column a name, enter <Y>. After the 25th element has ben entered, you are automatically returned to the menu.

Select number 6 to display the Y matrix. Notice that only 13 of the 25 columns of Y are displayed at one time. Pressing the down arrow on the keyboard displays the next "page" of data. Pressing the up arrow returns the first page to the screen. When a matrix has more than 4 columns, only four columns will be displayed on any given "page". The left and the right arrows move the "page" across the desired sections of the matrix. To return to the menu press the <CLEAR> button. To save the matrix Y for use with the regression program it is necessary to select number 4. You will be asked to name matrix Y. It is best not to label it "Y" because the program uses the names, Y, X, B, and XPXI to store intermediate results. You are then asked to specify the disk on which you want to store the matrix. Using the TRS-80, this should always be a disk other disk drive 0, the drive that contains the operating system. This practice ensures sufficient disk storage for the intermediate matrices. Enter the matrix X into memory the same way and save it using a name other than X.

Table 1.—*Example dependent variable (Y) and independent variables (X) from Draper and Smith (1966) p. 362*

COL:	Dependent variable Y	Independent variables X0	X8	X6
R 1 :	10.98	1	35.3	20
R 2 :	11.13	1	29.7	20
R 3 :	12.51	1	30.8	23
R 4 :	8.4	1	58.8	20
R 5 :	9.27	1	61.4	21
R 6 :	8.73	1	71.3	22
R 7 :	6.36	1	74.4	11
R 8 :	8.5	1	76.7	23
R 9 :	7.82	1	70.7	21
R 10:	9.14	1	57.5	20
R 11:	8.24	1	46.4	20
R 12:	12.19	1	28.9	21
R 13:	11.88	1	28.1	21
R 14:	9.57	1	39.1	19
R 15:	10.94	1	46.8	23
R 16:	9.58	1	48.5	20
R 17:	10.09	1	59.3	22
R 18:	8.11	1	70	22
R 19:	6.84	1	70	11
R 20:	8.88	1	74.5	23
R 21:	7.68	1	72.1	20
R 22:	8.47	1	58.1	21
R 23:	8.86	1	44.6	20
R 24:	10.36	1	33.4	20
R 25:	11.08	1	28.6	22

Once Y and X are stored on disk 1, select number 8. You are asked to enter the name of your Y matrix and of your X matrix. These two matrices are read into the active memory and the program commences. After some period of waiting, you will be asked to input the number of significant figures in the raw

MENU FOR: GENERAL LINEAR REGRESSION ANALYSIS SYSTEM

1. RETURN TO MAIN MENU
2. GOTO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. GENERAL LINEAR REGRESSION
9. BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCES

SELECT ONE OF THE ABOVE —>..

Figure 1.—*Menu of options, program GLR/BAS of the GLMR system.*

data. The presence of errors in the data and in the computational processes prevent the program from detecting when a singular value, computed as small but not zero, is in fact zero. If the matrix is less than full rank, a warning is issued (see discussion of multicollinearity below). You will be asked whether or not to include the mean in the ANOVA table. Including the mean displays the amount of the total sums of squares that has been accounted for by the mean. This sum of squares is not evident in the "usual" ANOVA table in which the mean is not included.

The ANOVA display presents the "usual" results (fig. 2). The estimates of the regression coefficients are given at the bottom of the list of output. The ANOVA table will persist until you depress the <SPACE> bar. You are queried about the mean again and a positive response will display the ANOVA table with the mean included (fig. 3). Finally, you are asked whether you want to continue the analysis by computing confidence intervals or performing error analysis. If the answer to the above question is yes, the pertinent matrices are saved on disk 1 and the second part of the program is moved into memory and activated.

The menu for part II will automatically appear (fig. 4). The Y matrix at this point shows four columns—the original Y values, the estimated YHAT values, the residuals (EHAT), and the normalized residuals (ZHAT) (table 2). The confidence intervals of the coefficients will result if selection number 8 is chosen (table 3). The test for $b_i = 0$ is easilly made by seeing whther or not 0 falls within the confidence limits of b_i. The standard deviation, which is needed to test the equality of regression coeffficients, is presented with this option. Choosing selection number 9 produces the confidence intervals around the predicted Y values (fig. 5). If you will want to plot the confidence band around the regression line, save the augmented Y matrix when queried.

Some indicators in evaluating multicollinearity are displayed upon choosing selection number 11 (fig. 6). The condition number of X can be used as an index of collinearity (Forsythe *et al.* 1977). If cond(X) is close to 1, the columns of X are very independent. If cond(X) is large, the colums of X are nearly dependent. A matrix A may be considered to be more singular than a matrix B if cond(A) > cond(B). The collinearity number is that condition number that

REGRESSION ANALYSIS OF VARIANCE TABLE

SOURCE	DF	SS	MS	F	PROBABILITY
REG	2	54.1858	27.0929	61.9027	0.0000
ERROR	22	9.62872	.437669		
TOTAL	24	63.8145			

MEAN OF RESPONSE VARIABLE Y = 9.424
STD. DEV. OF ERROR = .661566 C.V. of Y = 7.0
MULTIPLE CORRELATION COEFFICIENT (R**2) = .849115
REGRESSION COEFFICIENTS
B O = 9.12692 B 1 = -.0723934 B 2 = .202815

Figure 2.—*ANOVA display for the example problem without a line for the "MEAN".*

REGRESSION ANALYSIS OF VARIANCE TABLE

SOURCE	DF	SS	MS	F	PROBABILITY
MEAN	1	2220.3	2220.3	5073	0.0000
REG.	2	54.1858	27.0929	61.9027	0.0000
ERROR	22	9.62872	.437669		
TOTAL	25	2284.11			

MEAN OF RESPONSE VARIABLE Y = 9.424
STD. DEV. OF ERROR = .661566 C.V. of Y = 7.0
MULTIPLE CORRELATION COEFFICIENT (R**2) = .849115
REGRESSION COEFFICIENTS
B O = 9.12692 B 1 = -.0723934 B 2 = .202815

Figure 3.—*ANOVA display for the example problem with a line for the "MEAN".*

MENU FOR: GENERAL LINEAR REGRESSION ANALYSIS SYSTEM
PART II. CONFIDENCE INTERVALS AND ERROR ANALYSIS

1. RETURN TO MAIN MENU
2. GOTO BASIC
3. READ A MATRIX FROM DISK
4. WRITE A MATRIX TO DISK
5. ENTER A MATRIX MANUALLY
6. DISPLAY MATRIX ON VIDEO
7. PRINT MATRIX
8. CONFIDENCE INTERVALS FOR B
9. CONFIDENCE INTERVALS FOR YHAT
10. PLOT RESIDUALS
11. MULTICOLLINEARITY CHECK
12. HETEROSCEDASTICITY CHECK
13. AUTOCORRELATION CHECK
14. SCALED B
15. EQUALITY OF REG. LINES

SELECT ONE OF THE ABOVE —>..

Figure 4.—*Menu of options program GLRI/BAS of the GLMR system.*

would cause the matrix X to be rank deficient. If your original input to the number of significant digits in the data produces a less than full rank matrix, run the program again with more digits specified as significant. If the matrix is then of full rank, then you need to be wary in the use of the coefficients. The estimates of B will still be unbiased but a large standard error will result indicating that the probability of an individual b_i that is far from the true population b_i is increased (Mather 1976). This is not critical for predictive uses but would be unacceptable for explanatory purposes. One solution to problems of multicollinearity is ridge regression.

Table 2.—*The "Y" matrix for the example problem*

	Data set name —> Y			
COL:	Y	YHAT	EHAT	ZHAT
R 1 :	10.98	10.6277	.352269	.532478
R 2 :	11.13	11.0331	.0968657	.146419
R 3 :	12.51	11.5619	.948053	1.43305
R 4 :	8.4	8.92649	-.526486	-.795819
R 5 :	9.27	8.94108	.328921	.497186
R 6 :	8.73	8.4272	.302802	.457705
R 7 :	6.36	5.97181	.388187	.586771
R 8 :	8.5	8.23909	.260911	.394384
R 9 :	7.82	8.26782	-.44782	-.676909
R 10:	9.14	9.0206	.119403	.180485
R 11:	8.24	9.82416	-1.58416	-2.39457
R 12:	12.19	11.2939	.896136	1.35457
R 13:	11.88	11.3518	.528221	.798441
R 14:	9.57	10.1498	-.579822	-.876439
R 15:	10.94	10.4037	.536348	.810726
R 16:	9.58	9.67214	-.0921383	-.139273
R 17:	10.09	9.29592	.794081	1.20031
R 18:	8.11	8.52131	-.41131	-.621723
R 19:	6.83	6.29034	.539657	.815728
R 20:	8.88	8.39836	.481646	.728039
R 21:	7.68	7.96365	-.283653	-.428761
R 22:	8.47	9.17998	-.709977	-1.07318
R 23:	8.86	9.95447	-1.09447	-1.65437
R 24:	10.36	10.7653	-.405279	-.612606
R 25:	11.08	11.5184	-.438397	.662667

Option number 13 provides the Durbin-Watson statistic to test autocorrelation. In the example, this statistic has the value of 2.2. The evaluation of this statistic is presented by Durbin and Watson (1950, 1951, 1971). If significant autocorrelation is shown, three possibiities exist (Mather 1976). (1) The functional form of the model is incorrect, change it. (2) Some explanators are missing, plot the residuals against some possible candidates. If a relationship is found, include the new predictors into the model. (3) The data contain a true autoregressive structure.

Option number 14 activates the routine that calculates the standardized coefficients. These scaled coefficients allow evaluation of relative importance. The larger the scaled coefficient, the more that independent variable contributed to explaining the total variation.

Option number 12, check for heteroscedasticity, is currently not activated. Option number 10, plotting residuals, allows visual analysis for heteroscedasticity as recommended by Draper and Smith (1966). At this point, the Y matrix is augmented with the columns of the X matrix and one column for the observation numbers (table 4). If the augmented matrix exceeds 10 columns, an error will occur. Change the dimension statement in line 40000 to increase the number of columns as required by the problem.

Table 3.—*Confidence interval and standard deviation of the regression coefficient for the example problem*

	Coefficient	Upper level	Lower level	Stand. dev.
B 0	9.12692	11.4159	6.83794	1.1028
B 1	-.0723934	-.0557898	-.088997	7.99942E-03
B 2	.202815	.29781	.10782	.0457676

CONFIDENCE INTERVALS FOR PREDICTED Y VALUES T-VALUE FOR 95 TWO-SIDED CONFIDENCE INTERVAL = 2.0756 DF = 22

YHAT	UPPER CI	LOWER CI	STAND. ERR.
10.6277	11.0292	10.2262	.193436
11.0331	11.5065	10.5597	.2281
11.5619	12.047	11.0768	.233737
8.92649	9.21899	8.634	.140921
8.94108	9.26728	8.61488	.15716
8.4272	8.89789	7.95651	.226775
5.97181	6.89029	5.05333	.442512
8.23909	8.82925	7.64893	.284334
8.26782	8.69214	7.8435	.204434
9.0206	9.30658	8.73462	.13778
9.82416	10.12	9.52832	.142535
11.2939	11.7667	10.8211	.227802
11.3518	11.8353	10.8683	.232955
10.1498	10.5379	9.7617	.18698
10.4037	10.7817	10.0258	.182092
9.67214	9.95714	9.38714	.137312
9.29592	9.64742	8.94442	.169349
8.52131	8.9764	8.06623	.219254
6.29034	7.19751	5.38317	.437063
8.39836	8.96083	7.83589	.270993
7.96365	8.38516	7.54214	.203078
9.17998	9.48286	8.8771	.145926
9.95447	10.2625	9.64648	.148385
10.7653	11.1903	10.3403	.204757
11.5184	12.0024	11.0344	.233167

Figure 5.—*Confidence intervals around the predicted Y values of the example problem as viewed on the video display.*

The Y matrix is on disk, so you have lost nothing. If the variances of the residuals are not equal, then the least squares estimators B will not be of minimum variance although they will still be unbiased. Variances will be heterosedastic if they increase or decrease with any of the independent variables. Because the predicted values of Y are linear functions of the independent variables, the residuals, EHAT, can also be plotted against Y and the scatter observed for trends. If a functional relationship between the residuals and Y's can be found then weighted least sqares is a solution.

CHECK FOR MULTICOLLINEARITY

RANK (X) = 3 X HAS FULL RANK
MAXIMUM SINGULAR VALUE (X) = 292.563
MINIMUM SINGULAR VALUE (X) = .599452
CONDITION NUMBER FOR (X) = 488.051
COLLINEARITY NUMBER FOR (X) = 1E+06

IF CONDITION NUMBER IS CLOSE TO 1, X = INDEPENDENT (NONSINGULAR).
IF CONDITION NUMBER IS INFINITE, X = DEPENDENT (SINGULAR).
IF CONDITION NUMBER CLOSE TO COLLINEARITY NUMBER, X = NEARLY DEPENDENT (SINGULAR)

Figure 6.—*Multicollinearity indicators for the example problem.*

Selection of option number 10 activates the plotting part of the program. Use option number 3 and number 6 to review the Y matrix. You will see the original Y vector, the predicted Y, the errors, the normalized errors and if you chose to save them, the confidence intervals around the predicted Y. In addition, the columns of the X matrix are now columns of the Y matrix as well (table 4). Select option number 10. Answer the first question with 1 for disk and enter the name of the data matrix as Y. You are given the choice of viewing Y and reviewing the names of the variables in Y. Choose variable number 9 as the x-axis designate and the residuals, EHAT, variable number 3, as the Y-axis designate. Decline the sorting option because the X-axis is in ascending order already and the graphing status menu is displayed. (fig. 7). The X and Y axis designates are presented along with their ranges. The ranges can be changed by selecting either number 3 or number 4. Selections number 5 and number 6 allow manipulation of vertical and horizontal control lines on the plot output. Options number 7 through number 9 allow designation of title and labeling of the axes. Option number 10 executes the plotting routine (fig. 8). The form of the lineplotter output is patterned after Bahn (1979).

This regression system can also be used to test for homogeneity of variances and for the equality of regression coefficients. Option number 9 of the main regression menu (fig. 1) is used to enter the test for homogeneity of variance. Using the example on page 370 of Sokal and Rohlf (1969), the program requires as input the variances and their degrees of freedom.

Table 4.—*Augmented Y matrix for plotting*

COL:	Y	YHAT	EHAT	ZHAT	Y-UPPER	Y-LOWER	X1	X2	OBS
R1 :	10.98	10.6277	0.352269	0.532478	11.0292	10.2262	35.3	20	1
R2 :	11.13	11.0331	.0968657	.146419	11.5065	10.5597	29.7	20	2
R3 :	12.51	11.5619	.948053	1.43305	12.047	11.0768	30.8	23	3
R4 :	8.4	8.92649	-.526486	-.795819	9.21899	8.634	58.8	20	4
R5 :	9.27	8.94108	.328921	.497186	9.26728	8.61488	61.4	21	5
R6 :	8.73	8.4272	.302802	.457705	8.89789	7.95651	71.3	22	6
R7 :	6.36	5.97181	.388187	.586771	6.89029	5.05333	74.4	11	7
R8 :	8.5	8.23909	.260911	.394384	8.82925	7.64893	76.7	23	8
R9 :	7.82	8.26782	-.44782	-.676909	8.69214	7.8435	70.7	21	9
R10:	9.14	9.0206	.119403	.180485	9.30658	8.73462	57.5	20	10
R11:	8.24	9.82416	-1.58416	-2.39457	10.12	9.52832	46.4	20	11
R12:	12.19	11.2939	.896136	1.35457	11.7667	10.8211	28.9	21	12
R13:	11.88	11.3518	.528221	.798441	11.8353	10.8683	28.1	21	13
R14:	9.57	10.1498	-.579822	-.876439	10.5379	9.7617	39.1	19	14
R15:	10.94	10.4037	.536348	.810726	10.7817	10.0258	46.8	23	15
R16:	9.58	9.67214	-.0921383	-.139273	9.95714	9.38714	48.5	20	16
R17:	10.09	9.29592	.794081	1.20031	9.64742	8.94442	59.3	22	17
R18:	8.11	8.52131	-.41131	-.621723	8.9764	8.06623	70	22	18
R19:	6.83	6.29034	.539657	.815728	7.19751	5.38317	70	11	19
R20:	8.88	8.39836	.481646	.728039	8.96083	7.83589	74.5	23	20
R21:	7.68	7.96365	-.283653	-.428761	8.38516	7.54214	72.1	20	21
R22:	8.47	9.17998	-.709977	-1.07318	9.48286	8.8771	58.1	21	22
R23:	8.86	9.95447	-1.09447	-1.65437	10.2625	9.64648	44.6	20	23
R24:	10.36	10.7653	-.405279	-.612606	11.1903	10.3403	33.4	20	24
R25:	11.08	11.5184	-.438397	-.662667	12.0024	11.0344	28.6	22	25

1. X-VARIABLE = VAR. #9
2. Y-VARIABLE = VAR. #3
3. RANGE OF X-VARIABLE —> 1 TO 25
4. RANGE OF Y-VARIABLE —> -2 TO 2
5. VERTICAL CONTROL LINE STATUS —> OFF
6. HORIZONTAL CONTROL LINE STATUS ON
7. TITLE OF GRAPH —> PLOT OF EXAMPLE PROBLEM FROM DRAPER & SMITH P. 364
8. X-AXIS LABEL —> X-AXIS: OBSERVATION NUMBER
9. Y-AXIS LABEL —> Y-AXIS: RESIDUALS Y–YHAT
10. EXECUTE THE GRAPHING ROUTINE
11. GET ANOTHER MATRIX
12. REGRESSION PART II
13. DISPLAY MATRIX
14. PRINT DATA MATRIX

SELECT STATUS CHANGES OR CONTINUATION —>..

Figure 7.—*Graphing status menu for the example problem.*

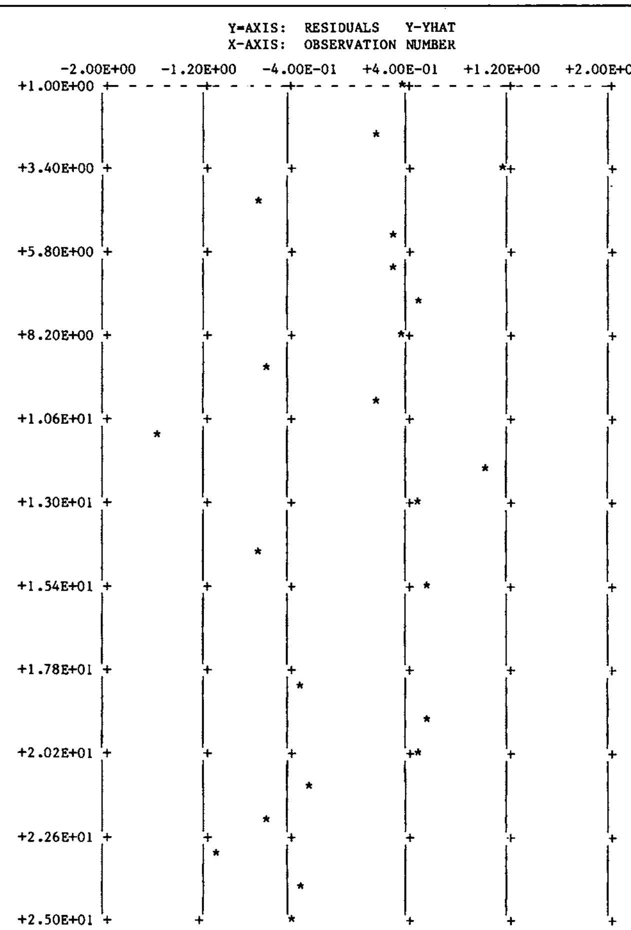


Figure 8.—*Plot of residuals for the example problem from Draper & Smith (1966).*

HOW MANY EQUATIONS ARE TO BE COMPARED? —> 2.
ARE THE VARIANCES ASSUMED HOMOGENEOUS <0> OR HETEROGENEOUS <E>?

PLEASE ENTER THE REQUESTED INFORMATION FOR EACH EQUATION.

EQUATION #1
VALUE OF B COEFFICIENT —> 3.238
STANDARD DEVIATION OF B —> 1.307
ERROR SUM OF SQUARES —> 21521
ERROR DEGREES OF FREEDOM —> 9

EQUATION #2
VALUE OF B COEFFICIENT —> 2.52
STANDARD DEVIATION OF B —> 0.284
ERROR SUM OF SQUARES —> 26874
ERROR DEGREES OF FREEDOM—> 17

WISH TO REVIEW THE INPUT? (Y/N) —> Y.

COL:	B	STAND.DEV.	ERROR SS	ERROR DF
R 1:	3.238	1.307	21521	9
R 2:	2.52	.284	26874	17

CONTINUE <C> OR REDO INPUT<R>? —> C.

Figure 9.—*Procedure for testing equality of regression coefficients.*

The values of the variances must be greater than or equal to 1 because a logarithmic transformation is used in the calculation. This program allows the user to multiply all variances by an appropriate values. Be warned that Bartlett's test is sensitive to departures from non-normality in the data. Hence, a significant qui-squared value may indicate non-normality rather than heteroscedasticity (Sokal and Rohlf 1969).

The test for equality of regression coefficients is entered from menu part II (fig. 4). Data requirements are: the values of the coefficients, their standard deviation, the error sum of squares for each regression equation, the error degrees of freedom, and whether the error variances are assumed to be homogeneous or heterogeneous. The example of this procedure comes from Snedecor and Cochran (1967, p. 435). Follow the procedure in figures 9 and 10. The computational details are presented in the "Equality of Regression Coefficients—Theory" section.

(A)

RESULTS OF THE TEST FOR EQUALITY OF REGRESSION COEFFICIENTS.

WEIGHTED AVERAGED B = 2.56786
SUM OF SQUARE FOR COMBINED B'S = 673.531
NUMERATOR DEGREES OF FREEDOM = 1
MEAN SQUARE FOR B'S = 673.531
COMBINED ERROR SUM OF SQUARES = 48395
DENOMINATOR DEGREES OF FREEDOM = 26
COMBINED ERROR MEAN SQUARE = 1861.35
THE CALCULATED F-VALUE = 361852
THE PROBABILITY OF REJECTING THE NULL HYPOTHESIS = .4407

(B)

RESULTS OF THE TEST FOR EQUALITY OF REGRESSION COEFFICIENTS

WEIGHTED AVERAGED B = 2.64824
SUM OF SQUARES FOR COMBINED B'S = 653.573
NUMERATOR DEGREES OF FREEDOM = 1
MEAN SQUARE FOR B's = 653.573
COMBINED ERROR SUM OF SQUARES = 48395
DENOMINATOR DEGREES OF FREEDOM = 24
COMBINED ERROR MEAN SQUARE = 2016.46
THE CALCULATED F-VALUE = .324119
THE PROBABILITY OF REJECTING THE NULL HYPOTHESIS = .4192

Figure 10.—*Results of test for equality of regression coefficients for the example problem: (A) Homogeneous case (B) Heterogeneous case.*

PROCEDURES AND REQUIREMENTS

System Characteristics

Program GLMR is written in BASIC for the TRS-80 microcomputer. The computer should have a minimum of 48K of user defineable random access memory (RAM), two disk drives, and a lineprinter. Highly compatible versions of BASIC are available for most microcomputer systems. It is anticipated that, with a minimum of effort, this program can be converted to execute on any computer system that can use some dialect of the BASIC language.

Performance

A problem with 25 observations and 2 independent variables plus intercept takes approximately 2 minutes to solve. One with 13 observations and 5 independent variables plus intercept takes approximately 3 minutes to solve.

Data Compatibility

Program GLMR is compatible with the data standard used by the SPS system (Buhyoff *et al.* 1980). The numerical data of interest are built into a matrix, A_{ij}, where i = number of observations and j = number of variables. Each data set has three data files automatically defined for it. If "TEST", "TESTN", and "TESTL" will be automatically created or updated every time "TEST" is saved to disk. File "TESTM" contains the matrix A_{ij}. I have found that data file manipulations are greatly simplified by using this three file organization.

Matrix A(R,C) is used to transfer data to and from disk, R and C are the row/column dimensions, and NS(C) is used to store names of the variables. The input matrices required by this program may be created by GLM or by the numerical information manipulation system (NIMS) reported by Rauscher (1983). The output data matrices generated by GLMR are compatible with all of the above referenced programs.

Availability

Owners of Model I TRS-80 single sided, single density, 5-1/4″ disk based systems who desire a machine readable copy of this program must send a blank diskette to the author.

Regression Theory

GLMR is based upon the theory of general linear models as presented by Searle (1971). The problem in linear regression is to find a solution to:

$$Y = X\,B + E \tag{2}$$

where:

Y = the m by 1 matrix of observations on the dependent variable
X = the m by n matrix of observations on the independent variables
B = the n vector of parameters
E = the n vector of errors
r(X) = n; X has full rank

subject to the constraint that the sum of the squares of the errors is minimized. The solution to (2) is:

$$\hat{B} = (X'X)^{-1}\,X'Y \tag{3}$$

where:

$\hat{B}$ = the n vector of estimates of parameters
X' = transpose of X
$(X'X)^{-1}$ = inverse of X'X

The most reliable method for computing the coefficients ($\hat{B}$) for the general least-squares problem is known as singular value decomposition (Forsythe *et al.* 1977). Although other algorithms may require less computer time and storage, they are less effective in dealing with errors in the data, runoff errors, and linear dependence of X. The singular value decomposition algorithm of Golub and Reinsch (1970) was used as the basis for solving the least squares problem in this program. A singular value decomposition computes a factorization of the form:

$$X = U\,S\,V' \tag{4}$$

where:

S = m by n diagonal matrix of singular values if $S_{ii} < q$ then $S_{ii} = 0$
S_{ii} = ith singular value
q = a tolerance value based on the precision of the data in the X matrix
U = an m by m orthogonal matrix
V = an n by n orthogonal matrix.

and

$$\hat{B} = (X'X)^{-1}\,X'\,Y = V\,S^{+}U'Y \text{ where } S^{+}_{ii} = 1/S_{ii} \tag{5}$$

(Forsythe *et al.* 1977)

$$(X'X)^{-1} = V\,(S^{+})^{2}\,V' \text{ (Nash 1979)} \tag{6}$$

Using these results as the basis, the regression estimators, variances, and sums of squares can be computed as outlined in Searle (1971):

$$\hat{E}(Y) = Y = X B \quad (7)$$

where:

$\hat{E}(Y) = Y$ = estimates of Y

$E'E$ = error sum of squares = SSE (8)

$(\text{sigma})^2 = \text{SSE}/(m\text{-}r(X))$ (9)

where:

$(\text{sigma})^2$ = residual error estimate of variance

m = number of observations

r(X) = rank of X

SST = Y'Y = total sum of squares (10)

SSR = SST -SSE = sum of squares due to regression (11)

SSM = $m * \bar{Y}^2$ = correction term for the mean (12)

where:

$\bar{Y}$ = mean of Y

SSRm = SSR−SSM = regression sum of squares corrected for the mean (13)

SSTm = SST−SSM = corrected sum of squares of dependent variable Y (14)

R^2 = SSR/SST no intercept model b_0 (15)
= SSRm/SSTm intercept model $b_0 <> 0$

where:

R^2 = multiple correlation coefficient

$\text{var}(\hat{B}) = (X'X)^{-1} = V(S^+)^2 V'$ (16)

where:

$\text{var}(\hat{B})$ = variance of parameter estimators

$\text{var}(\hat{Y}) = X (X'X)^{-1} X' = X V S^+ U'$ (17)

where:

$\text{var}(\hat{Y})$ = variance of estimates of Y.

Regression Assumptions

A linear regression analysis may be used for predicting values of the dependent variable or as part of the process of scientific explanation of the variation in the dependent variable as accounted for in terms of the independent variables (Mather 1976). The assumptions needed for both of these purposes of regression were discussed in detail by Mather (1976) and they are summarized here.

1. The mean of the residuals is zero. This assumption implies that no important explanatory variable has been omitted in the model and that the chosen independent variables represent the major controls on the variability of the dependent variable.
2. The variance of the residuals and the variance of the dependent variable are constant at each level of the independent variable. This is the assumption of homoscedasticity.
3. The independent variables are nonrandom and are measured without error.
4. The independent variables are not multicollinear, that is, they are not perfectly linearly related.
5. The number of observations, m, exceeds the number of independent variables, n. The situation where m = n defines the general linear equations problem. The situation where m < n produces the mathematical programming (optimization) problem.
6. The residuals are independent. Dependence shown by similarity of adjacent values is known as autocorrelation.
7. If statistical tests of significance are to be used, the conditional distribution of the dependent variable given the independent variables should be normal.

When least squares regression is used as a predictive tool, only assumptions 4 and 5 need be true to produce acceptable estimates for the coefficients. If scientific explanation is the goal all the assumptions except 7 must hold. Assumption 7 is required whenever tests of significance are carried out or intervals are estimated.

Equality of Regression Coefficients—Theory

The procedures used as the theoretical base for this subroutine were presented in a college course in applied statistics by Dr. R. G. Krutchkoff at Virginia Polytechnic Institute and State University, Blacksburg. In the homogeneous case, the following equations are used:

$$\text{SSE} = \sum_i \text{SSE}_i \quad (18)$$

where:

SSE = combined error SS

SSE_i = ith equation error SS

$$\text{DFE} = \sum_i \text{DF}_i \quad (19)$$

where:

DFE = combined Deg. Freedom

DF_i = ith equation DF

$$\text{MSE} = \text{SSE}/\text{DFE} \quad (20)$$

$$\text{MSE}_i = \text{SSE}_i/\text{DF}_i \quad (21)$$

$$\text{SSX}_i = \text{MSE}_i/\text{DF}_i \quad (22)$$

$$D_i = 1/\text{SSX}_i \quad (23)$$

$$BB = \sum_i(B_i/D_i)/\sum_i(1/D_i) \quad (24)$$
$$= \sum_i(B_i * SSX_i)/\sum_i SSX_i$$

where

B_i = i'th regression coefficient

$$SSB = \sum_i(B_i-BB)**2/D_i) \quad (25)$$
$$= \sum_i (B_i-BB) **2 * SSX_i$$

$$DF = K-1 \quad (26)$$

where:

K = no. of coefficients compared

$$MSB = SSB/DF \quad (27)$$

$$FB = MSB/MSE \quad (28)$$

For the heterogeneous case, the computations are as follows:

SSE see above (18)

$$DFE = (SSE)^2 /\sum_i(SSE_i^2 / DF_i) \quad (29)$$

Round DFE down to the nearest integer

MSE see above (20)

$$BB = \sum_i(B_i/S2B_i) /\sum_i(1/S2B_i) \quad (30)$$

where:

$S2B_i$ = ith coefficient variance; standard deviation squared

$$SSB = (\sum_i(B_i-BB) ** 2/S2B_i)) * MSE \quad (31)$$

DF same as above (26)

MSB same as above (27)

FB same as above (28)

Program Code Documentation

The programs are coded in the Microsoft (TM) BASIC language for direct use with the Radio Shack (TM) Level II BASIC interpreter. Unlike many other languages, BASIC allows concatenation of several logical statements in one physical line. Microsoft BASIC uses the colon, ":", as the logical statement delimeter. Because each physical line costs 5 bytes of memory and each unnecessary space 1 byte, small computer BASIC programs are written in compressed code and appear impossible to decipher, especially for those who normally program in FORTRAN. To make matters worse, internal program documentation is reduced to a minimum in BASIC because these statements use precious memory. Given these memory constraints and the ability to refer unambiguously to any physical line in the program, I believe that adequate documentation of BASIC programs should be accomplished external to the code. Consequently, the code is presented in the Appendices in the normal, compressed fashion. Each section of code is explained in the following presentation by reference to physical linenumbers.

GLMR is composed of three separate programs: GLR/BAS, GLR1/BAS, and MULTPLT/BAS. The source code listings for these three programs are given in Appendices I, II, and III, respectively.

Program GLR/BAS

Linenumbers 13 to 695 comprise the same standard SEED program that was described in detail by Rauscher (1981). The general linear regression program, menu option number 8, begins with linenumber 2000. The matrix with the dependent variable, Y, and the matrix with the independent variable(s), X, must be resident on a data diskette prior to entering the regression program. Lines 2030 to 2043 request loading of the Y and X matrices. In addition, in line 2030, I transfer the matrix of dependent observations from the input/output matrix A(.,.) to the holding matrix Y(.,.), compute the total sum of squares, TSS, and the sum, MSS, of Y.

Line 2045 branches to the singular value decomposition subroutine located at line 1000. For a detailed presentation of the theory of least squares and the singular value decomposition see Forsythe *et al.* (1977). Lines 1000 to 1730 are a direct translation from the FORTRAN program SVD in chapter 9, pages 193 to 235 of Forsythe *et al.* (1977). The input to SVD is the A matrix which contains the independent variables. Upon return from SVD, W(.) contains the n unordered singular values of matrix A(.,.), and matrices U(.,.) and V(.,.) contain the orthogonal column vectors of U and V of the singular value decomposition (Forsythe *et al.* 1977). IE equals zero on a normal return from SVD, or else it is equal to K signifying that the K-th singular value has failed to converge after 30 iterations. This error is checked in line 2050 and fatally disrupts the program when IE does not equal to zero.

Lines 2060 to 2070 define WI as the value of the minimum singular value and WM the maximum, zero array B(.), and compute TA, which controls the effective rank in line 2080. The user is prompted to enter the number of significant digits in his or her raw data which is used to compute TA. This procedure is necessary to allow the computer to decide when a singular value is legitimately zero. In lines 2080 to 2100, I test for zero singular values (2080), set RR to the rank of the matrix, and calculate the regression coefficients in B(.).

In 2102 I check for singularity (RR < N where N is number of columns in matrix) warn the user of this fact if true, and save the minimum singular value in B(N + 6) if the matrix is nonsingular or save -1 if singular. In location B(N+4) I save the scale factor, RE, for computing effective rank, B(N+5) is loaded with the maximum singular value, and B(N+7) is equated to the overall mean of the dependent variable (line 2104). The array B is simply a convenient place to store these miscellaneous data. Next, I compute the error sum of squares (line 2110), the sum of squares of the mean MSS, and the standard error of the mean SE (line 2120). Notice that variable MSS has been redefined and used again. In line 2120 I compute the predicted Y values YHAT, the error deviations EHAT, and the normalized error deviations ZHAT, and save them in column 2, 3, and 4 of matrix Y(.,.), respectively.

I then compute the sum of squares due to regression RSS, the regression sum of squares corrected for the mean RMSS, the corrected total sum of squares TMSS, and set the variable MEANS equal to TRUE if the first column of the independent matrix is a column of 1's (line 2030). A column of 1's signifies that an intercept coefficient is present in the regression model. Without a column of 1's, MEANS is set to FALSE and line 2160 is executed. Rank is increased by 1, the Ftostatistic is computed FR, and the subroutine FPROB (lines 10000 to 10500) is called to compute the critical value. The code variables for the ANOVA display are initialized and the program jumps to lines 2220 through 2345 to print the ANOVA table on the video display.

If MEANS is TRUE, the user is asked whether the term for the mean should be explicitly included in the ANOVA table or not. Lines 2190 and 2200 compute the necessary statistics and load the display variables with their values. Lines 2380 to 2410 saves the Y, $(X'X)^{-1}$, and B matrices to disk prior to loading the second program, GLRI. B(N + 3) contains the rank, B(N+8) holds the number of coefficients N, and B(N + 1) contains the error mean square.

Menu selection number 9 executes Bartlett's test for the homogeneity of variance. This subroutine implements the procedure presented by Sokal and Rohlf (1969, p. 370). The calculations are straightforward enough not to require additional documentation.

Program GLRI/BAS

Lines 13 to 695 are the same as the above mentioned SEED program. Lines 700 to 920 manage the menu display and operation. The variable FMS (line 905) is a flag that, if FALSE, causes the intermediate data matrices Y, X, B, and $(X'X)^{-1}$ to be loaded from disk into memory (lines 10000 to 10010). Once these matrices have been loaded from disk, FMS turns TRUE, thus bypassing the subroutine at line 10000.

Choosing Option number 8 leads to the confidence interval calculations for the regression coefficients (lines 1000 to 1090). The subroutine at lines 11000 to 11090 calculates the T-table value for a two sided confidence interval. This subroutine also calculates and displays the standard error for each regression coefficient. The standard error is required input for testing equality of regression lines (Option no. 15).

The confidence interval around the predicted Y-values is computed in lines 2000 to 2130. Lines 2127 to 2130 allow the option to save the augmented Y matrix to disk for use with the plotting routine. Lines 3000 to 3080 merge the X matrix with the Y matrix by columns and save the combined matrix to the data disk which must reside on disk number 1. This is done to make the necessary column variables available to the plotting program, which is automatically called in line 3080.

Lines 4000 to 4140 calculate and explain the check for multicollinearity. Lines 5000 to 5040 mark the location of a heteroscedasticity test that will be implemented at some future data. The code for the autocorrelation check (lines 6000 to 6090) and the scaled B values (lines 7000 to 7130) is self explanatory.

Lines 8000 to 8230 contain the code to test for the equality of regression coefficients. Lines 8022 to 8065 guide the user and request the needed data input. If the error variances are assumed homogeneous, lines 8060 to 8090 are executed. If they are assumed heterogeneous, lines 8200 to 8230 are executed. The results are displayed in lines 8100 to 8140.

Program MULTPLT/BAS

Although the common input/output routines used in this program do not use the format of the SEED program that is used in GLR and GLR1, the function and structure of the routines are nearly the same. Lines 2 to 18 contain a TRS-80 specific formatted input routine functionally similar to that described in the SEED program by Rauscher (1981). Lines 19 to 20 and 21 to 24 contain the disk read and disk write routines, respectively. Lines 26 to 31 contain a video display routine. The dimension statements used by this program are in lines 124, 130, and 131. Lines 34 to 43 solicit the required input information from the user. If data are not sorted, lines 46 to 64 perform a matrix sort. Rows of the matrix are sorted according to ascending values of the column designated as the X-axis. The graphing status control menu is displayed in lines 66 to 74. Lines 75 to 102 contain the code for executing the plotting function on a lineprinter. This routine was adapted from Bahr (1979). Lines 103 to 118 perform the graphing status manipulations in the menu.

Lines 119 to 123 allow a function, defined in line 122, to generate the data to be plotted. The beginning, end, and interval values for the independent variable are used in a loop (121 to 123) to calculate the values of the dependent variable.

APPENDIX I. Source code listing for program GLR/BAS

```
10 CLEAR 300
   :TX=-1
   :GOTO700
   :'GLR/BAS  VERSION 10-21-80
13 '************************************************
14 'ANSWER INPUT ROUTINE
15 P=( 256*PEEK( 16417 )+PEEK( 16416 ) )-15359
   :IF ABS( FL )=1THEN16ELSE24
16 IN$=""
   :PRINT@P,CHR$( 136 );
   :LX=1
17 IN$=INKEY$
   :IFIN$=""THEN22
18 IFFL=-1 AND ASC( IN$ )>47 AND ASC( IN$ )<58 THEN 21
19 IF FL=1 AND ASC( IN$ )>64 AND ASC( IN$ )<91 THEN 21
20 GOTO 22
21 PRINT@P,IN$
   :RETURN
22 LX=LX+1
   :IFL<16THEN17
23 PRINT@P," ";
   :FORLX=1TO25
   :NEXT
   :GOTO16
24 PRINT@P, CHR$( 136 );
   :IN$=""
   :LX=ABS( FL )
   :GOSUB34
25 A$=INKEY$
   :IFA$=""THEN25
26 IFA$=CHR$( 13 )THENPRINTCHR$( 15 );
   :RETURN
27 IF A$=CHR$( 8 )THENIFIN$<>""THENPRINTA$;
   :PRINTCHR$( 136 );CHR$( 24 );
   :IN$=LEFT$( IN$,LEN( IN$ )-1 )
   :LX=LX+1
   :GOTO25
   :ELSEGOTO28
28 IFA$=CHR$( 24 )THENPRINTCHR$( 29 );
   :GOTO24
29 IF FL>0 AND ASC( A$ )>31 AND ASC( A$ )<91 THEN 32
30 IF FL<0 AND ASC( A$ )>=45 AND ASC( A$ )<58 THEN 32
31 GOTO 25
32 IF LX=0 THEN 25
```

```
33 PRINT A$;
   :IN$=IN$+A$
   :LX=LX-1
   :GOTO25
34 FOR IX=1TOLX-1
   :PRINT CHR$(136);
   :NEXT
   :FORJX=1TOLX
   :PRINT CHR$(24);
   :NEXT
   :RETURN
50 '*************************************
52 'CHANGE VARIABLE NAMES
54 '/////////////////
56 CLS
   :PRINT@448,"WHICH COLUMN'S NAME WILL BE CHANGED? --> ";
   :FL=-2
   :GOSUB15
   :XJ=VAL(IN$)
   :PRINT
   :PRINT"PLEASE ENTER NEW NAME --> ";
   :FL=8
   :GOSUB15
   :N$(XJ)=IN$
   :RETURN
100 '*******************************************
110 'READ SEQUENTIAL DATA SET FROM DISK
120 '///////////////////
125 CLS
130 PRINT@448,"ENTER FILENAME OF DESIRED DATA SET --> ";
   :INPUT F$
135 FF$=F$+"N"
   :L$=F$+"L"
   :F$=F$+"M"
140 OPEN "I",1,FF$
   :INPUT#1,R,C
   :CLOSE1
   :'GET R,C VALUES
145 OPEN "I",1,L$
   :FORK=1TOC
   :INPUT#1,N$(K)
   :NEXTK
146 CLOSE
150 OPEN "I",1,F$
   :
160 FORI=1TOR
   :FORJ=1TOC
   :INPUT#1,A(I,J)
   :NEXTJ,I
170 CLOSE 1
180 RETURN
190 '**********************************************
200 ' WRITE A SEQUENTIAL MATRIX TO DISK
210 '////////////////////
220 CLS
   :PRINT@448,"ENTER FILENAME OF NEW DATA SET --> ";
   :INPUT F$
   :PRINT
   :PRINT"ENTER DISK DRIVE # WHERE DATA SHOULD BE SAVED --> ";
   :INPUT DD$
222 FF$=F$+"N:"+DD$
   :L$=F$+"L:"+DD$
```

```
    :F$=F$+"M:"+DD$
225 FOR JQ=1TOC
    :IF LEN(N$(JQ))=0 THEN N$(JQ)=STR$(JQ)
226 NEXT
230 OPEN "O",1,FF$
    :PRINT#1,R;C;
    :CLOSE1
240 OPEN "O",1,F$
    :
250 FORI=1TOR
    :FORJ=1TOC
    :PRINT#1,A(I,J);
    :NEXTJ,I
260 CLOSE 1
265 OPEN"O",1,L$
    :FORI=1TOC
    :PRINT#1,N$(I);",";
    :NEXT
    :CLOSE
270 RETURN
290 REM ************************************************
295 REM                 SUBROUTINE INPUT
300 '1.  THIS SUB READS FROM THE KEYBOARD
305 '2.  MATRIX A IS LOADED AND CAN BE SAVED ON DISK
325 '6.  14-APR-79
335 REM ************************************************
340 CLS
345 INPUT "PLEASE ENTER: NO. ROWS AND COLS (R,C).--> ";R,C
350 FL=8
    :PRINT
    :FOR J=1TOC
    :PRINT"ENTER THE NAME FOR VARIABLE ";J;" --> ";
    :GOSUB15
    :N$(J)=IN$
    :PRINT
    :NEXT
355 CLS
365 FOR I=1 TO R
370 PRINT "ROW ";I
375 FOR J=1 TO C
380 FL=-12
    :PRINT "PLEASE ENTER:  VALUE OF COLUMN ";J" --> ";
    :GOSUB 15
385 PRINT
390 A(I,J)=VAL(IN$)
395 NEXT J
400 NEXT I
430 RETURN
435 END
440 '**********************************************
445 '          SUBROUTINE VIDEO
450 '  DISPLAYS MATRIX A(R,C),13 ROWS BY 4 COLUMNS AT ONCE
455 ' USES ARROWS IN ALL DIRECTIONS TO SEE DIFFERENT PAGES
460 ' OF DATA.  "CLEAR" WILL TERMINATE DISPLAY
500 '************************************************
510 CLS
    :IFR=0 OR C=0 THEN RETURN ELSE JJ=INT(C/4.1+1)
    :II=INT(R/13.1+1)
515 I=1
    :J=1
525      IF R-13*I>=0.0 THEN N=13 ELSE N=13+(R-13*I)
535      IF C-4*J>=0.0 THEN L=4 ELSE L=4+(C-4*J)
```

```
540 CLS
    :PRINT"COL:";
    :FOR KK=0TO(L-1)
    :PRINT TAB(14*KK+8) N$((KK+1)+4*(J-1));
545 NEXT KK
    :PRINT
550 PRINT"------------------------------------------------------------"
555 FOR M=1 TO N
560 PRINT "R";M+13*(I-1);":";
565 FOR K=1 TO L
570 A=A(M+13*(I-1),K+4*(J-1))
575 PRINT TAB(14*(K-1)+8) A;
580 NEXT K
    :PRINT
585 NEXT M
586 A$=INKEY$
    :IFA$=""GOTO586
587 IF ASC(A$)=31THEN619ELSEIFASC(A$)=9THEN588ELSE590
588 IFJ+1>JJTHENJ=1ELSEJ=J+1
589 GOTO 525
590 IF ASC(A$)=8THEN591ELSE593
591 IF J-1<=0THENJ=JJ ELSE J=J-1
592 GOTO 525
593 IF ASC(A$)=10THEN594ELSE596
594 IF I+1>IITHENI=1ELSEI=I+1
595 GOTO 525
596 IF ASC(A$)=91THEN597ELSE599
597 IF I-1<=0THENI=IIELSEI=I-1
598 GOTO 525
599 IF ASC(A$)=64 THEN 600 ELSE 601
600 GOSUB 670
    :GOTO525
601 IF ASC(A$)=46 THEN GOSUB 50 ELSE 618
602 GOTO 525
618 GOTO586
619 RETURN
    :END
620 '**************************************************
622 'LINEPRINTER OUTPUT ROUTINE
623 IF R=0 OR C=0 THEN RETURN
624 AN$="DATA SET NAME --> "
    :CLS
    :PRINT@448,"PRESS <ENTER> WHEN PRINTER IS READY.";
    :INPUT AN
    :T$=LEFT$(F$,LEN(F$)-1)
    :T$=AN$+T$
    :GOSUB42000
    :LPRINT TAB(ZZ);T$
    :LPRINT " "
    :LPRINT " "
630 JJ=INT(C/4.1+1)
    :II=INT(R/200.1+1)
632 FORI=1TOII
634 IFR-200*I>=0.0THENN=200ELSEN=200+(R-200*I)
636 FORJ=1TOJJ
638 IFC-4*J>=0.0THENL=4ELSE L=4+(C-4*J)
640 LPRINT"COL:";
    :FORKK=0TO(L-1)
    :LPRINT TAB(14*KK+8) N$((KK+1)+4*(J-1));
642 NEXT KK
644 LPRINT" "
646 LPRINT"------------------------------------------------------------"
```

```
648 FOR M=1TON
650 LPRINT "R";M+200*(I-1);":";
652 FOR K=1TO L
654 A=A(M+200*(I-1),K+4*(J-1))
656 LPRINT TAB(14*(K-1)+8) A;
658 NEXT K
660 LPRINT
662 NEXT M
664 LPRINT " "
    :LPRINT " "
    :LPRINT " "
666 NEXT J,I
668 RETURN
670 '**********************************
671 'SCREEN PRINT ROUTINE-  ACTIVATED BY PROGRAMMED "@" ASC(64)
672 '///////////////////////
673 IF PEEK(14312)<>63 THEN RETURN
674 LPRINT STRING$(64,"+")
675 FOR N= 15360 TO 16383 STEP 64
676 A$=" "
679 FOR Q=0TO63
680 Z=PEEK(N+Q)
    :IFZ>0ANDZ<27THENZ=Z+64ELSEIFZ<32THENZ=32
681 IFZ>=127 AND Z<=191 THEN Z=42
682 B$=CHR$(Z)
683 A$=A$+B$
684 NEXT Q
685 IF PEEK(14312)<>63THEN RETURN
686 LPRINT " "
    :LPRINT A$
687 NEXT N
688 LPRINT STRING$(64,"+")
689 FORN=1TO3
    :LPRINT "  "
    :NEXT
690 RETURN
695 RUN"MAIN/BAS"
    :END
700 '************************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '////////////////////
735 CLS
    :T$="NORTHERN HARDWOODS LABORATORY, USFS, MARQUETTE, MI"
    :GOSUB42000
736 T$="////////////////////////////////////////////////////////////"
    :GOSUB42000
737 PRINT
738 T$="MENU FOR:  GENERAL LINEAR REGRESSION ANALYSIS SYSTEM
739 GOSUB42000
    :PRINT
740 IF TX=-1 THEN GOSUB 40000 ELSE 750
750 TX=1
    :PRINT"  1.  RETURN TO MAIN MENU         2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK     4.  WRITE A MATRIX TO DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY     6.  DISPLAY MATRIX ON VIDEO
780 PRINT"  7.  PRINT MATRIX                8.  GENERAL LINEAR REGRESSION
790 PRINT"  9.  BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCES"
900 PRINT
    :PRINT"SELECT ONE OF THE ABOVE --> ";
    :FL=-2
    :GOSUB15
    :AN=VAL(IN$)
```

```
    :IFAN<=0ORAN>16THEN740
910 ON AN GOSUB 695,43000,125,220,340,510,620,2000,3000
920 GOTO 735
1000 '*******************************************
1010 'SINGULAR VALUE DECOMPOSITION
1020 'FORSYTHE 1977 CH.9 PP.193-235
1030 'INPUT
    : A(M,N) WHERE M>=N
1040 'OUTPUT
    : W CONTAINS N UNORDERED SINGULAR VALUES OF A
1050 '          U & V CONTAIN THEMSELVES
1060 '/////////////////////////
1065 MV$="TRUE"
    :MU$="TRUE"
1070 IE=0
    :R1=R
    :C1=C
    :M=R
    :N=C
    :NM=R
    :FORI=1TOM
    :FORJ=1TON
    :U(I,J)=A(I,J)
    :NEXTJ,I
1080 'HOUSEHOLDER REDUCTION TO BIDIAGONAL FORM
1090 G=0
    :SCALE=0
    :AR=0
1100 FORI=1TON
    :L=I+1
    :RV1(I)=SCALE*G
    :G=0
    :S=0
    :SCALE=0
    :IFI>MTHEN1160
1110 FORK=ITOM
    :SCALE=SCALE+ABS(U(K,I))
    :NEXTK
    :IFSCALE=0THEN1160
1120 FORK=ITOM
    :U(K,I)=U(K,I)/SCALE
    :S=S+U(K,I)[2
    :NEXTK
1130 F=U(I,I)
    :SS=SGN(F)
    :G=-1*SS*ABS(SQR(S))
    :H=F*G-S
    :U(I,I)=F-G
    :IFI=NTHEN1150
1140 FORJ=LTON
    :S=0
    :FORK=ITOM
    :S=S+U(K,I)*U(K,J)
    :NEXTK
    :F=S/H
    :FORK=ITOM
    :U(K,J)=U(K,J)+F*U(K,I)
    :NEXTK
    :NEXTJ
1150 FORK=ITOM
    :U(K,I)=SCALE*U(K,I)
    :NEXTK
```

```
1160 W(I)=SCALE*G
     :G=0
     :S=0
     :SCALE=0
     :IFI>MORI=NTHEN1230
1170 FORK=LTON
     :SCALE=SCALE+ABS(U(I,K))
     :NEXTK
     :IFSCALE=0THEN1230
1180 FORK=LTON
     :U(I,K)=U(I,K)/SCALE
     :S=S+U(I,K)[2
     :NEXTK
1190 F=U(I,L)
     :SS=SGN(F)
     :G=-1*SS*ABS(SQR(S))
     :H=F*G-S
     :U(I,L)=F-G
1200 FORK=LTON
     :RV1(K)=U(I,K)/H
     :NEXTK
     :IFI=MTHEN1220
1210 FORJ=LTOM
     :S=0
     :FORK=LTON
     :S=S+U(J,K)*U(I,K)
     :NEXTK
     :FORK=LTON
     :U(J,K)=U(J,K)+S*RV1(K)
     :NEXTK
     :NEXTJ
1220 FORK=LTON
     :U(I,K)=SCALE*U(I,K)
     :NEXTK
1230 TT=ABS(W(I))+ABS(RV1(I))
     :IFTT>ARTHENAR=TT
1240 NEXTI
1250 'FORM THE V MATRIX IF MV="TRUE"
1260 IFMV$="TRUE"THEN1270ELSE1350
1270 FORII=1TON
     :I=N+1-II
     :IFI=NTHEN1320
1280 IFG=0THEN1310
1290 FORJ=LTON
     :V(J,I)=(U(I,J)/U(I,L))/G
     :NEXTJ
1300 FORJ=LTON
     :S=0
     :FORK=LTON
     :S=S+U(I,K)*V(K,J)
     :NEXTK
     :FORK=LTON
     :V(K,J)=V(K,J)+S*V(K,I)
     :NEXTK
     :NEXTJ
1310 FORJ=LTON
     :V(I,J)=0
     :V(J,I)=0
     :NEXTJ
1320 V(I,I)=1.0
     :G=RV1(I)
     :L=I
```

```
1330 NEXT II
1340 'COMPUTE U IF MU="TRUE"
1350 IFMU$="TRUE"THEN1360ELSE1470
1360 MN=N
     :IFM<NTHENMN=M
1370 FORII=1TOMN
     :I=MN+1-II
     :L=I+1
     :G=W(I)
     :IFI=0THEN1390
1380 FORJ=LTON
     :U(I,J)=0
     :NEXTJ
1390 IFG=0THEN1430
1400 IFI=MNTHEN1420
1410 FORJ=LTON
     :S=0
     :FORK=LTOM
     :S=S+U(K,I)*U(K,J)
     :NEXTK
     :F=(S/U(I,I))/G
     :FORK=ITOM
     :U(K,J)=U(K,J)+F*U(K,I)
     :NEXTK
     :NEXTJ
1420 FORJ=ITOM
     :U(J,I)=U(J,I)/G
     :NEXTJ
     :GOTO1440
1430 FORJ=ITOM
     :U(J,I)=0
     :NEXTJ
1440 U(I,I)=U(I,I)+1
1450 NEXTII
1460 'DIAGONALIZATION OF THE BIDIAGONAL FORM
1470 FORKK=1TON
     :K1=N-KK
     :K=K1+1
     :ITS=0
1480 FORLL=1TOK
     :L1=K-LL
     :L=L1+1
     :IF(ABS(RV1(L))+AR)=ARTHEN1560
1490 IF(ABS(W(L1))+AR)=ARTHEN1510
1500 NEXTLL
1510 C=0
     :S=1
     :FORI=LTOK
     :F=S*RV1(I)
     :IF(ABS(F)+AR)=ARTHEN1560
1520 G=W(I)
     :H=SQR(F*F+G*G)
     :W(I)=H
     :C=G/H
     :S=-F/H
     :IFMU$="TRUE"THEN1530ELSE1540
1530 FORJ=1TOM
     :Y=U(J,L1)
     :Z=U(J,I)
     :U(J,L1)=Y*C+Z*S
     :U(J,I)=-Y*S+Z*C
     :NEXTJ
```

```
1540 NEXTI
1550 'TEST FOR CONVERGENCE
1560 Z=W(K)
     :IFL=KTHEN1670
1570 IFITS=30THEN1720
1580 ITS=ITS+1
     :X=W(L)
     :Y=W(K1)
     :G=RV1(K1)
     :H=RV1(K)
     :F=((Y-Z)*(Y+Z)+(G-H)*(G+H))/(2*H*Y)
     :G=SQR(F*F+1)
     :SS=SGN(F)
     :SF=SS*ABS(G)
     :F=((X-Z)*(X+Z)+H*(Y/(F+SF)-H))/X
     :C=1
     :S=1
1590 FORII=LTOK1
     :I=II+1
     :G=RV1(I)
     :Y=W(I)
     :H=S*G
     :G=C*G
     :Z=SQR(F*F+H*H)
     :RV1(II)=Z
     :C=F/Z
     :S=H/Z
     :F=X*C+G*S
     :G=-X*S+G*C
     :H=Y*S
     :Y=Y*C
     :IFMV$="TRUE"THEN1600ELSE1610
1600 FORJ=1TON
     :X=V(J,II)
     :Z=V(J,I)
     :V(J,II)=X*C+Z*S
     :V(J,I)=-X*S+Z*C
     :NEXTJ
1610 Z=SQR(F*F+H*H)
     :W(II)=Z
     :IFZ=0THEN1630
1620 C=F/Z
     :S=H/Z
1630 F=C*G+S*Y
     :X=-S*G+C*Y
     :IFMU$="TRUE"THEN1640ELSE1650
1640 FORJ=1TOM
     :Y=U(J,II)
     :Z=U(J,I)
     :U(J,II)=Y*C+Z*S
     :U(J,I)=-Y*S+Z*C
     :NEXTJ
1650 NEXTII
1660 RV1(L)=0
     :RV1(K)=F
     :W(K)=X
     :GOTO1480
1670 IFZ>=0THEN1700
1680 W(K)=-Z
     :IFMV$="TRUE"THEN1690ELSE1700
1690 FORJ=1TON
     :V(J,K)=-V(J,K)
```

```
    :NEXTJ
1700 NEXTKK
1710 GOTO1730
1720 IE=K
1730 RETURN
2000 '***************************************
2010 'LINEAR REGRESSION CONTROL MODULE
2020 '///////////////////////
2030 CLS
    :PRINT@128,"ENTER VECTOR Y(M) IN X*B=Y  WHERE R=M & C=1."
    :GOSUB130
    :TSS=0
    :MSS=0
    :FORI=1TOR
    :Y(I,1)=A(I,1)
    :TSS=TSS+Y(I,1)*Y(I,1)
    :MSS=MSS+Y(I,1)
    :NEXTI
2040 CLS
    :PRINT@128,"ENTER MATRIX X(M,N) OF X*B=Y."
    :GOSUB130
2042 IFF$="XM"THEN2045ELSEF$="X"
2043 DD$="1"
    :GOSUB222
2045 CLS
    :PRINT@458,"PLEASE BE PATIENT.  COMPUTATIONS IN PROGRESS."
    :GOSUB 1000
    :'SVD
2050 IF IE<>0THENCLS
    :PRINT@458,"ERROR RETURN FROM SVD"
    :STOP
2060 WI=10000
    :WM=0
    :FORJ=1TON
    :IFW(J)>WMTHENWM=W(J)
2065 IFW(J)<WITHENWI=W(J)
2070 B(J)=0
    :NEXTJ
    :CLS
    :PRINT@448,"NUMBER OF SIGNIFICANT FIGURES IN RAW DATA? --> ";
    :FL=-1
    :GOSUB15
    :S=VAL(IN$)
    :S=S+2
    :RE=0.1[S
    :TA=RE*WM
    :'TA SETS EFFECTIVE RANK AND IS ARBITRARILY COMPUTED
2080 RR=0
    :FORJ=1TON
    :IFW(J)<=TATHEN2100
2090 RR=RR+1
    :S=0
    :FORI=1TOM
    :S=S+U(I,J)*Y(I,1)
    :NEXTI
    :S=S/W(J)
    :FORI=1TON
    :B(I)=B(I)+S*V(I,J)
    :NEXTI
    :SV(J)=(1/W(J))[2
```

```
2100 NEXTJ
2102 IFRR<NTHENB(N+6)=-1
    :PRINT
    :PRINT"WARNING!!  MATRIX IS SINGULAR."
    :ELSEB(N+6)=WI
2104 B(N+4)=RE
    :B(N+5)=WM
    :B(N+7)=MSS/M
2110 ESS=0
    :FORI=1TOM
    :RI=0
    :FORJ=1TON
    :RI=RI+B(J)*A(I,J)
    :NEXTJ
    :ESS=ESS+(RI-Y(I,1))[2
    :NEXTI
2120 MSS=M*(MSS/M)[2
    :SE=SQR(ESS/(M-RR))
    :FORI=1TOM
    :Y(I,2)=0
    :FORJ=1TON
    :Y(I,2)=Y(I,2)+A(I,J)*B(J)
    :NEXTJ
    :Y(I,3)=Y(I,1)-Y(I,2)
    :Y(I,4)=Y(I,3)/SE
    :NEXTI
2130 RSS=TSS-ESS
    :RMSS=RSS-MSS
    :TMSS=TSS-MSS
    :IFA(1,1)=1ANDA(INT(N/2),1)=1ANDA(N,1)=1THENMEAN$="TRUE"ELSEMEAN$="FALSE"
2140 'DECISION BRANCHES FOR ANOVA TABLES
2150 IFMEAN$="TRUE"THEN2170
2160 RR=RR+1
    :FR=(RSS/RR)/(ESS/(M-RR))
    :DN=RR
    :DD=M-RR
    :GOSUB10000
    :A6=RR
    :A7=RSS
    :A8=RSS/RR
    :A9=FR
    :Z1=FB
    :Z2=M
    :Z3=TSS
    :GOTO2220
2170 CLS
    :PRINT@448,"INCLUDE TERM FOR MEAN IN ANOVA? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IFIN$="Y"THENFM$="TRUE"ELSEFM$="FALSE"
2180 IFFM$="FALSE"THEN2200
2190 A1=1
    :A2=MSS
    :A3=MSS
    :FR=MSS/(ESS/(M-RR))
    :A4=FR
    :DN=1
    :DD=M-RR
    :GOSUB10000
    :A5=FB
```

```
    :A6=RR-1
    :A7=RMSS
    :A8=RMSS/(RR-1)
    :FR=A8/(ESS/(M-RR))
    :A9=FR
    :DN=RR-1
    :DD=M-RR
    :GOSUB10000
    :Z1=FB
    :Z2=M
    :Z3=TSS
    :GOTO2220
2200 A6=RR-1
    :A7=RMSS
    :A8=RMSS/(RR-1)
    :A9=A8/(ESS/(M-RR))
    :FR=A9
    :DN=RR-1
    :DD=M-RR
    :GOSUB10000
    :Z1=FB
    :Z2=M-1
    :Z3=TMSS
2210 'ANOVA TABLE DISPLAY
2220 CLS
    :PRINT@10,"REGRESSION ANALYSIS OF VARIANCE TABLE"
2230 PRINT@128,"SOURCE";TAB(7)"DF";TAB(15)"SS";TAB(28)"MS";TAB(43)"F";TAB(50)"PRO
BABILITY"
2240 PRINT@192,"---------------------------------------------------------------"
2250 IFFM$="TRUE"THEN2260ELSE2270
2260 PRINT"MEAN";TAB(6)A1;TAB(12)A2;TAB(25)A3;TAB(40)A4;TAB(53);USING"#.####";A5
2270 PRINT"REG.";TAB(6)A6;TAB(12)A7;TAB(25)A8;TAB(40)A9;TAB(53);USING"#.####";Z1
2280 PRINT"ERROR";TAB(6)M-RR;TAB(12)ESS;TAB(25)ESS/(M-RR)
2290 PRINT"-------------------------------------------------------"
2300 PRINT"TOTAL";TAB(6)Z2;TAB(12)Z3
2305 PRINT"MEAN OF RESPONSE VARIABLE Y = "B(N+7)
2310 PRINT"STD.DEV. OF ERROR= "SQR(ESS/(M-RR));"  C.V. OF Y = ";USING"###.#";((SQ
R(ESS/(M-RR)))/B(N+7))*100
    :IFMEAN$="TRUE"THENR2=RMSS/TMSSELSER2=RSS/TSS
2312 PRINT"MULTIPLE CORRELATION COEFFICIENT (R**2) = ";R2
    :PRINT"REGRESSION COEFFICIENTS"
    :FORI=0TON-1
    :IFMEAN$="FALSE"THEN2330
2320 PRINT"B"I"= "B(I+1),
    :GOTO2340
2330 PRINT"B"I+1"= "B(I+1),
2340 NEXTI
2345 A$=INKEY$
    :IFA$=""THEN2345ELSEIFASC(A$)=64THENGOSUB670
2350 IFMEAN$="FALSE"THEN2380
2360 IFFM$="TRUE"THENB$="WITHOUT"ELSEB$="WITH"
2370 CLS
    :PRINT@448,"WANT TO VIEW THE ANOVA "B$" THE MEAN? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IFIN$="N"THEN2380ELSEIFIN$="Y"ANDB$="WITH"THENFM$="TRUE"
    :GOTO2190
    :ELSEFM$="FALSE"
    :GOTO2200
2380 'SAVE Y,(X'X)-1,AND B WITH B(0)=RR
```

```
2382 CLS
    :PRINT@448,"COMPUTE CONFIDENCE INTERVALS OR DO ERROR ANALSIS? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IFIN$="N"THENRETURN
2385 CLS
    :PRINT@448,"MATRICES Y(.,4), (X'X)-1, AND B(.) WILL NOW BE SAVED."
    :PRINT"BE SURE A DATA DISK IS IN DRIVE #1."
    :PRINT
    :PRINT"<ENTER> TO CONTINUE."
2386 A$=INKEY$
    :IFA$=""THEN2386
2390 FORI=1TOM
    :FORJ=1TO4
    :A(I,J)=Y(I,J)
    :NEXTJ,I
    :FF$="YN:1"
    :L$="YL:1"
    :F$="YM:1"
    :N$(1)="Y"
    :N$(2)="YHAT"
    :N$(3)="EHAT"
    :N$(4)="ZHAT"
    :R=M
    :C=4
    :GOSUB225
2395 IFMEAN$="TRUE"THENB(N+2)=1ELSEB(N+2)=2
2400 B(N+8)=N
    :B(N+1)=ESS/(M-RR)
    :B(N+3)=RR
    :FORI=1TON+8
    :A(I,1)=B(I)
    :NEXTI
    :N$(1)="1"
    :FF$="BN:1"
    :L$="BL:1"
    :F$="BM:1"
    :R=N+8
    :C=1
    :GOSUB225
2405 FORI=1TON
    :N$(I)=STR$(I)
    :NEXTI
    :'SET COLUMN LABELS CORRECTLY
2410 FORI=1TON
    :FORJ=1TON
    :U(I,J)=SV(I)*V(J,I)
    :NEXTJ,I
    :FORI=1TON
    :FORJ=1TON
    :A(I,J)=0
    :FORK=1TON
    :A(I,J)=A(I,J)+V(I,K)*U(K,J)
    :NEXTK,J,I
    :FF$="XPXIN:1"
    :L$="XPXIL:1"
    :F$="XPXIM:1"
    :R=N
    :C=N
    :GOSUB225
```

```
2420 RUN"GLR1/BAS"
3000 '****************************************
3010 'BARTLETT'S TEST FOR HOMOGENEITY OF VARIANCES
3020 '     AFTER
   : SOKAL AND ROLFE 1969, PP. 370-371
3030 'USER MUST INPUT VARIANCES AND THEIR DF'S
3040 '////////////////////////////////////////
3050 CLS
    :PRINT@448,"TO USE THIS ROUTINE YOU MUST HAVE THE VARIANCES AND THEIR DEGREES
 OF FREEDOM.  THEY ARE NOT COMPUTED FOR YOU IN THIS ROUTINE."
    :PRINT
    :PRINT"ENTER <Y> IF YOU HAVE THEM AND <N> IF YOU DO NOT."
    :PRINT
    :PRINT"ENTER --> ";
3060 FL=1
    :GOSUB15
3070:IFIN$="Y"THEN3070ELSERETURN
    :PRINT@448,"ENTER THE NUMBER OF VARIANCES TO BE COMPARED --> ";
    :FL=-2
    :GOSUB15
    :N1=VAL(IN$)
    :R=N1
    :C=2
    :N$(1)="VARIANCE"
    :N$(2)="D.F"
3075 CLS
    :PRINT"ENTER VARIANCE S2(I) AND D.F.= N(I)-1."
    :PRINT
3080 FORI=1TON1
    :PRINT"VARIANCE #"I"=";
    :FL=-8
    :GOSUB15
    :A(I,1)=VAL(IN$)
    :PRINT"            DF #"I"=";
    :FL=-3
    :GOSUB15
    :A(I,2)=VAL(IN$)
    :PRINT
    :NEXTI
3090 CLS
    :PRINT@448,"DO YOU WISH TO REVIEW THE ENTRIES ABOVE? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IF IN$="N"THEN3100ELSEGOSUB510
3100 CLS
    :PRINT@448,"ENTER <C> TO CONTINUE, <R> TO RESTART DATA ENTRY. --> ";
    :FL=1
    :GOSUB15
    :IFIN$="C"THEN3110ELSE3080
3110 CLS
    :PRINT@448,"TO AVOID NEGATIVE LOGARITHMS, ALL VARIANCES MUST BE GREATER"
    :PRINT"THAN ONE (1.0).  ARE THEY? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IFIN$="Y"THEN3130
3120 PRINT
    :PRINT"ENTER THE CONSTANT MULTIPLIER TO INSURE NO NEG. LOGS --> ";
    :FL=4
    :GOSUB15
```

```
    :C1=VAL(IN$)
    :FORI=1TON1
    :A(I,1)=C1*A(I,1)
    :NEXTI
    :GOSUB510
3130 CLS
    :PRINT@458,"PLEASE BE PATIENT.  COMPUTATIONS IN PROGRESS."
3140 S5=0
    :S4=0
    :S1=0
    :S2=0
    :FORI=1TON1
    :S1=S1+A(I,2)
    :S2=S2+A(I,2)*A(I,1)
    :S4=S4+A(I,2)*(LOG(A(I,1))/LOG(10))
    :S5=S5+1/A(I,2)
    :NEXTI
    :S3=S2/S1
    :S3=LOG(S3)/LOG(10)
3150 CHI2=2.3026*((S1*S3)-S4)
3170 CF=1+(1/(3*(N1-1)))*(S5-(1/S1))
    :CHI2=CHI2/CF
3180 CLS
    :PRINT@448,"CHI SQUARED = "CHI2" WITH "N1-1" DF."
    :PRINT
    :PRINT"REJECT HOMOGENEITY OF VARIANCES IF CALCULATED VALUE (ABOVE) IS  GREATE
R THAN TABLE VALUE. (LOOK IT UP)"
    :PRINT
    :PRINT"<ENTER> TO RETURN TO MAIN MENUE."
3190 A$=INKEY$
    :IFA$=""THEN3190ELSERETURN
10000 '*************************************
10010 'SUBROUTINE  FPROB
10040 'INPUT
    : FR=R RATIO  DN=NUMERATOR DF  DD=DENOMINATOR DF
10050 'OUTPUT
    : FB= PROBABILITY OF F
10060 '///////////////////////
10065 X#=1
    :IFFR<1THEN10080
10070 S#=DN
    :T#=DD
    :ZZ#=FR
    :GOTO10090
10080 S#=DD
    :T#=DN
    :ZZ#=1/FR
10090 J5=2/9/S#
    :K5=2/9/T#
    :Z#=K5*ZZ#[(2/3)+J5
    :GOSUB10500
    :Y#=ABS((1-K5)*ZZ#[(1/3)-1+J5)/Z3#
    :IFT#<4THEN10110
10100 X#=.5/(1+Y#*(.196854+Y#*(.115194+Y#*(.000344+Y#*.019527))))[4
    :X#=INT(X#*10000+.5)/10000
    :GOTO10120
10110 Y#=Y#*(1+.08*Y#[4/T#[3)
    :GOTO10100
10120 IFF#>1THEN10140
10130 X#=1-X#
```

```
10140 FB=1-X#
10150 RETURN
10500 Z3#=SQR(Z#)
    :Z3#=(Z3#+Z#/Z3#)/2
    :Z3#=(Z3#+Z#/Z3#)/2
    :RETURN
40000 M=100
    :N=10
    :DIM N$(N),A(M,N),W(N),U(M,N),V(N,N),RV1(N),Y(M,4),B(N+8),SV(N),XI(N,N)
40015 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
    :ZZ=INT(30-CC/2)
    :PRINT TAB(ZZ);T$
    :RETURN
    :'CENTERING ROUTINE
43000 END
```

APPENDIX II. Source code listing for program GLR1/BAS

```
10 CLEAR 500
   :TX=-1
   :GOTO700
   :'GLR1/BAS  VERSION 02/19/81
13 '*************************************************
14 'ANSWER INPUT ROUTINE
15 P=(256*PEEK(16417)+PEEK(16416))-15359
   :IF ABS(FL)=1THEN16ELSE24
16 IN$=""
   :PRINT@P,CHR$(136);
   :LX=1
17 IN$=INKEY$
   :IFIN$=""THEN22
18 IFFL=-1 AND ASC(IN$)>47 AND ASC(IN$)<58 THEN 21
19 IF FL=1 AND ASC(IN$)>64 AND ASC(IN$)<91 THEN 21
20 GOTO 22
21 PRINT@P,IN$
   :RETURN
22 LX=LX+1
   :IFL<16THEN17
23 PRINT@P," ";
   :FORLX=1TO25
   :NEXT
   :GOTO16
24 PRINT@P, CHR$(136);
   :IN$=""
   :LX=ABS(FL)
   :GOSUB34
25 A$=INKEY$
   :IFA$=""THEN25
26 IFA$=CHR$(13)THENPRINTCHR$(15);
   :RETURN
27 IF A$=CHR$(8)THENIFIN$<>""THENPRINTA$;
   :PRINTCHR$(136);CHR$(24);
   :IN$=LEFT$(IN$,LEN(IN$)-1)
   :LX=LX+1
   :GOTO25
   :ELSEGOTO28
28 IFA$=CHR$(24)THENPRINTCHR$(29);
   :GOTO24
29 IF FL>0 AND ASC(A$)>31 AND ASC(A$)<91 THEN 32
30 IF FL<0 AND ASC(A$)>=45 AND ASC(A$)<58 THEN 32
31 GOTO 25
32 IF LX=0 THEN 25
33 PRINT A$;
   :IN$=IN$+A$
   :LX=LX-1
   :GOTO25
34 FOR IX=1TOLX-1
   :PRINT CHR$(136);
   :NEXT
   :FORJX=1TOLX
   :PRINT CHR$(24);
   :NEXT
   :RETURN
50 '************************************
52 'CHANGE VARIABLE NAMES
54 '/////////////
56 CLS
   :PRINT@448,"WHICH COLUMN'S NAME WILL BE CHANGED? --> ";
```

```
    :FL=-2
    :GOSUB15
    :XJ=VAL(IN$)
    :PRINT
    :PRINT"PLEASE ENTER NEW NAME --> ";
    :FL=8
    :GOSUB15
    :N$(XJ)=IN$
    :RETURN
100 '*****************************************
110 'READ SEQUENTIAL DATA SET FROM DISK
120 '///////////////
125 CLS
130 PRINT@448,"ENTER FILENAME OF DESIRED DATA SET --> ";
    :INPUT F$
135 FF$=F$+"N"
    :L$=F$+"L"
    :F$=F$+"M"
140 OPEN "I",1,FF$
    :INPUT#1,R,C
    :CLOSE1
    :'GET R,C VALUES
145 OPEN "I",1,L$
    :FORK=1TOC
    :INPUT#1,N$(K)
    :NEXTK
146 CLOSE
150 OPEN "I",1,F$
    :
160 FORI=1TOR
    :FORJ=1TOC
    :INPUT#1,A(I,J)
    :NEXTJ,I
170 CLOSE 1
180 RETURN
190 '*********************************************
200 ' WRITE A SEQUENTIAL MATRIX TO DISK
210 '/////////////////
220 CLS
    :PRINT@448,"ENTER FILENAME OF NEW DATA SET --> ";
    :INPUT F$
    :PRINT
    :PRINT"ENTER DISK DRIVE # WHERE DATA SHOULD BE SAVED --> ";
    :INPUT DD$
222 FF$=F$+"N:"+DD$
    :L$=F$+"L:"+DD$
    :F$=F$+"M:"+DD$
225 FOR JQ=1TOC
    :IF LEN(N$(JQ))=0 THEN N$(JQ)=STR$(JQ)
226 NEXT
230 OPEN "O",1,FF$
    :PRINT#1,R;C;
    :CLOSE1
240 OPEN "O",1,F$
    :
250 FORI=1TOR
    :FORJ=1TOC
    :PRINT#1,A(I,J);
    :NEXTJ,I
260 CLOSE 1
```

```
265 OPEN"O",1,L$
    :FORI=1TOC
    :PRINT#1,N$(I);",";
    :NEXT
    :CLOSE
270 RETURN
290 REM ************************************************
295 REM                    SUBROUTINE INPUT
300 '1.  THIS SUB READS FROM THE KEYBOARD
305 '2.  MATRIX A IS LOADED AND CAN BE SAVED ON DISK
325 '6.  14-APR-79
335 REM ************************************************
340 CLS
345 INPUT "PLEASE ENTER: NO. ROWS AND COLS (R,C).--> ";R,C
350 FL=8
    :PRINT
    :FOR J=1TOC
    :PRINT"ENTER THE NAME FOR VARIABLE ";J;" --> ";
    :GOSUB15
    :N$(J)=IN$
    :PRINT
    :NEXT
355 CLS
365 FOR I=1 TO R
370 PRINT "ROW ";I
375 FOR J=1 TO C
380 FL=-12
    :PRINT "PLEASE ENTER:  VALUE OF COLUMN ";J" --> ";
    :GOSUB 15
385 PRINT
390 A(I,J)=VAL(IN$)
395 NEXT J
400 NEXT I
430 RETURN
435 END
440 '************************************************
445 '           SUBROUTINE VIDEO
450 '  DISPLAYS MATRIX A(R,C),13 ROWS BY 4 COLUMNS AT ONCE
455 ' USES ARROWS IN ALL DIRECTIONS TO SEE DIFFERENT PAGES
460 ' OF DATA.  "CLEAR" WILL TERMINATE DISPLAY
500 '************************************************
510 CLS
    :IFR=0 OR C=0 THEN RETURN ELSE JJ=INT(C/4.1+1)
    :II=INT(R/13.1+1)
515 I=1
    :J=1
525       IF R-13*I>=0.0 THEN N=13 ELSE N=13+(R-13*I)
535       IF C-4*J>=0.0 THEN L=4 ELSE L=4+(C-4*J)
540 CLS
    :PRINT"COL:";
    :FOR KK=0TO(L-1)
    :PRINT TAB(14*KK+8) N$((KK+1)+4*(J-1));
545 NEXT KK
    :PRINT
550 PRINT"------------------------------------------------------------"
555 FOR M=1 TO N
560 PRINT "R";M+13*(I-1);":";
565 FOR K=1 TO L
570 A=A(M+13*(I-1),K+4*(J-1))
575 PRINT TAB(14*(K-1)+8) A;
```

```
580 NEXT K
    :PRINT
585 NEXT M
586 A$=INKEY$
    :IFA$=""GOTO586
587 IF ASC(A$)=31THEN619ELSEIFASC(A$)=9THEN588ELSE590
588 IFJ+1>JJTHENJ=1ELSEJ=J+1
589 GOTO 525
590 IF ASC(A$)=8THEN591ELSE593
591 IF J-1<=0THENJ=JJ ELSE J=J-1
592 GOTO 525
593 IF ASC(A$)=10THEN594ELSE596
594 IF I+1>IITHENI=1ELSEI=I+1
595 GOTO 525
596 IF ASC(A$)=91THEN597ELSE599
597 IF I-1<=0THENI=IIELSEI=I-1
598 GOTO 525
599 IF ASC(A$)=64 THEN 600 ELSE 601
600 GOSUB 670
    :GOTO525
601 IF ASC(A$)=46 THEN GOSUB 50 ELSE 618
602 GOTO 525
618 GOTO586
619 RETURN
    :END
620 '**********************************************
622 'LINEPRINTER OUTPUT ROUTINE
623 IF R=0 OR C=0 THEN RETURN
624 AN$="DATA SET NAME --> "
    :CLS
    :PRINT@448,"PRESS <ENTER> WHEN PRINTER IS READY.";
    :INPUT AN
    :T$=LEFT$(F$,LEN(F$)-1)
    :T$=AN$+T$
    :GOSUB42000
    :LPRINT TAB(ZZ);T$
    :LPRINT " "
    :LPRINT " "
630 JJ=INT(C/4.1+1)
    :II=INT(R/200.1+1)
632 FORI=1TOII
634 IFR-200*I>=0.0THENN=200ELSEN=200+(R-200*I)
636 FORJ=1TOJJ
638 IFC-4*J>=0.0THENL=4ELSE L=4+(C-4*J)
640 LPRINT"COL:";
    :FORKK=0TO(L-1)
    :LPRINT TAB(14*KK+8) N$((KK+1)+4*(J-1));
642 NEXT KK
644 LPRINT" "
646 LPRINT"------------------------------------------------------------"
648 FOR M=1TON
650 LPRINT "R";M+200*(I-1);":";
652 FOR K=1TO L
654 A=A(M+200*(I-1),K+4*(J-1))
656 LPRINT TAB(14*(K-1)+8) A;
658 NEXT K
660 LPRINT
662 NEXT M
664 LPRINT " "
    :LPRINT " "
```

```
    :LPRINT " "
666 NEXT J,I
668 RETURN
670 '********************************
671 'SCREEN PRINT ROUTINE-  ACTIVATED BY PROGRAMMED "@" ASC(64)
672 '/////////////////
673 IF PEEK(14312)<>63 THEN RETURN
674 LPRINT STRING$(64,"+")
675 FOR N= 15360 TO 16383 STEP 64
676 A$=" "
679 FOR Q=0TO63
680 Z=PEEK(N+Q)
    :IFZ>0ANDZ<27THENZ=Z+64ELSEIFZ<32THENZ=32
681 IFZ>=127 AND Z<=191 THEN Z=42
682 B$=CHR$(Z)
683 A$=A$+B$
684 NEXT Q
685 IF PEEK(14312)<>63THEN RETURN
686 LPRINT " "
    :LPRINT A$
687 NEXT N
688 LPRINT STRING$(64,"+")
689 FORN=1TO3
    :LPRINT "  "
    :NEXT
690 RETURN
695 RUN"GLR/BAS"
    :END
700 '**********************************************
710 'GENERALIZED I/O SUBROUTINES FOR PROGRAM STARTERS
720 '/////////////////
735 CLS
    :T$="NORTHERN HARDWOODS LABORATORY, USFS, MARQUETTE, MI"
    :GOSUB42000
736 T$="////////////////////////////////////////////////////////"
    :GOSUB42000
737 PRINT
738 T$="MENU FOR:  GENERAL LINEAR REGRESSION ANALYSIS SYSTEM
739 GOSUB42000
742 T$="PART II.  CONFIDENCE INTERVALS AND ERROR ANALYSIS"
    :GOSUB42000
    :PRINT
    :IFTX=-1THENGOSUB40000ELSE750
750 TX=1
    :PRINT"  1.  RETURN TO MAIN MENU          2.  GOTO BASIC
760 PRINT"  3.  READ A MATRIX FROM DISK      4.  WRITE A MATRIX TO DISK
770 PRINT"  5.  ENTER A MATRIX MANUALLY      6.  DISPLAY MATRIX ON VIDEO
780 PRINT"  7.  PRINT MATRIX                 8.  CONFDNCE INTERVALS FOR B
790 PRINT"  9.  CONFDNCE INTVLS FOR YHAT     10.  PLOT RESIDUALS
800 PRINT" 11.  MULTICOLLINEARITY CHECK      12.  HETEROSCEDASTICITY CHECK"
810 PRINT" 13.  AUTOCORRELATION CHECK        14.  SCALED B
820 PRINT" 15.  EQUALITY OF REG. LINES"
900 PRINT
    :PRINT"SELECT ONE OF THE ABOVE --> ";
    :FL=-2
    :GOSUB15
    :AN=VAL(IN$)
    :IFAN<=0ORAN>16THEN740
905 IFAN<8THENFM$="FALSE"
    :'PREVENTS DATA MIXUPS
```

```
910 ON AN GOSUB 695,43000,125,220,340,510,620,1000,2000,3000,4000,5000,6000,7000,
8000
920 GOTO 735
1000 '*******************************************
1010 'CONFIDENCE INTERVALS FOR COEFFICIENTS B
1020 '//////////////////////
1030 IFFM$="FALSE"THENGOSUB 10000
    :'READ B,Y, AND (X'X)-1
1040 GOSUB11000
    :'GET TC2
1050 SIG=SQR(B(N+1))
    :FORI=1TON
    :CI(I)=SIG*TC2*SQR(XI(I,I))
    :NEXTI
    :IFA(1,1)=1ANDA(M,1)=1THENLL=0ELSELL=1
1060 CLS
    :PRINT@10,"CONFIDENCE INTERVALS FOR THE COEFFICIENTS."
    :PRINT"T-VALUE FOR 95% CONFIDENCE INTERVAL= ";TC2;" DF= "X
    :PRINT"COEFFICIENT","UPPER LEVEL","LOWER LEVEL","STAND. DEV."
1062 PRINT"---------------------------------------------------------------"
1065 K=1
    :FORI=1TON
    :IFLL=0THENK=I-1
1070 PRINT"B"K" ";B(I),B(I)+CI(I),B(I)-CI(I),SQR(XI(I,I)*B(N+1))
    :K=K+1
    :NEXTI
1080 GOSUB8130
1090 RETURN
2000 '*********************************************
2010 'CONFIDENCE INTERVAL AROUND PREDICTED Y VALUES
2020 '//////////////////////
2030 CN=0
    :IFFM$="FALSE"THENGOSUB10000
    :'CHECK IF Y AND (X'X)-1 ARE LOADED
2040 GOSUB11000
    :'FILL TC2
2045 SIG=SQR(B(N+1))
    :GOTO2055
2050 CLS
    :PRINT@10,"CONFIDENCE INTERVALS FOR PREDICTED Y VALUES."
    :PRINT"T-VALUE FOR 95% TWO-SIDED CONFIDENCE INTERVAL= ";TC2;" DF= "X
    :PRINT"YHAT","UPPER CI","LOWER CI","STAND. ERR."
2052 PRINT"---------------------------------------------------------------"
    :RETURN
2055 GOSUB2050
2060 FORI=1TOM
    :SY=0
    :VY=0
    :FORJ=1TON
    :CI(J)=0
    :FORK=1TON
    :CI(J)=CI(J)+A(I,K)*XI(K,J)
    :NEXTK,J
2070 FORJ=1TON
    :VY=VY+CI(J)*A(I,J)
    :NEXTJ
    :SY=SIG*SQR(VY)
    :CI=TC2*SY
2080 PRINTY(I,2),Y(I,2)+CI,Y(I,2)-CI,SY
2085 Y(I,5)=Y(I,2)+CI
```

```
    :Y(I,6)=Y(I,2)-CI
2090 IFCN<9THENCN=CN+1ELSE2110
2100 GOTO2120
2110 GOSUB8130
2115 CN=0
    :GOSUB2050
2120 NEXTI
2125 A$=INKEY$
    :IFA$=""THEN2125
2127 CLS
    :PRINT@448,"DO YOU WISH TO SAVE THE CONFIDENCE INTERVALS FOR YHAT IN THE Y  M
ATRIX? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IFIN$="N"THENRETURNELSE2130
2130 DD$="1"
    :F$="Y"
    :C=6
    :R=M
    :FORI=1TOM
    :FORJ=1TOC
    :A(I,J)=Y(I,J)
    :NEXTJ,I
    :FORI=1TO4
    :N$(I)=YN$(I)
    :NEXTI
    :N$(5)="Y-UPPER"
    :N$(6)="Y-LOWER"
    :GOSUB222
    :RETURN
3000 '************************************
3010 'PREPARE THE Y MATRIX FOR PLOTTING OF RESIDUALS
3020 '//////////////////////
3022 CLS
    :T$="PREPARING FOR MATRIX PLOTTING OF RESIDUALS"
    :GOSUB42000
3030 F$="Y"
    :GOSUB135
    :FORI=1TOR
    :FORJ=1TOC
    :Y(I,J)=A(I,J)
    :NEXTJ,I
    :FORI=1TO6
    :YN$(I)=N$(I)
    :NEXTI
    :F$="X"
    :GOSUB135
3040 N=C
    :C=6
    :FORJ=1TON
    :IFA(1,J)=1ANDA(R,J)=1THEN3060
3050 C=C+1
    :FORI=1TOR
    :Y(I,C)=A(I,J)
    :NEXTI
    :N$(C)="X"+STR$(C-6)
3060 NEXTJ
3062 C=C+1
    :FORI=1TOR
    :Y(I,C)=I
```

```
    :NEXTI
    :N$(C)="OBS #"
3065 FORI=1TO6
    :N$(I)=YN$(I)
    :NEXTI
    :'NOTE X'S LABELED ABOVE
3070 DD$="1"
    :F$="Y"
    :FORI=1TOM
    :FORJ=1TOC
    :A(I,J)=Y(I,J)
    :NEXTJ,I
    :GOSUB222
3080 RUN"MULTPLT/BAS"
4000 '******************************************
4010 'MULTICOLLINEARITY CHECK
4020 '////////////////////
4021 CLS
    :T$="CHECK FOR MULTICOLLINEARITY"
    :GOSUB42000
    :PRINT
4022 IFFM$="TRUE"THEN4035
4025 F$="B"
    :GOSUB135
    :FORI=1TOR
    :B(I)=A(I,1)
    :NEXTI
    :N=B(R)
4035 IFB(N+3)<NTHENZI$="REDUCED RANK"ELSEZI$="FULL RANK"
4040 PRINT
    :PRINT"RANK(X)= "B(N+3);
    :PRINT"  X HAS "ZI$
4050 PRINT"MAXIMUM SINGULAR VALUE (X) = "B(N+5)
    :IFB(N+6)=-1THENZI=0ELSEZI=B(N+6)
4060 PRINT"MINIMUM SINGULAR VALUE (X) = "ZI
4070 IFB(N+6)=-1THENZI$="INFINITE"ELSEZI$=STR$(B(N+5)/B(N+6))
4080 PRINT"CONDITION NUMBER FOR (X) = "ZI$
4090 PRINT"COLLINEARITY NUMBER FOR (X) = "1/B(N+4)
    :PRINT
    :PRINT
4100 PRINT"IF CONDITION NUMBER IS CLOSE TO 1, THEN X = INDEPENDENT (NONSINGULAR).
"
4110 PRINT"IF CONDITION NUMBER IS INFINITE THEN X = DEPENDENT (SINGULAR)."
4120 PRINT"IF CONDITION NUMBER CLOSE TO COLLINEARITY NUMBER, X = NEARLY DEPENDENT
 (SINGULAR)."
4130 GOSUB8130
4140 RETURN
5000 '******************************************
5010 'HETEROSCEDASTICITY CHECK
5020 '/////////////////////
5030 CLS
    :PRINT@448,"NO NUMERICAL EVALUATION IMPLEMENTED AT PRESENT."
    :PRINT"VISUAL EXAMINATION OF RESIDUALS PLOTS IS RECOMMENDED."
5035 A$=INKEY$
    :IFA$=""THEN5035
5040 RETURN
6000 '******************************************
6010 'AUTOCORRELATION CHECK
6020 '////////////////////
6025 CLS
```

```
    :T$="AUTOCORRELATION CHECK"
    :GOSUB42000
    :PRINT
6030 IFFM$="TRUE"THEN6045
6045 F$="Y"
    :GOSUB135
6050 S1=0
    :S2=0
    :FORI=2TOR
    :S1=S1+(A(I,4)-A(I-1,4))[2
    :S2=S2+A(I,4)*A(I,4)
    :NEXTI
    :S2=S2+A(1,4)*A(1,4)
6060 PRINT"THE DURBIN-WATSON STATISTIC:  D= "S1/S2
6070 PRINT
    :PRINT"SEE USER'S GUIDE FOR EVALUATION."
6080 GOSUB8130
6090 RETURN
7000 '******************************************
7010 'PRODUCE THE SCALED B VALUES
7020 '////////////////////
7030 CLS
    :T$="SCALED B COEFFICIENTS"
    :GOSUB42000
    :PRINT
7033 IF FM$="FALSE"THENGOSUB10000
    :GOTO7040
    :ELSE7035
7035 F$="X"
    :GOSUB135
    :'NEED X MATRIX IN A( ) FOR TEST IN 7040
7040 IFA(1,1)=1ANDA(M,1)=1THENLL=2ELSELL=1
7050 FORI=LLTON
    :C(I)=0
    :XZ(I)=0
    :NEXTI
7060 FORI=1TOM
    :FORJ=LLTON
    :CI(J)=CI(J)+A(I,J)
    :NEXTJ,I
    :'COL SUMS
7070 FORJ=LLTON
    :CI(J)=CI(J)/M
    :NEXTJ
    :'COL AVGS
7080 SYY=0
    :FORI=1TOM
    :SYY=SYY+(Y(I,1)-B(N+7))[2
    :FORJ=LLTON
    :XZ(J)=XZ(J)+(A(I,J)-CI(J))[2
    :NEXTJ,I
7090 K=1
    :FORI=LLTON
    :IFLL=2THENK=I-1
7100 PRINT"B"K" SCALED = "B(I)*SQR(XZ(I)/SYY)
7110 K=K+1
    :NEXTI
7120 GOSUB8130
7130 RETURN
8000 '**************************************************
```

```
8010 'TEST FOR EQUALITY OF REGRESSION LINES
8020 '//////////////////////
8022 CLS
    :PRINT"YOU MUST BE PREPARED TO ENTER THE FOLLOWING INFORMATION"
    :PRINT
    :PRINT"1.  THE B(I) VALUE FOR EVERY EQUATION"
    :PRINT"2.  ITS STANDARD DEVIATION"
    :PRINT"3.  THE ERROR SUM OF SQUARES FOR EACH EQUATION."
8024 PRINT"4.  THE ERROR DEGREES OF FREEDOM FOR EACH EQUATION ."
    :PRINT
    :PRINT"IF YOU HAVE THE ABOVE DATA PIECES FOR EACH EQUATION TO BE"
    :PRINT"COMPARED THEN ENTER <Y>.  IF NOT ENTER <N> AND YOU WILL BE"
8026 PRINT"RETURNED TO THE MENUE PART II. TO COMPUTE THE ABOVE INFORMATION."
    :PRINT
    :PRINT"ENTER --> ";
    :FL=1
    :GOSUB15
    :IFIN$="Y"THEN8028ELSERETURN
8028 CLS
    :PRINT@448,"HOW MANY EQUATIONS ARE TO BE COMPARED? --> ";
    :FL=-2
    :GOSUB15
    :K=VAL(IN$)
    :PRINT
    :R=K
    :C=4
    :N$(1)="B"
    :N$(2)="STAND.DEV."
    :N$(3)="ERROR SS"
    :N$(4)="ERROR DF"
8040 CLS
    :PRINT"PLEASE ENTER THE REQUESTED INFORMATION FOR EACH EQUATION."
    :PRINT
    :FORI=1TOK
    :PRINT"EQUATION #"I"
8050 PRINT"VALUE OF B COEFFICIENT --> ";
    :FL=-10
    :GOSUB15
    :A(I,1)=VAL(IN$)
    :PRINT
    :PRINT"STANDARD DEVIATION OF B --> ";
    :GOSUB15
    :A(I,2)=VAL(IN$)
    :PRINT
    :PRINT"ERROR SUM OF SQARES --> ";
    :GOSUB15
    :A(I,3)=VAL(IN$)
    :PRINT
8060 PRINT"ERROR DEGREES OF FREEDOM --> ";
    :GOSUB15
    :A(I,4)=VAL(IN$)
    :PRINT
    :CLS
    :NEXTI
8062 CLS
    :PRINT@448,"WISH TO REVIEW THE INPUT? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IFIN$="Y"THENGOSUB510
8063 FORI=1TOR
```

```
    :A(I,2)=A(I,2)*A(I,2)
    :NEXTI
8065 CLS
    :PRINT@448,"CONTINE <C> OR REDO INPUT <R> --> ";
    :GOSUB15
    :IFIN$="C"THEN8067ELSE8040
8067 CLS
    :PRINT"ARE THE VARIANCES ASSUMED HOMOGENEOUS <O> OR"
    :PRINT"HETEROGENEOUS <E> ? --> ";
    :FL=1
    :GOSUB15
    :IFIN$="O"THEN8068ELSEIFIN$="E"THEN8200ELSE8067
8068 CLS
    :PRINT@458,"PLEASE BE PATIENT.  COMPUTATIONS IN PROGRESS."
8070 K=R
    :S1=0
    :S2=0
    :FORI=1TOK
    :S1=S1+A(I,3)
    :S2=S2+A(I,4)
    :A(I,2)=(A(I,3)/A(I,4))/A(I,2)
    :NEXTI
    :M1=S1/S2
    :ES=S1
    :DD=S2
8080 S1=0
    :S2=0
    :FORI=1TOK
    :S1=S1+A(I,1)*A(I,2)
    :S2=S2+A(I,2)
    :NEXTI
    :BB=S1/S2
8090 S1=0
    :FORI=1TOK
    :S1=S1+((A(I,1)-BB)*(A(I,1)-BB))/(1/A(I,2))
    :NEXTI
    :SB=S1
    :DN=K-1
    :M2=SB/DN
    :FR=M2/M1
    :GOSUB20000
8100 CLS
    :PRINT"RESULTS OF THE TEST FOR EQUALITY OF REGRESSION COEFFICIENTS."
    :PRINT
    :PRINT"WEIGHTED AVERAGED B = "BB
    :PRINT"SUM OF SQUARES FOR COMBINED B'S = "SB
8110 PRINT"NUMERATOR DEGREES OF FREEDOM = "DN
    :PRINT"MEAN SQUARE FOR B'S = "M2
    :PRINT"COMBINED ERROR SUM OF SQUARES = "ES
    :PRINT"DENOMINATOR DEGREES OF FREEDOM = "DD
    :PRINT"COMBINED ERROR MEAN SQUARE = "M1
8120 PRINT"THE CALCULATED F-VALUE = "FR
    :PRINT"THE PROBABILITY OF REJECTING THE NULL HYPOTHESIS = "FB
8130 A$=INKEY$
    :IFA$=""THEN8130ELSEIFASC(A$)=64THENGOSUB670
8140 RETURN
8200 'HETEROGENEOUS VARIANCES
8210 CLS
    :PRINT@448,"PLEASE BE PATIENT.  COMPUTATIONS IN PROGRESS."
8220 ES=0
```

```
:DD=0
:K=R
:FORI=1TOK
:ES=ES+A(I,3)
:DD=DD+((A(I,3)*A(I,3))/A(I,4))
:NEXTI
:DD=INT((ES*ES)/DD)
:M1=ES/DD
8230 S1=0
:S2=0
:FORI=1TOK
:S1=S1+A(I,1)/A(I,2)
:S2=S2+1/A(I,2)
:NEXTI
:BB=S1/S2
:SB=0
:FORI=1TOK
:SB=SB+((A(I,1)-BB)*(A(I,1)-BB))/A(I,2)
:NEXTI
:SB=SB*M1
:DN=K-1
:M2=SB/DN
:FR=M2/M1
:GOSUB20000
:GOTO8100
10000 CLS
:PRINT@448,"MATRICES B, Y, X AND (X'X)-1 WILL NOW BE READ FROM DISK."
:F$="B"
:GOSUB135
:FORI=1TOR
:B(I)=A(I,1)
:NEXTI
:N=B(R)
:F$="Y"
:GOSUB135
:FORI=1TOR
:FORJ=1TOC
:Y(I,J)=A(I,J)
:NEXTJ,I
:M=R
10010 FORI=1TO6
:YN$(I)=N$(I)
:NEXTI
:F$="XPXI"
:GOSUB135
:FORI=1TON
:FORJ=1TON
:XI(I,J)=A(I,J)
:NEXTJ,I
:F$="X"
:GOSUB135
:FM$="TRUE"
:RETURN
11000 '*********************************
11010 'INTERPOLATION FOR T C2 VALUES
11020 '//////////////////////
11025 IFXT(1)<>0THEN11050
11030 DATA 1,12.706,2,4.303,3,3.182,4,2.776,5,2.571,6,2.447,7,2.365,8,2.306,9,2.2
62,10,2.228,15,2.131,20,2.086,25,2.060,30,2.042,40,2.021,60,2.00,120,1.98
11040 FORI=1TO17
```

```
    :READXT(I),YT(I)
    :NEXTI
11050 X=M-B(N+3)
    :IFX<XT(1)THENTC2=YT(1)ELSEIFX>XT(17)THENTC2=YT(17)ELSE11070
11060 RETURN
11070 FORI=2TO17
    :IFX<=XT(I)THEN11080ELSENEXTI
11080 TC2=(YT(I)-YT(I-1))/(XT(I)-XT(I-1))*(X-XT(I-1))+YT(I-1)
11090 RETURN
20000 '***********************************
20010 'SUBROUTINE  FPROB
20020 'INPUT
    : FR=R RATIO  DN=NUMERATOR DF  DD=DENOMINATOR DF
20030 'OUTPUT
    : FB= PROBABILITY OF F
20040 '//////////////////////
20050 X#=1
    :IFFR<1THEN20070
20060 S#=DN
    :T#=DD
    :ZZ#=FR
    :GOTO20080
20070 S#=DD
    :T#=DN
    :ZZ#=1/FR
20080 J5=2/9/S#
    :K5=2/9/T#
    :Z#=K5*ZZ#[(2/3)+J5
    :GOSUB20150
    :Y#=ABS((1-K5)*ZZ#[(1/3)-1+J5)/Z3#
    :IFT#<4THEN20100
20090 X#=.5/(1+Y#*(.196854+Y#*(.115194+Y#*(.000344+Y#*.019527))))[4
    :X#=INT(X#*10000+.5)/10000
    :GOTO20110
20100 Y#=Y#*(1+.08*Y#[4/T#[3)
    :GOTO20090
20110 IFF#>1THEN20130
20120 X#=1-X#
20130 FB=1-X#
20140 RETURN
20150 Z3#=SQR(Z#)
    :Z3#=(Z3#+Z#/Z3#)/2
    :Z3#=(Z3#+Z#/Z3#)/2
    :RETURN
40000 M=200
    :N=11
    :DIM N$(N),A(M,N),B(N+3),Y(M,N),CI(N),XT(17),YT(17),XI(N,N),YN$(6),XZ(N)
40015 MV=-99.99
    :FM$="FALSE"
40020 RETURN
42000 CC=LEN(T$)
    :ZZ=INT(30-CC/2)
    :PRINT TAB(ZZ);T$
    :RETURN
    :'CENTERING ROUTINE
43000 END
```

APPENDIX III. Source code listing for program MULPPLT/BAS

```
1 CLS
    :CLEAR550
    :PRINT"MULTIPLE PLOT ROUTINE VERSION 8-24-80"
    :PRINT"NORTHERN HARDWOODS LABORATORY, USFS, MARQUETTE, MI."
    :GOSUB4000
    :PRINT
    :GOTO33
2 IN$=""
    :W$=INKEY$
    :WD=0
    :WS=WD
    :WL%=WD
    :IFFL=WDTHENFL=1
3 PRINTSTRING$(ABS(FL),CHR$(136));STRING$(ABS(FL),CHR$(24)
);
4 PRINTCHR$(14);
    :FORW%=1TO25
    :W$=INKEY$
    :IFW$<>""THEN5
ELSENEXT
    :PRINTCHR$(15);
    :FORW%=1TO25
    :W$=INKEY$
    :IFW$<>""THEN5
ELSENEXT
    :GOTO4
5 IFW$<>CHR$(13)THEN7ELSEPRINTSTRING$(ABS(FL)-WL%," "
);
6 PRINTCHR$(15);
    :W%=25
    :NEXT
    :RETURN
7 PRINTCHR$(14);
    :IFW$=CHR$(24)THENPRINTSTRING$(WL%,CHR$
(24));
    :GOTO2
8 IFW$<>CHR$(8)THEN12ELSEIFWL%=0THEN4ELSEPRINTCHR$(24);
    :IFFL>0THEN10ELSEIFPEEK(16418)=44THEN11
9 IFPEEK(16418)=46THENWD=0
    :GOTO10ELSEIFPEEK(16418)=43
ORPEEK(16418)=45THENWS=0
10 IN$=LEFT$(IN$,LEN(IN$)-1)
11 WL%=WL%-1
    :POKE16418,136
    :GOTO4
12 IFABS(FL)=WL%THEN4ELSEIFFL>0THENIFW$>=" "ANDW$<=
"Z"THEN17
13 IFW$="."ANDWD=0THENWD=1
    :GOTO17
14 IFW$=","THENPRINTW$;
    :WL%=WL%+1
    :GOTO18
15 IF(W$="-"ORW$="+")ANDWS=0ANDWL%=0THENWS=1
    :GOTO17
16 IFW$<"0"ORW$>"9"THEN4
17 PRINTW$;
    :IN$=IN$+W$
    :WL%=WL%+1
18 IFABS(FL)=1THEN6ELSE4
19 CLS
```

```
    :PRINT@448,"INSERT DATA DISK <ENTER> FILENAME OF DESIRED DATA SET --> ";
    :INPUTF$
    :FJ$=F$
    :FF$=F$+"N"
    :F$=F$+"M"
    :G$=FJ$+"L"
    :OPEN"I",1,FF$
    :INPUT#1,R,C
    :CLOSE1
    :GOSUB130
    :OPEN"I",1,F$
    :FORI=1TOR
    :FORJ=1TOC
    :INPUT#1,A(I,J)
    :NEXTJ,I
    :CLOSE1
20 OPEN"I",1,G$
    :FORI=1TOC
    :INPUT#1,N$(I)
    :NEXT
    :CLOSE
    :RETURN
21 CLS
    :PRINT@448,"INSERT DATA DISK <ENTER> FILENAME OF NEW DATA SET --> ";
    :INPUTF$
    :FJ$=F$
    :PRINT
    :PRINT"ENTER DISK DRIVE # WHERE DATA SHOULD BE SAVED --> ";
    :INPUTDD$
    :FF$=F$+"N:"+DD$
    :F$=F$+"M:"+DD$
    :G$=FJ$+"L:"+DD$
22 OPEN"O",1,FF$
    :PRINT#1,R;C;
    :CLOSE
    :OPEN"O",1,F$
    :
23 FORI=1TOR
    :FORJ=1TOC
    :PRINT#1,A(I,J);
    :NEXTJ,I
    :CLOSE1
    :OPEN"O",1,G$
    :FORI=1TOC
    :PRINT#1,STR$(I);",";
    :NEXT
    :CLOSE
    :RETURN
24 END
25 REM
26 CLS
    :JJ=INT(C/4.1+1)
    :II=INT(R/13.1+1)
    :
    :
    :FORI=1TOII
    :
    :IFR-13*I>=0.0THENN=13ELSEN=13+(R-13*I)
27 FORJ=1TOJJ
    :
```

```
    :IFC-4*J>=0.0THENL=4ELSEL=4+(C-4*J)
28 CLS
    :PRINT"VAR:";
    :FORKK=0TO(L-1)
    :PRINTTAB(14*KK+8)(KK+1)+4*(J-1);
    :NEXTKK
    :PRINT
    :PRINT"---------------------------------------------------------------"
    :FORM=1TON
    :PRINT"O";M+13*(I-1);":";
    :FORK=1TOL
    :A=A(M+13*(I-1),K+4*(J-1))
29 PRINTTAB(14*(K-1)+8)A;
    :NEXTK
    :PRINT
    :NEXTM
30 A$=INKEY$
    :IFA$="" GOTO30
31 NEXTJ,I
    :
    :
    :
    :
    :RETURN
32 END
33 GOSUB124
34 PRINT@448,"PLEASE ENTER: "
    :PRINT"MATRIX OF INPUT DATA FROM (1) DISK (2)  FUNCTION --> ";
    :FL=-1
    :GOSUB2
    :B=VAL(IN$)
    :IFB=0ORB>2THEN34
35 IFB=1GOSUB19ELSEGOSUB119
36 CLS
    :PRINT@448,"PLEASE ENTER: DISPLAY MATRIX ON VIDEO? (Y/N) -> ";
    :FL=1
    :GOSUB2
    :Q=(IN$="Y")OR(IN$="N")
    :IFQ=0GOTO36
37 IFIN$="Y"THENGOSUB26
38 CLS
    :PRINT
    :PRINT
    :PRINT
    :PRINT
    :PRINT"DO YOU WANT TO SEE VAR. #'S AND LABELS (Y/N) -->";
    :FL=1
    :GOSUB2
    :IFIN$="Y"GOSUB127ELSE39
39 CLS
    :PRINT@192,"PLEASE DESIGNATE X-AXIS VARIABLE  # --> ";
    :FL=-2
    :GOSUB2
    :ZX=VAL(IN$)
    :IFZX=0ORZX>CTHEN39ELSEPRINT
40 PRINT"HOW MANY VARIABLES WILL BE PLOTTED ON Y-AXIS (<=6) --> ";
    :FL=-1
    :GOSUB2
    :NI=VAL(IN$)
    :IFNI=0ORNI>6GOTO40
```

```
41 FORCI=1TONI
   :PRINT
42 PRINT"DESIGNATE VARIABLE  # FOR DEPENDENT VARIABLE ";CI;" --> ";
   :FL=-2
   :GOSUB2
   :IW(CI)=VAL(IN$)
   :IFIW(CI)=0ORIW(CI)>CTHEN42
43 NEXTCI
44 CLS
   :PRINT@448,"IS THE MATRIX ALREADY SORTED ON THE X-AXIS (Y/N)? --> ";
   :FL=1
   :GOSUB2
   :IFIN$="Y"THENGOSUB99ELSEIFIN$="N"THEN46ELSE44
45 GOTO65
46 CLS
   :PRINT@458,"PLEASE BE PATIENT.  SORTING IN PROGRESS."
   :YL=9999
   :YH=-9999
   :M=R
47 M=INT(M/2)
   :IFM=0THEN58
48 J=1
   :K=R-M
49 I=J
50 L=I+M
   :FORCI=1TONI
   :ZY=IW(CI)
   :IFA(I,ZY)=-99.99THEN53
51 IFYL>A(I,ZY)THENYL=A(I,ZY)
52 IFYH<A(I,ZY)THENYH=A(I,ZY)
53 NEXTCI
   :IFA(I,ZX)<=A(L,ZX)THEN56
54 FORJJ=1TOC
   :T(JJ)=A(I,JJ)
   :A(I,JJ)=A(L,JJ)
   :A(L,JJ)=T(JJ)
   :NEXTJJ
   :I=I-M
   :IFI<1THEN56
55 GOTO50
56 J=J+1
   :IFJ>KTHEN47
57 GOTO49
58 XL=A(1,ZX)
   :XH=A(R,ZX)
59 IFI>=RTHENGOTO63
60 I=I+1
   :FORCI=1TONI
   :ZY=IW(CI)
   :IFYL>A(I,ZY)THENYL=A(I,ZY)
61 IFYH<A(I,ZY)THENYH=A(I,ZY)
   :NEXTCI
62 GOTO59
63 CLS
   :PRINT@448,"PLEASE ENTER: DISPLAY SORTED MATRIX ON VIDEO (Y/N) ? --> ";
   :FL=1
   :GOSUB2
   :IFIN$="Y"GOSUB26ELSEIFIN$="N"THEN64ELSE63
64 PRINT
   :PRINT"PLEASE ENTER:  SAVE SORTED MATRIX (Y/N) ? --> ";
```

```
   :FL=1
   :GOSUB2
   :IFIN$="Y"GOSUB21ELSEIFIN$="N"GOTO65ELSE64
65 REM
66 CLS
   :T$="GRAPHING STATUS MENU"
   :CC=LEN(T$)
   :GOSUB126
   :PRINTTAB(ZZ);T$
   :PRINT
   :PRINT"  1.  X-VARIABLE = VAR. #";ZX
   :PRINT"  2.  Y-VARIABLE = VAR. #";
   :FORCI=1TONI
   :PRINTIW(CI);
   :NEXTCI
   :PRINT" "
67 PRINT"  3.  RANGE OF X-VARIABLE  --> ";XL;" TO ";XH
   :PRINT"  4.  RANGE OF Y-VARIABLE  --> ";YL;" TO ";YH
   :IFZV=1THENZV$="ON"ELSEZV$="OFF"
68 IFZH=1THENZH$="ON"ELSEZH$="OFF"
69 PRINT"  5.  VERTICAL CONTROL LINE STATUS --> ";ZV$
   :PRINT"  6.  HORIZONTAL CONTROL LINE STATUS --> ";ZH$
   :PRINT"  7.  TITLE OF GRAPH --> ";ZO$
   :PRINT"  8.  X-AXIS LABEL --> ";W1$
   :PRINT"  9.  Y-AXIS LABEL --> ";W0$
70 PRINT" 10.  EXECUTE THE GRAPHING ROUTINE"
   :PRINT" 11.  GET ANOTHER MATRIX                12. REGRESSION PART II
71 PRINT" 13.  DISPLAY MATRIX                    14. PRINT DATA MATRIX"
72 PRINT
   :PRINT"SELECT STATUS CHANGES OR CONTINUATION --> ";
   :FL=-2
   :GOSUB2
   :AN=VAL(IN$)
   :IFAN<=0ORAN>14THEN72
73 ONANGOTO39,39,103,104,105,107,109,110,111,75,132,112,113,114
74 GOTO66
75 CLS
   :PRINT@448,"PRESS <ENTER> WHEN PRINTER IS READY.";
   :INPUTAN
   :I=1
   :J=ZX
   :Z5=XL
   :Z6=XH
   :Z1=YL
   :Z2=YH
   :ZO(13)=Z1
   :ZO(14)=(Z2-Z1)/5.0
   :Z7=INT(30-LEN(ZO$)/2)
   :Z8=INT(30-LEN(W0$)/2)
   :Z9=INT(30-LEN(W1$)/2)
   :LPRINTTAB(Z7);ZO$
   :LPRINT" "
76 LPRINTTAB(Z8);W0$
   :LPRINTTAB(Z9);W1$
   :LPRINT
   :LPRINTTAB(6);
   :FORZ8=0TO5
   :ZO(15)=ZO(13)+ZO(14)*Z8
   :LPRINTUSING"+#.##[[[[";ZO(15);
   :IFZ8<5THENLPRINT" ";
```

```
77 NEXTZ8
    :Z0(14)=(Z2-Z1)/50
    :LPRINT
    :Z0(11)=(Z6-Z5)/50
    :Z7=1
    :FORI8=0TO52
    :FORZ9=0TO52
    :Z1$(Z9)=" "
    :NEXTZ9
    :IFI8=0GOTO95
78 IFI8=52GOTO88
79 IFZ7>0GOTO86
80 IFI8=1THENGOTO82
81 IFZV=0THENGOTO83
82 FORZ9=3TO51STEP2
    :Z1$(Z9)="-"
    :NEXTZ9
83 FORZ9=1TO51STEP10
    :IFI8=1THENGOTO85
84 IFZ9<>1ANDZH=0ANDZV=0THENGOTO88
85 Z1$(Z9)="+"
    :NEXTZ9
    :GOTO88
86 FORZ9=1TO51STEP10
    :IFZH=0ANDZ9<>1THENGOTO88
87 Z1$(Z9)="!"
    :NEXTZ9
88 IFI>RTHENGOTO95
89 IFXL>A(I,J)THENGOTO94
90 IFZ5+(I8-1)*Z0(11)<A(I,J)THENGOTO95
91 FORCI=1TONI
    :J=IW(CI)
    :IFA(I,J)=-99.99THEN93
92 Z1$(INT(1.5+(A(I,J)-Z1)*50/(Z2-Z1)))=WF$(CI)
93 NEXTCI
94 I=I+1
    :J=ZX
    :GOTO88
95 IFZ7>0GOTO97
96 Z0(12)=Z5+Z0(11)*(I8-1)
    :LPRINTUSING"+#.##[[[["";Z0(12);
    :Z7=5
    :GOTO98
97 LPRINT"            ";
98 FORZ9=0TO52
    :LPRINTZ1$(Z9);
    :NEXTZ9
    :LPRINT
    :Z7=Z7-1
    :NEXTI8
    :Z1=Z1+1
    :Z0(0)=55
    :GOTO66
99 XL=A(1,ZX)
    :XH=A(R,ZX)
    :YL=9999
    :YH=-9999
    :FORI=1TOR
    :FORCI=1TONI
    :ZY=IW(CI)
```

```
    :IFA(I,ZY)=-99.99THEN102
100 IFYL>A(I,ZY)THENYL=A(I,ZY)
101 IFYH<A(I,ZY)THENYH=A(I,ZY)
102 NEXTCI
    :NEXTI
    :RETURN
103 XL(1)=XL
    :XH(1)=XH
    :CLS
    :PRINT@448,"PLEASE ENTER NEW MINIMUM VALUE FOR X --> ";
    :FL=-12
    :GOSUB2
    :XL=VAL(IN$)
    :PRINT
    :PRINT"PLEASE ENTER NEW MAXIMUM VALUE FOR X --> ";
    :GOSUB2
    :XH=VAL(IN$)
    :GOTO66
104 YL(1)=YL
    :YH(1)=YH
    :CLS
    :PRINT@448,"PLEASE ENTER NEW MINIMUM VALUE OF Y --> ";
    :FL=-12
    :GOSUB2
    :YL=VAL(IN$)
    :PRINT
    :PRINT"PLEASE ENTER NEW MAXIMUM VALUE OF Y --> ";
    :GOSUB2
    :YH=VAL(IN$)
    :GOTO65
105 IFZV=1THENZV=0ELSEZV=1
106 GOTO65
107 IFZH=1THENZH=0ELSEZH=1
108 GOTO66
109 CLS
    :PRINT@448,"PLEASE ENTER NEW TITLE --> "
    :FL=50
    :GOSUB2
    :ZO$=IN$
    :GOTO66
110 W1$="X-AXIS:   "
    :CLS
    :PRINT@448,"PLEASE ENTER NEW X-AXIS LABEL --> "
    :FL=50
    :GOSUB2
    :W1$=W1$+IN$
    :GOTO66
111 WO$="Y-AXIS:   "
    :CLS
    :PRINT@448,"PLEASE ENTER NEW Y-AXIS LABEL --> "
    :FL=50
    :GOSUB2
    :WO$=WO$+IN$
    :GOTO66
112 RUN"GLR1/BAS"
113 GOSUB25
    :GOTO66
114 GOSUB133
    :AN$="DATA SET NAME --> "
    :CLS
```

```
    :PRINT@448,"PRESS <ENTER> WHEN PRINTER IS READY.";
    :INPUTAN
    :DN$=LEFT$(F$,LEN(F$)-1)
    :DN$=AN$+DN$
    :CC=LEN(DN$)
    :GOSUB126
    :LPRINTTAB(ZZ);DN$
    :LPRINT" "
    :LPRINT" "
    :JJ=INT(C/4.1+1)
    :II=INT(R/200.1+1)
    :FORI=1TOII
115 IFR-200*I>=0.0THENN=200ELSEN=200+(R-200*I)
116 FORJ=1TOJJ
    :IFC-4*J>=0.0THENL=4ELSEL=4+(C-4*J)
117 LPRINT"VAR:";
    :FORKK=0TO(L-1)
    :LPRINTTAB(14*KK+8)(KK+1)+4*(J-1);
    :NEXTKK
    :LPRINT" "
    :LPRINT"--------------------------------------------------------------"
    :FORM=1TON
    :LPRINT"0";M+200*(I-1);":";
    :FORK=1TOL
    :A=A(M+200*(I-1),K+4*(J-1))
118 LPRINTTAB(14*(K-1)+8)A;
    :NEXTK
    :LPRINT
    :NEXTM
    :LPRINT" "
    :LPRINT" "
    :LPRINT" "
    :NEXTJ,I
    :GOTO66
119 CLS
    :PRINT"YOU MUST ENTER THE EQUATION BY FIRST TYPING LINE#122"
    :PRINT"THEN ENTER THE EQUATION -- E.G. 122 Y=X[2  <ENTER> THEN RUN 120"
    :STOP
120 YU=1
    :GOSUB131
    :GOSUB124
121 CLS
    :PRINT@448,"ENTER BEGINNING VALUE OF X -->  ";
    :FL=-12
    :GOSUB2
    :BEGIN=VAL(IN$)
    :PRINT
    :PRINT"ENTER THE ENDING VALUE OF X -->  ";
    :GOSUB2
    :ED=VAL(IN$)
    :PRINT
    :PRINT"ENTER THE INCREMENT OF X --> ";
    :GOSUB2
    :INC=VAL(IN$)
    :I=1
    :GOSUB136
    :I=1
    :FORX=BE TO EDSTEPIN
122 Y1=0.252*X[2.239
    : Y2=0.178*X[2.333
```

```
    : Y3=0.165*X[2.362
123 A(I,1)=X
    :A(I,2)=Y1
    :A(I,3)=Y2
    :A(I,4)=Y3
    :I=I+1
    :NEXTX
    :R=I-1
    :C=4
    :GOTO36
124 DIMZO(20),Z1$(52),Z0$(52),W0$(52),W1$(52)
    :DM=500
    :ZV=1
    :ZH=1
    :Z0$="FIGURE 1."
    :W0$="Y-AXIS"
    :W1$="X-AXIS"
    :WF$(1)="*"
    :WF$(2)="O"
    :WF$(3)="X"
    :WF$(4)="@"
    :WF$(5)="#"
    :WF$(6)="&"
    :RETURN
125 FORI=1TO500
    :NEXTI
    :RETURN
126 ZZ=INT(30-CC/2)
    :RETURN
127 I8=0
    :CLS
    :FORI=1TOC
    :GOSUB128
    :PRINT"VAR #";I;" = ";N$(I)
    :NEXT
    :INPUT"<ENTER> TO RETURN";A9
    :RETURN
128 I8=I8+1
    :IFI8<=10THEN129ELSEINPUT"<ENTER> FOR CONT.";A9
    :I8=1
129 RETURN
130 DIMA(R,C),T(C)
    :RETURN
131 DIMA(200,7),T(7)
    :RETURN
132 RUN
133 IFYU<>1THEN135
134 CLS
    :INPUT"<ENTER> NAME FOR DATA SET";F$
135 RETURN
136 PRINT
    :PRINT"COMPUTING VALUES"
    :RETURN
4000 A$=INKEY$
    :IFA$=""GOTO4000
```

```
    :LPRINT"0";M+200*(I-1);":";
    :FORK=1TOL
    :A=A(M+200*(I-1),K+4*(J-1))
118 LPRINTTAB(14*(K-1)+8)A;
    :NEXTK
    :LPRINT
    :NEXTM
    :LPRINT" "
    :LPRINT" "
    :LPRINT" "
    :NEXTJ,I
    :GOTO66
119 CLS
    :PRINT"YOU MUST ENTER THE EQUATION BY FIRST TYPING LINE#122"
    :PRINT"THEN ENTER THE EQUATION -- E.G. 122 Y=X[2  <ENTER> THEN RUN 120"
    :STOP
120 YU=1
    :GOSUB131
    :GOSUB124
121 CLS
    :PRINT@448,"ENTER BEGINNING VALUE OF X -->  ";
    :FL=-12
    :GOSUB2
    :BEGIN=VAL(IN$)
    :PRINT
    :PRINT"ENTER THE ENDING VALUE OF X -->  ";
    :GOSUB2
    :ED=VAL(IN$)
    :PRINT
    :PRINT"ENTER THE INCREMENT OF X --> ";
    :GOSUB2
    :INC=VAL(IN$)
    :I=1
    :GOSUB136
    :I=1
    :FORX=BE TO EDSTEPIN
122 Y1=0.252*X[2.239
    : Y2=0.178*X[2.333
```

```
    : Y3=0.165*X[2.362
123 A(I,1)=X
    :A(I,2)=Y1
    :A(I,3)=Y2
    :A(I,4)=Y3
    :I=I+1
    :NEXTX
    :R=I-1
    :C=4
    :GOTO36
124 DIMZ0(20),Z1$(52),Z0$(52),W0$(52),W1$(52)
    :DM=500
    :ZV=1
    :ZH=1
    :Z0$="FIGURE 1."
    :W0$="Y-AXIS"
    :W1$="X-AXIS"
    :WF$(1)="*"
    :WF$(2)="O"
    :WF$(3)="X"
    :WF$(4)="@"
    :WF$(5)="#"
    :WF$(6)="&"
    :RETURN
125 FORI=1TO500
    :NEXTI
    :RETURN
126 ZZ=INT(30-CC/2)
    :RETURN
127 I8=0
    :CLS
    :FORI=1TOC
    :GOSUB128
    :PRINT"VAR #";I;" = ";N$(I)
    :NEXT
    :INPUT"<ENTER> TO RETURN";A9
    :RETURN
128 I8=I8+1
    :IFI8<=10THEN129ELSEINPUT"<ENTER> FOR CONT.";A9
    :I8=1
129 RETURN
130 DIMA(R,C),T(C)
    :RETURN
131 DIMA(200,7),T(7)
    :RETURN
132 RUN
133 IFYU<>1THEN135
134 CLS
    :INPUT"<ENTER> NAME FOR DATA SET";F$
135 RETURN
136 PRINT
    :PRINT"COMPUTING VALUES"
    :RETURN
4000 A$=INKEY$
    :IFA$=""GOTO4000
```


U.S. GOVERNMENT PRINTING OFFICE:1983-666-763/5044

Rauscher, Harald M.

The microcomputer scientific software series 2: general linear model—regression. Gen. Tech. Rep. NC-85. St. Paul, MN: U.S. Department of Agriculture, Forest Service, North Central Forest Experiment Station; 1983. 52 p.

The general linear model regression (GLMR) program provides the micro-computer user with a sophisticated regression analysis capability. The output provides a regression ANOVA table, estimators of the regression model coefficients, their confidence intervals, confidence intervals around the predicted Y-values, residuals for plotting, a check for multicollinearity, a check for autocorrelation, and the scaled regression coefficients. A plotting routine is part of the regression program to facilitate quick plotting of residuals.

KEY WORDS: Regression, microcomputer, BASIC, statistics.