United States Department of Agriculture

Forest Service

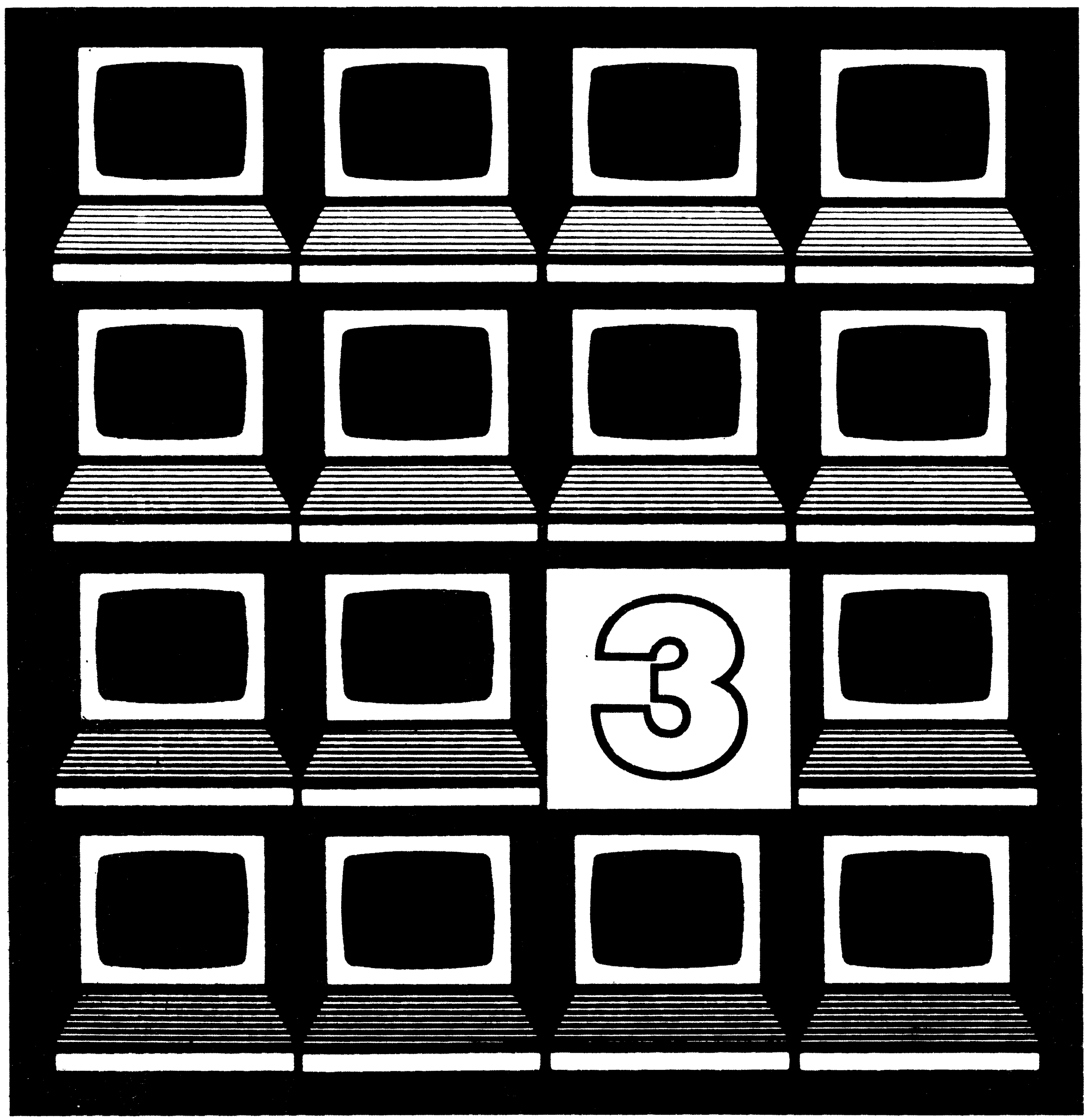
North Central Forest Experiment Station

General Technical Report **NC-86**

The Microcomputer Scientific Software Series 3

General Linear Model - Analysis of Variance

Harold M. Rauscher



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
1983

CONTENTS

THE MICROCOMPUTER SCIENTIFIC SOFTWARE SERIES 3: GENERAL LINEAR MODEL—ANALYSIS OF VARIANCE

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

GENERAL INFORMATION

Identification

Name: General Linear Model–ANOVA (GLMA)
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

Program GLMA performs an analysis of variance (ANOVA). No a-priori restrictions are imposed on the types of experimental designs that can be analyzed. Unbalanced as well as balanced designs may be analyzed and the linear model may contain nested or cross classified factors. The user is required to set up estimable functions that lead to testable hypotheses. An example illustrates how to obtain the proper estimable functions to compute the most frequently needed sums of squares for hypothesis testing.

The GLMA program displays the overall model ANOVA table, the degrees of freedom, sums of squares, and mean square for each hypothesis to be tested. The user must select the appropriate error mean square for each F-test, compute the F statistic, and look up its significance level in a table of F-values.

The current version of GLMA does not perform covariance analysis, compute variance components, provide postiori tests on the hypotheses, nor estimate the expected values of the model parameters or their confidence intervals.

Objectives

The GLMA program and user's guide will:

(a) Provide users of small computers with a general purpose analysis of variance program, and

(b) Document the program to allow user implementation of extensions.

Intended Readership

Self-sufficiency in the area of computer data analysis is becoming increasingly more important, desirable, and financially attractive. As used above, self-sufficiency in data analysis is defined as (1) immediate, desk top access to data analytical tools, (2) cost of computer use not a function of time on the system, and (3) the power to add to, delete from, and alter the data analysis software. This is one of a series of BASIC lanquage programs designed to help interested users achieve data analysis self-sufficiency. The most important non-equipment related requirements to achieve data analytical self-sufficiency is the ability of the user to program. Those who choose not to be programmers should either have access to a programmer or depend upon packaged data analysis systems to meet their needs.

All users of BASIC programs should find this software of value either directly or as a source of computer algorithms. Users of small computers with Microsoft (tm) BASIC will be able to implement GLMA with a minimum of effort and programming knowledge. Users of small computers with other BASIC language implementations will need to reprogram

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

certain basic statements. Frederick *et al.* (1979) have prepared an excellent guide for translating programs between dialects of BASIC.

All users should be familiar with the assumptions behind analysis of variance, how to construct a design matrix based on a linear model, and how to select the appropriate error mean square for each F-test.

The numerical methods are presented to allow the reader to judge their suitability.

References

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

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

Frederick, J. N.; Ferguson, S. Hirst; Trotter, P., eds. CONDUIT Basic Guide. CONDUIT, P.O. Box 388, Iowa City, Iowa 52244; 1977; 233 p.

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

Goodnight, J. H. Tests of hypotheses in fixed effects linear models. SAS Tech. Rep. R-101, SAS Institute Inc.; Cary, N.C. 1978. 15 p.

Helwig, J. T.; Council, K. A. eds. SAS user's guide. Cary, N.C.: SAS Institute Inc.; 1979. 494 p.

Nash, J. C. 1979. 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; 1983a. 44 p.

Rauscher, H. 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; 1983b. 52 p.

Rauscher, H. M.; Buhyoff, G. J. Utility subroutines for data manipulations. Behavior Research Methods and Instrumentation 14: 37-38, 1982.

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

Steele, R. G. D.; Torrie, J. H. Principles and procedures of statistics. New York: McGraw-Hill: 1960. 481 p.

Quick Reference Guide

Input requirements

The user must formulate the statistical model to be tested and label each element of the model as a random or a fixed effect element. GLMA requires that an m by 1 matrix containing the observations on the dependent variable (Y) be defined and located on disk. The design matrix (X) of dimensions m by n must also be created by the user and saved on disk before accessing the analysis of variance program. The m stands for the number of observations and the n stands for the number of elements (columns) in the design matrix.

GLMA Output

(a) ANOVA table,

(b) type I sum of squares for each hypothesis,

(c) mean square and degrees of freedom for each hypothesis tested.

APPLICATION

Introduction

This section of the guide shows you how to use GLMA. Each option in the system relevant to analysis of variance is presented, discussed, and illustrated by example.

After the model has been specified, you must set up the design matrix (X). The data from a two-way, unbalanced ANOVA experiment with interaction terms (Searle 1971, p. 289) is presented in table 1 and will be used as an example. The linear model is:

$Y_{ijk} = mu + (alpha)_i + (beta)_j + (gamma)_{ij} + e_{ijk}$

where: i varies from 1 to 3

j varies from 1 to 4

k varies from 0 to 4

With balanced data, every one of the ij cells would have a constant, k, number of observations. In the unbalanced case k varies.

To begin, initiate the operating system and activate the BASIC interpreter of your computer system. Next, run program "GLM1/BAS". The menu of choices displayed in figure 1 will appear on the screen. Option number 1 will redisplay the menu, number 2

Table 1.—*Example data for a two-way, unbalanced ANOVA with interactions terms (Searle 1971, p. 289)*

y	Y	μ	α_1	α_2	α_3	β_1	β_2	β_3	β_4	γ_{11}	γ_{13}	γ_{14}	γ_{21}	γ_{22}	γ_{32}	γ_{33}	γ_{34}	B	E
8	y_{111}	1	1	.	.	1	.	.	.	1	.	.	.	.	.	.	.	μ	ε_{111}
13	y_{112}	1	1	.	.	1	.	.	.	1	.	.	.	.	.	.	.	χ_1	ε_{112}
9	y_{113}	1	1	.	.	1	.	.	.	1	.	.	.	.	.	.	.	χ_2	ε_{113}
12	y_{131}	1	1	.	.	.	.	1	.	.	1	.	.	.	.	.	.	χ_3	ε_{131}
7	y_{141}	1	1	.	.	.	.	.	1	.	.	1	.	.	.	.	.	β_1	ε_{141}
11	y_{142}	1	1	.	.	.	.	.	1	.	.	1	.	.	.	.	.	β_2	ε_{142}
6	y_{211}	1	.	1	.	1	.	.	.	.	.	.	1	.	.	.	.	β_3	ε_{211}
12	y_{212}	1	.	1	.	1	.	.	.	.	.	.	1	.	.	.	.	β_4	ε_{212}
12	y_{221}	1	.	1	.	.	1	.	.	.	.	.	.	1	.	.	.	γ_{11}	ε_{221}
14	y_{222}	1	.	1	.	.	1	.	.	.	.	.	.	1	.	.	.	γ_{13}	ε_{222}
9	y_{321}	1	.	.	1	.	1	.	.	.	.	.	.	.	1	.	.	γ_{14}	ε_{321}
7	y_{322}	1	.	.	1	.	1	.	.	.	.	.	.	.	1	.	.	γ_{21}	ε_{322}
14	y_{331}	1	.	.	1	.	.	1	.	.	.	.	.	.	.	1	.	γ_{22}	ε_{331}
16	y_{332}	1	.	.	1	.	.	1	.	.	.	.	.	.	.	1	.	γ_{32}	ε_{332}
10	y_{341}	1	.	.	1	.	.	.	1	.	.	.	.	.	.	.	1	γ_{33}	ε_{341}
14	y_{342}	1	.	.	1	.	.	.	1	.	.	.	.	.	.	.	1	γ_{34}	ε_{342}
11	y_{343}	1	.	.	1	.	.	.	1	.	.	.	.	.	.	.	1		ε_{343}
13	y_{344}	1	.	.	1	.	.	.	1	.	.	.	.	.	.	.	1		ε_{344}

$$Y = X B + E$$

FORESTRY SCIENCES LABORATORY, U.S. FOREST SERVICE

MENU FOR: GENERAL LINEAR MODEL–ANOVA

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 MODEL

SELECT ONE OF THE ABOVE —

Figure 1.—*Control Menu for Program GLMA.*

will exit the program and return control to the BASIC interpreter, number 3 is used to read a data matrix from the disk, and number 7 allows printing of the data matrix.

If matrix Y is not already stored on disk, select number 5 to enter it from the keyboard. When you are asked for the number of rows and columns (m,n) in the matrix, enter <18,1>. Then, when you are asked to give the first column a name, enter <Y>. Finally, you are asked to enter the data elements. For 1,1 enter <8>, for 2,1 enter <13>, and so on (see data in table 1).

After the 18th element has been entered, you are returned to the menu of options. Select number 6 to display your matrix. Notice that only 13 of the 18 rows 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 four columns, only four columns will be displayed on any given "page". The left and right arrows move the "page" across the desired sections of the matrix. To return to the menu, press the <CLEAR> button.

Selecting option number 4 allows you to save Y on disk. When queried for the file name under which the matrix will be saved, avoid labeling it "Y" because the program uses the disk file names Y, X, XPX, and XPXI to store intermediate results. Next, you are asked to specify the disk where you want to store the matrix. Using the TRS-80, always choose a drive other than drive O. This practice ensures sufficient disk storage capacity for the intermediate matrices.

The design matrix (X) (table 1) can be entered onto disk the same way. Notice, however, the sparcity of 1's in a field of 0's. The X matrix can be created easily by using the numerical information manipulation system (NIMS) (Rauscher 1983a). NIMS was designed to help you manipulate data matrices. It can be used to create an 18 by 16 matrix of zeroes in one command. The alter subroutine in NIMS can then be used to create full or partial columns of 1's.

Once Y and X are stored on disk 1, select number 9 (fig. 1). You are asked to enter the filename of the Y and X matrices. Remember that these filenames are not necessarily "Y" and "X". These are read into the active memory and the program executes. After a variable waiting period, you are asked to enter the number of significant figures in the raw data. Because of errors in the original data and errors due to computer floating point arithmetic, it is not possible to distinguish small singular values from zero singular values. A tolerance value, based on the number of significant figures in the data and on machine precision, is used to identify the effective number of non-zero singular values.

Before the display of the ANOVA table you are asked if you want the grand mean displayed. Including the grand mean causes the ANOVA to dislay the sum of squares that has been accounted for by the mean. This sum of squares is not presented in the "usual" ANOVA table.

The ANOVA display (fig. 2) presents the sum of squares associated with the overall model, the error sum of squares, and the total sum of squares for the example. Additionally, pertinent summaries concerning the dependent variable Y are given. Depressing the <ENTER> key returns you to the query about including the mean or not. Fianlly, the B and the $(X'X)^{-1}$ matrices are saved onto the disk and you are given the chance to review and/or print the $(X'X)^{-1}$ matrix. The second program, GLM2/BAS, then automatically loads and executes.

Program GLM2 reads the X matrix from disk and computes the X'X matrix. After transferring the X'X matrix onto disk, you are given the opportunity to display it on the screen, print it, or continue processing. Next, you are asked if you want the estimable functions that lead to Type I sum of squares. If the answer is yes, the computations proceed and the result is automatically displayed on the video screen (table 2). Enter <CLEAR> and you can choose to print the estimable functions.

The third program o the series, GLM3/BAS, is read into memory, the generalized inverse of X'X and the X'X matrices are read from disk, and the H matrix is computed. After the B vector is read from disk, you are asked to create the test matrix K by supplying the estimable functions. You may enter estimable functions of your own design or you may enter the previously computed estimable functions that will lead to Type I sums of squares. The matrix of estimable functions is a square matrix in which the name of row i corresponds to the name of column i (table 2). For instance, both columns 2-4 and rows 2-4 correspond to the alpha effects in the model. It is best to draw lines across the rows to separate the first row, representing mu, population mean, from the remaining rows. Next, draw a line separating the alpha effect, rows 2, 3, and 4, from the remainder. Draw a line after row 8 to separate out the beta effect. Rows 9 to 16 correspond to the interaction effect, gamma. To create the test matrix, which will compute the sum of squares for the alpha effect, simply enter every non-zero row from the estimable function block that designatesd the alpha effect. In this case, there are two non-zero rows in the alpha effect block, i.e. row 2 and 3. Thus the K matrix will have two rows giving the alpha effect 2 degrees of freedom. Similarly, the beta effect has three non-zero

REGRESSION ANALYSIS OF VARIANCE TABLE

Source	SS	SS	MS	F	Probability
Model	7	81.9988	11.7141	2.0918	0.1399
Error	10	56	5.6		
Total	17	137.999			

Mean of Response Variable Y = 11
Std. Dev. of Error = 2.36643 C.V. of Y = 21.5
Multiple Correlation Coefficient (R**2) = .549199

Figure 2.—*ANOVA display for example problem.*

Table 2.—*Estimable functions that lead to Type I sum of squares for the example problem*

	μ	α_1	α_2	α_3	β_1	β_2	β_3	β_4	γ_{11}	γ_{13}	γ_{14}	γ_{21}	γ_{22}	γ_{32}	γ_{33}	γ_{34}
μ	1	.33	.22	.44	.28	.22	.17	.33	.17	.06	.11	.11	.11	.11	.11	.22
α_1	0	1	-.33	-.67	.33	-.33	0	0	.5	.17	.33	-.17	-.17	-.17	-.17	-.33
α_2	0	0	1	-1	.5	.25	-.25	-.5	0	0	0	.5	.5	-.25	-.25	-.5
α_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
β_1	0	0	0	0	1	-.4	-.2	-.4	.6	-.2	-.4	.4	-.4	0	0	0
β_2	0	0	0	0	0	1	-.33	-.67	.29	-.1	-.19	-.29	.29	.71	-.24	-.48
β_3	0	0	0	0	0	0	1	-1	0	.33	.33	0	0	0	.67	-.67
β_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
γ_{11}	0	0	0	0	0	0	0	0	1	-.33	-.67	-1	1	-1	.33	.67
γ_{13}	0	0	0	0	0	0	0	0	0	1	-1	0	0	0	-1	1
γ_{14}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
γ_{21}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
γ_{22}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
γ_{32}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
γ_{33}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
γ_{34}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

rows, i.e. 5, 6, and 7. These rows are entered as estimable functions, one at a time, thus creating a K matrix with three rows and 3 degrees of freedom for the beta effect sum of squares. The interaction effect, gamma, has only two non-zero rows which become the estimable functions in K, yielding 2 degrees of freedom for the interaction sum of squares. The sum of the degrees of freedom of the alpha, beta, and gamma effects must equal to the overall model degrees of freedom.

When each set of estimable functions has been entered, you are asked whether the M vector (right hand side of the hypotheses) is zero. Usually it is. In the example, assume that M is a zero vector. The program executes the estimability check. If a K matrix is found that is not estimable, you are advised and allowed to build another K matrix. If you used the computed estimable functions, then you will pass the error check.

The program computes and displays the hypothesis sum of squares, the mean square, and the associated degrees of freedom for each set of estimable functions (table 3). The F-statistic, used for testing a null hypothesis, is computed by dividing the mean square for the null hypothesis by an appropriate error mean square. The selection of the denominator mean square depends on the expected mean squares (ems) for each effect in the model. GLMA does not compute ems. The user must first decide whether the model is a fixed effects, random effects, or mixed effects model an then calculate the appropriate ems. Given the appropriate ems and the numerator mean squares calculated by this program, the user can readily compute the appropriate F-statistic to test hypotheses.

Table 3.—*Analysis of variance results for the example two-way classification with interaciton (see Table 1)*

Source	Clf.	Sum of squares	Mean square
Alpha	2	10.50	5.25
beta	3	36.79	12.26
gamma	2	34.71	17.36

PROCEDURES AND REQUIREMENTS

System Characteristics

Program GLMA is written in BASIC for the TRS-80 microcomputer. The computer should have a minimum of 48K of user definable random access memory (RAM), two disk drives, and a lineprinter. Experience with the TRS-80 has shown that it is best to mount the operating disk with the GLMA programs on disk drive 0 (see TRS-80 owners manual) and use disk drive 1 to store and retrieve the data matrices. Because highly compatible versions of BASIC are available for most microcomputer systems, I believe that this program can easily be converted to execute on most computer systems that execute some dialect of the BASIC language.

Performance

Available memory and speed of computation are the primary limits to performing ANOVA using GLMA. The dimensions of the design matrix (X) must be set to the number of observations (rows) in the analysis and the number of effects (columns) in the model to be analyzed. These dimensions will vary between problems. Line number 40000 in each of the three subprograms that make up GLMA contains the BASIC dimension statement that defines the size of the response and design matrices. In general the program can process an X matrix with 100 rows and 25 columns. Using fewer rows increases the maximum number of columns possible and vice versa.

Processing speed varies with the dimensions of the X matrix. The following examples may be used as an index of processing speed: X(12,8) about 4.5 minutes; X(18,16) about 17 minutes; X(37,5) about 7 minutes; and X(24,20) about 37 minutes. Processing times were measured from the start of the program until the overall model ANOVA table was displayed. These times could be improved by a factor of 10 by compiling GLMA using the BASIC compiler available for the TRS-80.

Data Compatibility

Program GLMA uses the data interchange standard (DIS) proposed by Buhyoff *et al.* (1980). This standard is described in detail by Rauscher and Buhyoff (1982). Using a common DIS decreases program development time and isolates system dependencies into small, explicit subroutines. Using a common DIS enforces a "standard" input/output structure on data transferred between disks. This "standard" structure unites otherwise independent data analysis programs. The input matrices required by this program may be created by GLMA or by the numerical information manipulation system (NIMS) reported by Rauscher (1983a). These two programs and many others (Buhyoff *et al.* 1980) are compatible because they both use the DIS. Because the X matrix contains mostly 0's, it is efficient to allow the computer to create a design matrix with all zeroes. The alteration routine in NIMS may then be used to change the appropriate zeroes into 1's.

Analysis of Variance—Theory

Program GLMA is based upon the theory of general linear models presented by Searle (1971). The problem in analysis of variance (ANOVA), as in regression, is to find a solution to:

$$Y = X B + E \qquad (2)$$

where:

Y = the m by 1 matrix of observations on the dependent variable
X = the m by n design matrix
B = the n vector of parameters
$E = Y - XB$ = an m vector of random errors

Every element in E is assumed to have variance $(\text{sigma})^2$ and zero covariance with every other element (Searle 1971). The regression analysis problem is defined when X is of full rank, i.e. $r(X) < n$.

In ANOVA, the X matrix is called the design matrix because it carries the code for the design of the analysis. It contains only 0's and 1's derived directly from the hypothesized design model under investigation. The solution to (2) is:

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

where:

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

However, X'X has no unique inverse because it is not full ranked. X'X does have many generalized inverses, any one of which can be used to solve the problem. Equation (3) may be restated as:

$$\hat{B} = GY \qquad (4)$$

where:

G = any generalized inverse of X.

The singular value decomposition algorithm of Golub and Reinsch (1970) is used to solve (4). A singular value decomposition of an m by n real matrix, X, is any factorization of the form:

$X = U S V'$ (5)

where:

S = an 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.

$G = V S^+ U'$ where: S^+ = as above except $S^+_{ii} = 1/S_{ii}$ (6)

$\hat{B} = (X'X)^{-1} X' Y = V S^+ U' Y$ (Forsythe *et al.* 1977) (7)

$(X'X)^{-1} = V (S^+)^2 V'$ (Nash 1979) (8)

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

$E'E$ = error sum of squares = SSE (9)

$sigma^2 = SSE/(m-r(X))$ (10)

where:

$sigma^2$ = Residual error estimate of variance
m = number of observations
r(X) = rank of X

SST = $Y'Y$ = total sum of squares (11)

SSR = SST -SSE = sum of squares due to regression (12)

SSM = $m * Y^2$ = correction term for the mean (13)

where:

Y = mean of Y

SSRm = SSR−SSM = regression sum of squares corrected for the mean (14)

SSTm = SST−SSM = corrected sum of squares of dependent variable Y (15)

R**2 = SSRm/SSTm = coefficient of determination. (16)

To test hypotheses, we must assume that the errors are normally, independently, and identically distributed. If we can make these assumptions, F(R) (fig. 3A) tests the hypothesis that XB = 0. If F(R) is significant, we conclude that the model accounts for a significant portion of the variation in the y-variable. F(M) (fig. 3B) tests the hypothesis that E(Y) = 0, i.e. the expected value of the mean of the Y observations is zero. F(M) is given by GLMA when the ANOVA display with the mean is requested. F(R) can be computed from the ANOVA display with the mean by first summing up the sums of squares due to the mean and the model after the mean has been accounted for and then dividing by the sum of the model degrees of freedom and the error mean square. F(Rm) (fig. 3C) provides a test of the model E(Y) = XB over and above the mean. F(Rm) is displayed in the ANOVA without the mean displayed. When F(Rm) is significant, we conclude the model satisfactorily accounts for variation over and above the mean. If all three F-statistics are significant, then:

"F(R), F(M), and F(Rm) indicate, respectively, that the model accounts for a significant portion of the variation in y, that the mean is unlikely to be zero, and that the model needs in it something more than the mean to explain variation in y" (Searle 1971, p. 180).

Estimable Functions

Once the sum of squares for the overall model has been computed, we generally wish to decompose this value into the various sum of squares for each main effect and interaction term in the model. This amounts to partitioning the sum of squares of regression (SSRm) after fitting the mean (fig. 3C). The major statistical analysis packages such as SAS (Helwig and Council 1979) partition the model sum of squares automatically by assuming the investigator desires to test "standard" hypotheses. An option is available for the user who needs to test additional hypotheses. GLMA is not capable of automatically partitioning the model sum of squares.

The user must create the matrix of estimable functions (K) for testing of hypotheses:

HO: $K' B = M$ (17)

where:

HO = null hypothesis under test

K = matrix of estimable functions

M = right hand side of the hypothesis (often equal to the zero vector) The B vector is supplied by the program and calculated as in (7).

The GLMA program supplies the user with the X'X matrix, the $X'X^{-1}$ matrix, and the matrix of estimable functions that lead to the computation of Type I sums of squares. Type I sums of squares are discussed by Helwig and Council (1979, p. 254) and by Goodnight (1978, p. 6). This program uses the Forward Doolittle method (Steel and Torrie 1960) on the X'X matrix to compute the estimable functions that lead directly to the Type I sum of squares.

(MODEL: $Y_{ijk} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ijk}$)

	Source	D.F.	SS	MS	F-Statistic
	Model	8	2260 (SSR)	282.5	50.4 = F(R)
(A)	Error	10	56 (SSE)	5.6	
	Total	18	2316 (SST)		

	Source	D.F.	SS	MS	F-Statistic
	Mean:	1	2178 (SSM)	2178	388.9 = F(M)
(B)	Model: $\alpha, \beta, \gamma \mid \mu$	7	82 (SSRm)	11.71	2.1 = F(Rm)
	Error	10	56 (SSE)	5.6	
	Total	18	2316 (SST)		

	Source	D.F.	SS	MS	F-Statistic
	Model: $\alpha, \beta, \gamma \mid \mu$	7	82 (SSRm)	11.71	2.1 = F(Rm)
(C)	Error	10	56 (SSE)	5.6	
	Total	17	138 (SSTm)		

Note: $\alpha, \beta, \gamma \mid \mu$ = model effects after the effect of the mean has been removed.

Figure 3.—*Example of ANOVA tables from program GLMA.*

A testable hypothesis is one that can be expressed in terms of estimable functions. To ensure that the user specified estimable functions are testable, the following identity is verified in the GLMA program:

$$K' = K' H \qquad (18)$$

where:

$H = G X' X$

The sum of squares necessary to test the hypothesis HO: $K'B = M$ is supplied by the program and computed as:

$$SS(HO) = (K'B-M)' (K'GK)^{-1} (K'B-M) \qquad (19)$$

= sum of squares of hypothesis

$$s = r(K') = \text{number of rows in } K' \qquad (20)$$

where:

K' is by definition of full rank

MS(HO) = SS(HO)/s = mean square of hypothesis (21)

Program Code Documentation

Introduction

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, some versions of BASIC allow concatenation of several logical statements in one physical line. Microsoft BASIC uses the colon, ":", as the logical statement delimiter. 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 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. The code is presented in the appendix with only one logical statement per line. Each logical statement may begin either with a linenumber or a colon. One physical line consists of a linenumbered statement and all the colon statements until the next linenumbered statement. Each section of code is explained in the following presentation by reference to physical linenumbers.

GLMA is composed of three separate programs: GLMI/BAS, GLM2/BAS, and GLM3/BAS.

Program GLM1/BAS

Program GLMI/BAS is almost identical to program GLR/BAS, fully documented by Rauscher (1983b). Lines 2382–2395 in GLR have been deleted and line 2415 added to GLM. Line 2415 allows display or printing of the generalized inverse matrix (X'X).

Program GLM2/BAS

Lines 13-690 comprise the common SEED program documented by Rauscher (1983a). In line 1045, the dimensioning routine is accessed in line 40000 and in line 1070, the X matrix is read from the disk. Lines 1080-1110 compute the X'X matrix, transfer it to the input-output matrix, A(.,.), save it to disk, and give the opportunity to display it on the video screen and/or print it on a lineprinter. In lines 1115-1180, the matrix of estimable functions for use in calculating type I sum of squares is computed. The procedure used is the Forward Doolittle method (Goodnight 1978, p. 12-14). The matrix of estimable functions is displayed, the user is asked whether it is to be printed, and the next program is called (line 1190).

Program GLM3/BAS

Lines 13-920 are the SEED program subroutines that accept input from the keyboard, read data from disk, and write data to disk. Lines 1050-1100 read in the matrices, $(X'X)^{-1}$, X'X, and B, and compute the H matrix. Lines 1110-1180 handle the input from the keyboard of the estimable functions, matrix K and matrix M. Subroutine 6000-6100 tests for estimability. Lines 1205-1260 calculate the pieces of equation (17) using the matrix inversion routine (lines 8500-9450) documented by Rauscher (1983a). The degrees of freedom, sum of squares, and mean square appropriate to the estimable functions used is displayed by line 1280. The user is then routed to another hypothesis test or prompted to run another problem.

APPENDIX I. Source code listing for program GLMI/BAS

```
10 CLEAR 500
    :TX=-1
    :GOTO700
    :'GLM1/BAS VERSION 09/01/81
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$(64,"+")
675 FOR N1= 15360 TO 16383 STEP 64
676 A$=" "
679 FOR Q=0TO63
680 Z=PEEK(N1+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 N1
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 MODEL - ANOVA
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 MODEL
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
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(R,C) 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$="1"
    :GOSUB222
    :N=C1
2045 CLS
    :PRINT@458,"PLEASE BE PATIENT.  COMPUTATIONS IN PROGRESS."
    :GOSUB 1000
    :'SVD
2050 IF IE<>0THENCLS
    :PRINT@458,"ERROR RETURN FROM SVD"
    :STOP
2060 WI=10000
    :WM=0
    :FORJ=1TON
    :IFW(J)>WMTHENWM=W(J)
2065 IFW(J)<WITHENWI=W(J)
2070 B(J)=0
    :NEXTJ
    :CLS
    :PRINT@448,"NUMBER OF SIGNIFICANT FIGURES IN RAW DATA? --> ";
    :FL=-1
    :GOSUB15
    :S=VAL(IN$)
    :RE=0.1ES
    :TA=RE*WM
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))L2
2100 NEXTJ
2102 IFRR<NTHENB(N+6)=-1ELSEB(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))
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"MODEL";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
    :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$="TRUE"THENR2=RMSS/TMSSELSER2=RSS/TSS
2312 PRINT"MULTIPLE CORRELATION COEFFICIENT (R**2) = ";R2
2345 A$=INKEY$
    :IFA$=""THEN2345
2350 GOSUB44000
2360 IFFM$="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 IFMEAN$="TRUE"THENB(N+2)=1ELSEB(N+2)=2
2390 CLS
    :PRINT@448,"SAVING THE COEFFICIENT MATRIX "B" TO DISK."
2400 B(N+8)=N
    :B(N+1)=ESS/(M-RR)
    :B(N+3)=RR
    :FORI=1TON+8
    :A(I,1)=B(I)
    :NEXTI
    :N$(1)="1"
    :FF$="BN:1"
    :L$="BL:1"
    :F$="BM:1"
    :R=N+8
    :C=1
    :GOSUB225
2402 CLS
    :PRINT@448,"COMPUTING INVERSE(X'X) AND SAVING IT TO DISK."
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
    :F$="XPXI"
    :BB$="1"
    :R=C1
    :C=C1
    :GOSUB222
2415 CLS
    :PRINT@448,"CARE TO <S>EE OR <P>RINT THE X'X-1 MATRIX OR "
    :PRINT"<C>ONTINUE? --> ";
    :FL=1
    :GOSUB15
    :IFIN$="C"THEN2420ELSEIFIN$="S"THENGOSUB440ELSEIFIN$="P"THENGOSUB620
2416 GOTO2415
2420 RUN"GLM2/BAS"
2430 END
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=50
   :N=25
   :DIM N$(N),A(M,N),W(N),U(M,N),V(N,N),RV1(N),Y(M,1),B(N+8),SV(N)
40015 MV=-99.99
40020 RETURN
42000 CC=LEN(T$)
   :ZZ=INT(30-CC/2)
   :PRINT TAB(ZZ);T$
   :RETURN
   :'CENTERING ROUTINE
43000 END
44000 A$=INKEY$
   :IFA$=""THEN44000ELSEIFASC(A$)=64THENGOSUB670
44010 RETURN
```

APPENDIX II. Source code listing for program GLM2/BAS

```
10 CLEAR 500
   :TX=-1
   :GOTO1041
   :'GLM2/BAS  VERSION 10-29-80
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$(64,"+")
689 FORN=1TO3
    :LPRINT "  "
    :NEXT
690 RETURN
1000 '*****************************************
1010 'ANOVA TESTS OF HYPOTHESES
1020 'INPUT REQUIRED IS XPXI,B,Y,X FROM DISK
1030 '
1040 '///////////////////////
1041 CLS
    :PRINT@458,"READING THE X MATRIX FROM DISK."
1045 GOSUB 40000
    :'DIMENSIONING ROUTINE
1070 F$="X"
    :GOSUB3135
    :'READ X FROM DISK
1075 FORI=1TON
    :X$(I)=N$(I)
    :NEXTI
    :'CREATE GUIDE FOR DESIGN MATRIX
1078 CLS
    :PRINT@458,"COMPUTING THE X'X MATRIX."
1080 FORI=1TOC
    :FORJ=1TOC
    :XPX(I,J)=0
    :FORK=1TOR
    :XPX(I,J)=XPX(I,J)+A(K,I)*A(K,J)
    :NEXTK,J,I
    :'COMPUTE XPX MATRIX-NOTE X'-->A(K,I)
1085 R=C
```

```
     :FORI=1TOC
     :FORJ=1TOC
     :A(I,J)=XPX(I,J)
     :NEXTJ,I
     :CLS
     :PRINT@448,"SAVING XPX MATRIX TO DISK."
     :F$="XPX"
     :DD$="1"
     :GOSUB222
1090 CLS
     :PRINT@448,"DISPLAY THE XPX MATRIX? (Y/N) --> ";
     :FL=1
     :GOSUB315
     :IFIN$="N"THEN1110
1100 GOSUB440
1110 CLS
     :PRINT@448,"PRINT THE XPX MATRIX? (Y/N) --> ";
     :FL=1
     :GOSUB315
     :IFIN$="N"THEN1115ELSEF$="XPX "
     :GOSUB620
1115 CLS
     :PRINT@448,"COMPUTE TYPE I ESTIMABLE FUNCTIONS? (Y/N) --> ";
     :FL=1
     :GOSUB315
     :IFIN$="N"THEN1190
1120 CLS
     :PRINT@448,"COMPUTING THE TYPE-I ESTIMABLE FUNCTIONS."
1130 FORI=1TOC
     :FORJ=1TOC
     :A(I,J)=0
     :B(I,J)=0
     :NEXTJ,I
     :'SET A,B TO ZERO
1140 FORJ=1TOC
     :A(1,J)=XPX(1,J)
     :B(1,J)=A(1,J)/A(1,1)
     :NEXTJ
1150 FORK=2TOC
     :FORJ=KTOC
     :SUM=0
     :FORL=1TOK-1
     :SUM=SUM+A(L,K)*B(L,J)
     :NEXTL
     :A(K,J)=XPX(K,J)-SUM
     :IFABS(A(K,J))<1.0E-5THENA(K,J)=0
1155 NEXTJ
     :FORJ=KTOC
     :IFA(K,K)<>0THENB(K,J)=A(K,J)/A(K,K)ELSE1158
1156 NEXTJ
1158 NEXTK
1160 R=C
     :FORI=1TOC
     :FORJ=1TOC
     :A(I,J)=B(I,J)
     :NEXTJ,I
     :GOSUB3510
1170 CLS
     :PRINT@448,"DO YOU WISH TO PRINT THE ESTIMABLE FUNCTIONS? (Y/N) --> ";
     :FL=1
     :GOSUB15
     :IFIN$="N"THEN1190
1180 F$="ESTIMABLE FUNCTIONS "
     :GOSUB622
```

```
1190 RUN"GLM3/BAS"
40000 M=50
    :N=35
    :DIM N$(N),A(M,N),X(M,N),B(M,N),XPX(M,N),X$(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
44000 A$=INKEY$
    :IFA$=""THEN44000ELSEIFASC(A$)=64THENGOSUB670
44010 RETURN
```

APPENDIX III. Source code listing for program GLM3/BAS

```
5 'CHANGED THE ORDER OF ON ERROR IN STATEMENT 9020-9051
10 CLEAR 500
    :TX=-1
    :GOTO1041
    :'GLM3/BAS VERSION 12-18-81
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)
    :GOSUB334
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
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$(I);",";
    :NEXT
    :CLOSE
270 RETURN
920 GOTO 735
1000 '******************************************
1010 'ANOVA TESTS OF HYPOTHESES
1020 'INPUT REQUIRED IS XPXI,B,Y,X FROM DISK
1030 '
1040 '//////////////////////
1041 CLS
    :PRINT@458,"READING THE G AND XPX MATRICES FROM DISK."
1045 GOSUB 40000
    :'DIMENSIONING ROUTINE
1050 F$="XPXI"
    :GOSUB135
    :'READ XPXI FROM DISK
1060 FORI=1TOR
    :FORJ=1TOC
    :G(I,J)=A(I,J)
    :NEXTJ,I
1070 F$="XPX"
    :GOSUB135
    :'READ X FROM DISK
1075 N=C
    :C8=C
    :FORI=1TON
    :X$(I)=N$(I)
    :NEXTI
    :'CREATE GUIDE FOR DESIGN MATRIX
1085 CLS
    :PRINT@458,"COMPUTING THE H MATRIX."
1090 FORI=1TOC
    :FORJ=1TOC
    :H(I,J)=0
    :FORK=1TOC
    :H(I,J)=H(I,J)+G(I,K)*A(K,J)
    :NEXTK,J,I
    :'COMPUTE THE H MATRIX
1095 CLS
    :PRINT@458,"READING THE B MATRIX FROM DISK."
1100 F$="B"
    :GOSUB135
    :N=A(R,1)
    :FORI=1TON
    :B(I)=A(I,1)
    :NEXTI
    :SIG=A(N+1,1)
    :'READ THE B MATRIX FROM DISK
1110 CLS
    :PRINT"CREATE THE TEST MATRIX (K) OF  K'B=M"
    :PRINT
    :J=1
1120 N=C8
    :FORI=1TON
    :PRINT"ENTER THE DESIGN VALUE FOR "X$(I)" --> ";
    :FL=-8
    :GOSUB15
    :K(I,J)=VAL(IN$)
    :PRINT
```

```
    :NEXTI
    :PRINT
    :PRINT"ANOTHER DESIGN COLUMN? (Y/N) --> ";
    :FL=1
    :GOSUB15
1130 IFIN$="Y"THEN1140ELSE1160
1140 J=J+1
    :IFJ<=NTHEN1120ELSEPRINT"SORRY. NO MORE DEGREES OF FREEDOM REMAINING."
1150 A$=INKEY$
    :IFA$=""THEN1150
1160 JJ=J
    :CLS
    :PRINT"CREATE THE MATRIX M OF K'B=M"
    :PRINT
1165 FORI=1TON
    :M(I)=0
    :NEXTI
    :'INITIATE M AS THE NULL VECTOR
1170 PRINT"IS M THE NULL VECTOR? (Y/N) --> ";
    :FL=1
    :GOSUB15
    :IFIN$="N"THEN1180ELSE1190
1180 FORI=1TOJJ
    :PRINT"ENTER M VALUE FOR CONTRAST # "I" --> ";
    :FL=-4
    :GOSUB15
    :M(I)=VAL(IN$)
    :PRINT
    :NEXTI
1190 GOSUB6000
    :'CHECK FOR ESTIMABILITY
1195 IFTX=-1THEN1110
1200 CLS
    :PRINT@448,"PLEASE BE PATIENT.  COMPUTING SUMS OF SQUARES."
1205 FORI=1TON
    :KB(I)=0
    :NEXTI
1210 FORJ=1TOJJ
    :FORI=1TON
    :KB(J)=KB(J)+K(I,J)*B(I)
    :NEXTI,J
1220 FORI=1TOJJ
    :KB(I)=KB(I)-M(I)
    :NEXTI
    :'K'B-M NOW IN KB(.)
1230 FORI=1TON
    :FORJ=1TOJJ
    :T(I,J)=0
    :FORK=1TON
    :T(I,J)=T(I,J)+G(I,K)*K(K,J)
    :NEXTK,J,I
    :FORI=1TOJJ
    :FORJ=1TOJJ
    :A(I,J)=0
    :FORK=1TON
    :A(I,J)=A(I,J)+K(K,I)*T(K,J)
    :NEXTK,J,I
    :'   K'GK COMPUTED AS A
1235 FORI=1TON
    :FORJ=1TON
    :IFABS(A(I,J))<1E-6THENA(I,J)=0
1236 NEXTJ,I
1240 'DD$="1"
    :F$="KPGK"
```

```
   :R=JJ
   :C=JJ
   :FORI=1TOJJ
   :N$(I)=STR$(I)
   :NEXTI
   :GOSUB222
   :'SAVE K'GK ON DISK #1
1250 IFJJ=1THENA(1,1)=1/A(1,1)
   :GOTO1260
   :ELSEGOSUB8500
   :'IF JJ=1 THEN JUST INVERT, ELSE CALCULATE THE INVERSE
1260 FORI=1TOJJ
   :J(I)=0
   :FORJ=1TOJJ
   :J(I)=J(I)+A(I,J)*KB(J)
   :NEXTJ,I
   :Q=0
   :FORI=1TOJJ
   :Q=Q+KB(I)*J(I)
   :NEXTI
   :' CALCULATION OF Q
1280 CLS
   :PRINT@448,"DF= "JJ"    SS(HO)= "Q"     MS(HO)= "Q/JJ
   :PRINT
   :PRINT"TEST ANOTHER HYPOTHESIS? (Y/N) --> ";
   :FL=1
   :GOSUB15
   :IFIN$="N"THEN1290ELSE1110
1290 CLS
   :PRINT@448,"RUN ANOTHER PROBLEM? (Y/N) --> ";
   :FL=1
   :GOSUB15
   :IFIN$="Y"THENRUN"GLM1/BAS"ELSEEND
6000 '*********************************************
6010 'ROUTINE TO CHECK FOR ESTIMABILITY OF KB=M
6020 ' TEST IS
   : K'H=K'
6030 '/////////////////////////
6035 CLS
   :PRINT@458,"CHECKING FOR ESTIMABILITY."
6040 FORI=1TOJJ
   :FORJ=1TON
   :KH=0
   :FORK=1TON
   :KH=KH+K(K,I)*H(K,J)
   :NEXTK
   :IFABS(KH)<1E-5THENKH=0
6050 IF ABS(K(J,I))>(ABS(KH)+1E-3)ORABS(K(J,I))<(ABS(KH)-1E-3)THEN6080
6060 NEXTJ,I
6070 TX=1
   :RETURN
6080 CLS
   :PRINT@448,"THE FUNCTIONS IN THE K MATRIX ARE NOT ESTIMABLE.  PLEASE TRY AGAI
N."
   :PRINT"<ENTER> TO CONTINUE."
6090 A$=INKEY$
   :IFA$=""THEN6090
6100 TX=-1
   :RETURN
8500 REM ************************************************
8510 REM       SUBROUTINE FOR MATRIX INVERSION
8520 '  EMPLOYS GAUSS-JORDAN ELIMINATION WITH COLUMN
8530 '  SHIFTING TO MAXIMIZE PIVOT ELEMENTS.
8540 '
```

```
8550 '  INPUTS
    :
8560 '    A(I,I) MUST BE A SQUARE, NONSINGULAR MATRIX
8570 '    N = ORDER OF MATRIX
8580 '
8590 '  OUTPUTS
    :
8600 '    A(N,N) CONTAINS THE INVERSE
8610 '    DETM  CONTAINS THE DETERMINANT
8620 '    DTNRM  CONTAINS MAGNITUDE OF DET/EUCLIDIAN NORM
8630 '
8640 '  NOTE
    :  THIS ROUTINE TESTS FOR A SINGULAR MATRIX
8650 '         AND PRINTS AN ERROR OUT IF A(N,N) IS SINGULAR
8660 REM ***************************************************
8670 NN=N
    :N=JJ
8710 PD=1.0
8720 FOR L=1 TO N
8730 DD=0
8740 FOR K=1 TO N
8750 DD=DD+A(L,K)*A(L,K)
8760 NEXT K
8770 DD=SQR(DD)
8780 PD=PD*DD
8790 NEXT L
8800 DETM=1.0
8810 FOR L=1 TO N
8820 J(L+20)=L
8830 NEXT L
8840 FOR L=1 TO N
8850 CC=0.0
8860 M=L
8870 FOR K=L TO N
8880 IF((ABS(CC)-ABS(A(L,K))) > 0.0) GOTO 8910
8890 M=K
8900 CC=A(L,K)
8910 NEXT K
8920 IF L=M GOTO 9010
8930 K=J(M+20)
8940 J(M+20)=J(L+20)
8950 J(L+20)=K
8960 FOR K=1 TO N
8970 S=A(K,L)
8980 A(K,L)=A(K,M)
8990 A(K,M)=S
9000 NEXT K
9010 A(L,L)=1.0
9020 DETM=DETM*CC
9029 ON ERROR GOTO 9380
9030 FOR M=1 TO N
9031 PRINT A(L,M),L,M,CC
9040 A(L,M)=A(L,M)/CC
9050 NEXT M
9051 ON ERROR GOTO 0
9060 FOR M=1 TO N
9070 IF L=M GOTO 9140
9080 CC=A(M,L)
9090 IF CC=0.0 GOTO 9140
9100 A(M,L)=0.0
9110 FOR K=1 TO N
9120 A(M,K)=A(M,K)-CC*A(L,K)
9130 NEXT K
```

```
9140 NEXT M
9150 NEXT L
9160 FOR L=1 TO N
9170 IF J(L+20) = L GOTO  9290
9180 M=L
9190 M=M+1
9200 IF J(M+20) = L GOTO 9220
9210 IF N > M GOTO 9190
9220 J(M+20)=J(L+20)
9230 FOR K=1 TO N
9240 CC=A(L,K)
9250 A(L,K)=A(M,K)
9260 A(M,K)=CC
9270 NEXT K
9280 J(L+20)=L
9290 NEXT L
9300 DETM=ABS(DETM)
9310 DTNRM=DETM/PD
9311 CLS
9312 IF DETM=0.0 GOTO 9370
9320 N=NN
    :RETURN
    :'RETURN N TO ITS ORIGINAL VALUE
9324 END
9370 '       ROUTINE TO HANDLE A SINGULAR MATRIX
9380 CLS
9390 PRINT" AM SORRY.  MATRIX A  IS SINGULAR."
9400 PRINT"A SINGULAR MATRIX HAS NO UNIQUE INVERSE."
9410 PRINT"IF YOU MUST GET AN INVERSE,OBTAIN A PROGRAM"
9420 PRINT"WHICH CAN CALCULATE THE PSEUDO-INVERSE OF A"
9430 PRINT"SINGULAR MATRIX.  "
9450 END
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
```

```
40000 M=35
    :N=35
    :DIM N$(N),A(M,N),B(N),Y(M,1),G(N,N),XPX(N,N),H(N,N),J(N+20),X$(N),T(N,N),K(2
5,20),M(N),KB(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
```


U.S. GOVERNMENT PRINTING OFFICE:1983-666-764/5046

Rauscher, Harald M.

The microcomputer scientific software series 3: general linear model—analysis of variance. Gen. Tech. Rep. NC-86. St. Paul, MN: North Central Forest Experiment Station; 1983. 35 p.

A BASIC language set of programs, designed for use on microcomputers,is presented. This set of programs will perform the analysis of variance for any statistical model describing either balanced or unbalanced designs. The program computes and displays the degrees of freedom, Type I sum of squares, and the mean square for the overall model, the error, and each factor in the model.

KEY WORDS: Balanced designs, BASIC, unbalanced designs, TRS-80, singular value decomposition.