MAC-TR-40

ON-LINE ANALYSIS FOR SOCIAL SCIENTISTS

by

James R. Miller

May 1967

Project MAC

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ABSTRACT

A library of computer routines has been compiled to facilitate the analysis of social science research data. Many of these routines are designed to test statistical hypotheses.

All routines are operated on-line and permit conversational interaction between the user and a time-shared computer. Input data are typed directly into the computer through a teletype console. Explicit typing directions and error diagnostics, where appropriate, are printed out by each routine to guide the input process. Analyses are executed immediately, and computed results are printed out in typical publication language.

These routines are designed primarily for social science researchers who do not possess extensive prior training in mathematics, statistics, or computer operations. They provide a rapid, flexible, and immediately accessible method of testing preliminary hypotheses and hunches on small to intermediate amounts of data. They also provide a useful pedagogical tool for training students in practical data analysis.

Detailed instructions for gaining access to the routines are provided in Appendix A of this paper. References to standard statistical texts are also provided so that the user may obtain more detailed information concerning the assumptions underlying each routine and the criteria for selecting them.

ACKNOWLEDGEMENT

The research described in this paper was supported (in part) by NsG-235 of the National Aeronautics and Space Administration for research studies in the organization and management of research and development. The findings and views reported herein are those of the author and do not necessarily reflect those of the sponsoring agency.

TABLE OF CONTENTS

This empty page was substituted for a blank page in the original document.

SECTION I

INTRODUCTION

During the past four years, a concerted effort has been made to compile a library of working computer programs appropriate for analyzing social science research data. The original impetus for this effort sprang from the strong desire of researchers within the Sloan School of Management to have their data analyzed more rapidly, more efficiently, and more incisively than had been possible previously. This paper describes one of the products of that effort.

The primary objective was to provide working social scientists with a library of analytical (mostly statistical) routines for investigating real-world data. Since data collected by social scientists typically do not satisfy all of the assumptions required to justify parametric statistical analyses, a large number of the routines are non-parametric or distribution-free in nature.

A second related objective was to orient the routines both in structure and in input/output language toward the working social scientist's point of view. Specifically, this means four things.

1) The routines are problem-oriented rather than technique-oriented. The organization of the library and its documentation reflect an assumption on the writer's part that users will start with a statement of their overall research objectives and infer from these objectives which routine or set of routines would be appropriate to analyze their data. In fact, several routines ask the user while he is conversing with them on-line to select the kind of analysis he would like to perform (see Appendix A at the end of this paper for an illustration of the kinds of questions and answers which may transpire on-line between a routine and its user).

2) The routines are oriented toward the type of data structures familiar to social scientists rather than toward the internal structure of the computer. Thus, it is assumed that social scientists will tabulate or cross-tabulate their data in ways which will facilitate a test of their research hypotheses. The tables so created will then contain direct observations or category frequencies arranged by variables, by categories, or by sub-samples, whichever is appropriate to testing the research hypothesis previously formulated. The routines, in turn, reference and discuss these data according to the structure set down by the user.

3) The routines assume that the user is quite knowledgeable with respect to his research problem and the data he has gathered, but that he may be relatively naive with respect to mathematics, statistics, and computer operations. Consequently, the user must formulate his hypotheses, structure his data, and select one or more appropriate tests prior to initiating conversational interaction with the routine. On the other hand, once the routine has been initiated, detailed instructions are given by the routine itself concerning its purpose and scope, its restrictions, the proper way to enter data (with error diagnostics if errors occur), and the proper way to interpret results.

4) In addition to error diagnostics, other man-machine interface aids are incorporated within the routines to bridge the gap between a very demanding computer and a naive user. These additional aids are discussed in Section III.

A third objective was to specialize these routines to provide a rapid, flexible, and immediately accessible means of testing preliminary hypotheses and hunches. Answers so generated could be used to guide further, more extensive analyses. By breaking the overall analytical process down in this two-step manner, it was hoped that substantial long-run savings could be realized in the amounts of time and effort expended both by research personnel and by computing equipment. It is this same two-step philosophy which underlies the frequently used research strategies of pilot sampling and questionnaire pre-testing.

The accessibility and rapidity of these routines is provided by M.I.T.'s two compatible time-sharing systems (i.e., by the on-line facilities of both Project MAC and the Computation Center.) A user can initiate one of these routines almost any time of the day or night, seven days a week. The computations involved in performing a single analysis via each routine typically require much less than one minute of machine time and no more than fifteen minutes of the user's time. Analytical flexibility is provided by frequent choice points programmed into the routines themselves such that the user may decide on-line what kind of analysis to perform next conditional upon the results of previously performed analyses.

A fourth objective was to incorporate as fully as possible into the logic of each routine whatever automatic decisions could be made strictly on the basis of problem and data descriptions provided by the user (e.g., whether to compute a binomial sampling distribution or to approximate it with a normal distribution, depending upon the sample size). This objective follows from the previous assumption that many users would be relatively naive (and probably disinterested) with respect to mathematics, statistics, and computer operations.

SECTION II

ORGANIZATION OF THE EXISTING ROUTINE LIBRARY

The library of existing routines might best be described in terms of a two-way classification. The primary mode of classification refers to the types of research hypotheses typically formulated by social scientists. Categories included therein are:

1) *Homogeneity Tests* (i.e., whether or not a sample or samples could have been drawn from a specified or the same underlying population),

2) *Independence Tests* (i.e., whether or not two or more variables are statistically independent),

3) *Estimation/Prediction Problems* (i.e., attempts to fit specified mathematical curves to data, to arrive at statistical estimates of various numerical parameters, and to test the statistical significance of these parameters).

The secondary mode of classification refers to the operational significance of whatever data have been gathered. Categories included therein are:

1) *Nominal Data* (i.e., data generated by so crude a measuring process that the only legitimate inferences that may be drawn concerning two observations of different numerical value is that they signify differences with respect to the attribute under observation. No additional inferences may be legitimately drawn concerning whether one observed value signifies either more or less of that attribute than another, nor may any legitimate conclusions be drawn concerning the magnitude of such differences),

2) *Ordinal Data* (i.e., data generated by a measuring process of intermediate refinement such that legitimate inferences may be drawn concerning whether one observation signifies more or less of an attribute than another, depending upon the ordinal rank of the observation values. However, the measuring process is not sufficiently refined to permit legitimate conclusions concerning the magnitude of such differences),

3) *Cardinal Data* (i.e., data generated by a measuring process so refined that legitimate conclusions may be drawn from mere inspection of the observation values concerning the magnitude of differences in the extent to which an underlying attribute is possessed).

Combining these two modes of classification creates a table of nine cells. The particular routines contained in each of the nine cells are listed as follows.[1]

[1] Most of the mathematical and statistical theory underlying these routines can be found either in S. Siegel, **Non-Parametric Statistics**, McGraw-Hill, New York, 1956 or in W. L. Hays, **Statistics for Psychologists**, Holt, Rinehart and Winston, New York, 1963. A more complete description of each routine can be found in Appendix B.

Cell 1 – Homogeneity Tests – Nominal Data

a) binomial test
b) Chi square test of homogeneity among independent samples
c) test of percentage or proportion difference between two independent samples
d) sign test of differences between two matched samples

Cell 2 – Homogeneity Tests – Ordinal Data

a) Mann-Whitney (or Wilcoxon) two-sample test for two independent samples
b) Wilcoxon matched-pairs, signed-ranks test

Cell 3 – Homogeneity Tests – Cardinal Data

a) T-test of the difference between means of two independent samples
b) T-test of the mean difference between two matched samples
c) one-way analysis of variance (fixed effects model)
d) two-way analysis of variance (fixed effects model, perfectly balanced design)
e) test for the symmetry of sample data
f) test for the normality of sample data

Cell 4 – Independence Tests – Nominal Data

a) generalized two-way contingency analysis
b) Fisher exact test of small, two-way contingency tables

Cell 5 – Independence Tests – Ordinal Data

a) Kendall rank-order correlation analysis
b) partial correlation analysis

Cell 6 – Independence Tests – Cardinal Data

a) Pearson product-moment correlation analysis
b) partial correlation analysis
c) two-way analysis of variance (analysis of interaction effects)

Cell 7 – Estimation/Prediction Problems – Nominal Data

No such routines currently exist in the library.

Cell 8 – Estimation/Prediction Problems – Ordinal Data

No such routines currently exist in the library.

Cell 9 – Estimation/Prediction Problems – Cardinal Data

a) simple linear regression
b) multiple linear regression
c) simple and multiple linear regression with certain linear constraints on the fitting coefficients
d) polynomial regression
e) polynomial regression with certain linear constraints on the fitting coeeficients
f) one-way analysis of variance (fixed-effects model)
g) two-way analysis of variance (fixed-effects model, perfectly balanced design)

SECTION III

MAN-MACHINE INTERFACE AIDS

In the course of developing these routines, substantial effort was directed toward making them both intelligible to and manageable by a naive user—without requiring extensive prior education and training on his part. Four kinds of man-machine interface aids were built into every routine to satisfy this objective. These four kinds of aids are discussed below.

3.1 IDENTIFICATION AND SELF-DESCRIPTION

All routines start out with a printed statement which identifies them by name and which describes their overall analytical purpose. Additional information is then printed out including:

a) A description of the type of hypothesis to which the routine may be applied as a legitimate test,
b) A description of the type of data required to justify application of the inherent analytical procedures,
c) A list of additional assumptions about the data (e.g., its distribution characteristics) required to support valid interpretation of computed results,
d) A list of computational limitations (e.g., maximum sample size) built into the programming structure of the routine.

3.2 DIRECTIONS FOR ENTERING INPUT DATA

Immediately following the statements of identification and self-description are precise directions for entering input data. Since the scope of these routines is limited to testing preliminary hypotheses, only small to intermediate amounts of data (e.g., no more than a total of 250 data points) are anticipated. In addition, since the routines are operated on-line, data inputs are typed directly into the computer by means of an online teletype console. Precise directions are printed out regarding:

a) Which major parameters of the data structure (e.g., number of samples to be analyzed, number of observations in each sample, etc.) are required to execute the routine,
b) The exact sequence in which to enter these parameters,
c) The exact sequence in which to enter the data.

3.3 UNIFORM INPUT CONVENTIONS

To provide further assistance to the naive or infrequent user of these routines, uniform conventions have been established to control the manner in which information is typed into the computer. These conventions are described below.

1) Only one unit of information (e.g., one parameter value or one observation number) is entered on every input line, and it may be entered anywhere on the line. This relieves the user from having to count spaces horizontally across

a line of input to insure that all data are properly centered within their respective fields.

2) All observations are entered column-wise with respect to the table or matrix into which the user has structured his data. This facilitates rapid visual comparision of typed inputs with tabulated data.

3) All numerical inputs require explicit decimal points. This tends to reduce the likelihood of certain kinds of typing errors resulting in differential misinterpretation of the order of magnitude of observation values. It also permits the user to apply any consistent scaling factor to his data, if desired.

4) Error checks are made on the spot to detect failures to insert explicit decimal points, logically impossible observation numbers, and violations of computational limitations. Whenever such an error is detected, a diagnostic error message is printed out along with a request to correct and re-type the offending pieces of information.

5) In addition, the user is encouraged to review each column of observations visually and to verify that all typed inputs are numerically correct. If one or more incorrectly typed observations are detected, the user is permitted to retype the erroneous numbers.

3.4 UNIFORM OUTPUT CONVENTIONS

Computed results and legitimate interpretations are printed out according to uniform conventions. Whatever computational decisions were made internally by the routine (e.g., that a Fisher exact test instead of a Chi square test was performed on contingency data) are spelled out in a printed message. Computed results are then summarized in a form typically found in social science publications.

SECTION IV

DEBUGGING, ADVANTAGES AND LIMITATIONS

All of the routines have been subjected to substantial debugging effort. Three devices were used over a three-year period to purge them of internal errors. First, the sequence of instructions in each source language program was checked carefully against the mathematical formulas which it was supposed to implement. Second, test data for which the correct results and conclusions had already been derived by independent means were typed into each routine to validate its accuracy. Finally, and most important, when these routines were released for general use, subsequent action was taken to correct whatever additional errors became apparent in the course of their operation.

The advantages of these routines lie primarily in their design characteristics as previously discussed.

1) They provide a rapid, flexible, and immediately accessible means of analyzing small to moderate amounts of data.

2) The fact that they are used on-line and in a conversational mode with the computer permits flexible reformulation and immediate test of hypotheses conditional upon already computed results.

3) They encourage pre-analyzing data prior to running a full-scale analysis, which would normally require substantial expenditures of manpower, energy, and computer time. Just as pilot sampling can guide researchers toward an economically more efficient allocation of their data-gathering resources, so also can pre-analyzing data guide researchers toward a more efficient allocation of their computational resources.

4) They contain an extensive number of internal devices which facilitate considerably the practical task of testing hypotheses and generating research conclusions.

5) Whatever computational decisions depend upon and can be made on the basis of mathematical, statistical, and internal machine considerations are made automatically without taxing the user's knowledge and judgment. This frees the user to concentrate exclusively upon his research problem.

6) Consequently, these routines may be used without extensive prior training in mathematics, statistics, or computer operations.

7) They converse with the user in his language and state conclusions in a form easily transferable to a published report.

8) Finally, they provide an excellent pedagogical vehicle by which to train students in practical data analysis. Although not originally designed for this purpose, experience has shown them to be quite effective as a training tool. By lifting the burden of tedious computation from the student's shoulders, his attention may be directed toward the more important problems of formulating

hypotheses and choosing appropriate ways to test them. In addition, the greater ease in implementing such choices motivates the student to formulate and test many more hypotheses than he might otherwise have done, and hopefully, to learn more from the experience.

On the other hand, these routines do possess distinct limitations.

1) They cannot accept large amounts of data. The constraint here is not the computer but rather the user, who is generally unwilling to type in more than about 250 pieces of information.

2) Frequent use of the same routine can become boring to the user, since he soon learns how to enter his data, and he no longer needs to be led by the hand every step of the way.

3) Finally, these routines provide no assistance in formulating hypotheses, in originally selecting appropriate tests, and in arranging data in a form appropriate to implementing whatever test has been selected. It is uniformly assumed that these tasks have been performed by the user prior to initiating any particular routine.

SECTION V

EXTENSIONS AND PLANS FOR FUTURE DEVELOPMENT

Some of the limitations mentioned in Section IV have already been dealt with in various ways. First, the limited data problem has been partially relieved by splitting each of the main routines into a large system of compatible subroutines, each one of which has been altered to accept substantially greater amounts of input data. These subroutines may then be called (either on-line or within the more frequent batch-processing context) by a main program specially coded for a particular analysis.

An even greater step in the same direction was made when these routines were incorporated within an on-line programming system called OPS–3.[2] Within OPS–3, users have the capability to analyze either small or large amounts of data. In addition, they may type data directly into the computer, or they may submit the same data in the form of punch cards or magnetic tape. Finally, all descriptive and directive information has been deleted so that only results and conclusions are printed out.

Finally, the writer is currently engaged in designing and implementing an on-line interpretive language to help researchers prepare data for whichever tests are indicated by their hypotheses. This language assumes that a large data base has been made available to the computer (e.g., that the complete results of a survey questionnaire have been punched up on cards and transferred to the researcher's on-line disk file) and that the researcher wishes to test many hypotheses concerning many separate segments of the total data base. The interpretive language will provide means by which the researcher can select and prepare any segment of the total data base for test by a specific analytical routine.

In addition to the above plans, which involve a major expansion of the computer's role in the overall research process, it is also planned to add new routines to the current library and to continue improving the existing ones.

[2] For a complete description of the OPS–3 system, see M. Greenberger, M.M. Jones, J. H. Morris, and D. N. Ness, **On-Line Computations and Simulation: The OPS–3 System**, M.I.T. Press, Cambridge, Massachusetts, 1965. Many of the routines described in this paper are discussed in Chapter 12 of the OPS–3 manual.

REFERENCES

1. Greenberger, M., M. M. Jones, J. H. Morris, and D. N. Ness, **On-line Computation and Simulation: The OPS–3 System**, The M.I.T. Press, Cambridge, Massachusetts

2. Hays, W. L., **Statistics for Psychologists**, Holt, Rinehart and Winston, New York, 1963

3. Miller, J. R., *On-Line Analytical Routines for Social Science Research,* **Proceedings of the International Symposium on Mathematical and Computational Methods in Social Sciences**, Paris, 1966

4. Siegel, S., **Non-Parametric Statistics**, McGraw–Hill, New York, 1956

APPENDIX A

INSTRUCTIONS FOR ACCESSING THE LIBRARY

The entire library of routines described in Appendix B exists in the Project MAC 7094 in the disk files of user number T169 2750, and in the M.I.T. Computation Center 7094 in the disk files of user number M4384 4590.

The procedure for gaining access to the routines at either computer is relatively simple. It is outlined below in step-by-step form.

Step 1. Locate a remote console which can communicate with Project MAC or the Computation Center. There are many such consoles scattered around the M.I.T. area Model 35 Teletypes, IBM 1050s, and IBM 2741s.

Step 2. If the console is a Western Electric Model 35 Teletype, proceed to *Step 3.* If the console is an IBM Model 1050 Teletypewriter, perform the following operations in the indicated sequence to reach the Project MAC machine.

A. Turn power on by pushing up the white plastic switch located on the inside face of the control box situated below and usually to the right of the typewriter keyboard.

B. Pick up the dataphone (it looks like an ordinary grey telephone).

C. Press the second plastic button from the left on the row of buttons located below the dial on the dataphone. An ordinary dial tone should be audible. If not, try some of the other buttons, moving to the right.

D. Dial 8.

E. Press the **Hold** button (left-most button in the row of plastic buttons) when you hear a high-pitch whistle over the phone.

F. Wait for the green **Proceed** light on the top right-hand face of the 1050 console. You should now be in contact with the Project MAC computer.

Proceed to *Step 4.*

Step 3. Perform the following operations in the indicated sequence to communicate with Project MAC via teletype console:

A. Turn on power by pressing the **ORIG** button. This button is the left-most of a row of plastic buttons located at the bottom right-hand corner of the face of the teletype console. It should light up when pressed, and an ordinary telephone dial tone may be audible (although not necessarily).

B. Dial 7 on the telephone dial located directly above the **ORIG** button.

C. There should follow a sequence of buzzes, squeals, and chattering noises from the teletype, and some random characters should appear on the printer. If no chattering and no printing occurs, type in any alphabetic letter and push the carriage return. This should induce chattering and printing. Wait at least five seconds after the last noise is heard before attempting to type anything else. You should now be in contact with the Project MAC computer.

Step 4. Type in **LOGIN** *NUMBER NAME* where *NUMBER* is your problem number assigned by Project MAC (e.g., **T169**), and *NAME* is the last six alphabetic characters of your programmer name (e.g., **MILLER**). Be sure to push the carriage return after this line and *every* line of information typed into the machine. Otherwise, the computer will never receive your instructions.

Step 5. The computer will then type back the letter **W** followed by the current time of day. It will then ask you for your **Password**, and turn off the printer so that whatever you type cannot be read by observers. Type in your password, along with a carriage return.

Step 6. Assuming that your problem number, programmer name, and password are acceptable to the computer, and assuming that the computer is not currently being used at capacity, you will be logged in. The computer will print-out a message to this effect, along with some additional information. You will know that the computer has terminated the logging-in process when it prints out the letter **R** (meaning *Ready*), followed by two numbers (indicating how much time was required to log in). If the machine shuts itself off, this indicates that the computer is currently being used to capacity. Try again later.

Step 7. Now type in **LINK XECUTE SAVED T169 2750**, and push the carriage return. Wait for the *ready* signal.

Step 8. Now type in **LINK LITLIB SQZBSS T169 2750**, and push the carriage return. Wait for the *ready* signal.

Step 9. Now type in **LINK BIGLIB SQZBSS T169 2750**, and push the carriage return. Wait for the *ready* signal.

Step 10. Now type in **LINK SPCLIB SQZBSS T169 2750**, and push the carriage return. Wait for the *ready* signal.

Step 11. Now type in **LINK** *RNAME* **SQZBSS T169 2750**, and push the carriage return. *RNAME*, here, is one of the eighteen names of routines displayed in Appendix B. The effect of this and the preceeding commands is to permit you to access whichever analytical routine you wish to operate on-line.

Step 12. Finally, type in **R XECUTE** *RNAME*, where *RNAME* is the name of the routine you wish to operate. The effect of this command is to pass control to routine *RNAME*. All further instructions will be given by the computer. (The routine **LINFIT** is demonstrated in Appendix C. For details of operation for the other routines, see Appendix B.)

Step 13. Repeat steps 11 and 12 for any additional routines you wish to operate, except step 11 need only be performed once for each routine, while step 12 must be repeated each time you wish to operate that routine.

Step 14. After you are through for the day, type **LOGOUT**. The computer will log you out and shut itself off. (If you haven't used the computer during a 60-minute interval, you will be automatically logged out.)

> (NOTE: To reach the library of routines at the M.I.T. Computation Center, the above procedure must be amended in the following ways.
>
> 1. Dial **0** in step 2, D.
> 2. or dial **9** in step 3, B.
> 3. substitute the number pair **M4384 4590** for **T169 2750** in steps 7 and 8.)

APPENDIX B

CURRENT PROGRAM LIBRARY OF ANALYTICAL ROUTINES

The following is a list of main programs currently stored at Project MAC and the Computation Center. These are all statistical routines. Source decks (MADTRN files) are stored separately and will be produced in their entirety upon request. Please direct all requests to Professor Christopher R. Sprague, Room E52-170 (Ext. 6617), Sloan School of Management, M.I.T., Cambridge, Massachusetts.

NAME — FUNCTIONS PERFORMED

1. *ANLVR1* — Performs one-way analysis of variance on N samples of data. Outputs include:

    1. Computed means for each sample;
    2. Computed F-ratio and degrees of freedom;
    3. 1-tail and 2-tail probabilities that an F-ratio at least as large as the one actually observed could have been generated by chance alone from homogeneous samples.

2. *ANLVR2* — Performs two-way analysis of variance on N samples of data. Outputs include:

    1. Computed means for each sample;
    2. Computed F-ratio and degrees of freedom associated with row effects, column effects, and interaction effects, respectively;
    3. 1-tail and 2-tail probabilities that an F-ratio at least as large as each one actually observed could have been generated by chance alone from homogeneous samples;
    4. Percentage reduction in the variance of an estimate of a randomly selected observation realizable from knowing its row position, its column position, and both.

3. *BRNULI* — Computes an exact binomial probability and all tail probabilities associated with any fixed number of successes out of any fixed number of trials under a fixed success probability. Exact and tail probabilities of occurrence are printed out.

4. *CNTING* — Performs a two-way contingency analysis on two discrete variables with up to 25 levels each. Outputs include:

    1. Chi square value;
    2. Exact 1-tail and 2-tail probabilities of occurrence of a Chi square value at least as large as the one generated under the assumption that the two discrete variables are statistically independent;
    3. Measures of the magnitude of whatever association may exist between the two variables.

5. *FISHER*

   Performs a Fisher exact test on small 2 x 2 contingency tables and gives same outputs as CNTING.

6. *HOMNOM*

   Performs a Chi square test of homogeneity on two samples classified into the same number of discrete categories. Outputs include:

   1. Computed value of Chi square;
   2. Probability that samples at least as heterogeneous as the ones compared could have been drawn from the same population.

7. *KENDAL*

   Performs a Kendall Tau rank-order marginal intercorrelation and partial correlation analysis on up to six ordinal or cardinal variables. Outputs include:

   1. All Kendall Tau pair-wise marginal intercorrelation coefficients;
   2. 1-tail and 2-tail probabilities that Tau values at least as large in absolute value could have been generated if the variables were statistically independent;
   3. Selected incomplete and/or complete partial correlation coefficients upon user request.

8. *LINFIT*

   Fits a linear function to collected data via least-squares. Optional constraints may be applied to the fitting coefficients to make them non-negative, add to a constant, etc. If there is only one independent variable, polynomials of various degrees may be fitted to the data. Outputs include:

   1. Optimum least-squares fitting coefficients;
   2. The square root of the proportional variance reduced.

9. *NRMTST*

   Performs a general test of goodness-of-fit between cardinal sample data and an assumed normal population with unknown mean and variance. Outputs include:

   1. Conditional probability of drawing a sample at least as extreme as the sample actually drawn from a normal population.
   2. The outputs of SYMTST.

10. *PARCOR*

    Performs partial correlation analyses on Pearson product-moment and/or Kendall Tau intercorrelations of up to six variables. Outputs include:

    1. Partial correlation coefficients;
    2. No probability of occurrence provided.

11. *PEARSN*

    Same as KENDAL, but with Pearson product-moment correlation coefficients.

12. *PRPNDF*

    Performs a significance-of-difference analysis on two percentages or proportions. Outputs include:

    1. Computed difference;

2. 1-tail and 2-tail probabilities that a difference at least as large as the computed difference could have been generated if the samples from whence the two proportions arose were drawn from the same population.

13. *SINTST*

Performs a binomial test on the signs of differences between matched pairs of either ordinal or cardinal sample observations. Outputs include:

1. The total number of shifts occurring within matched pairs;
2. The number of these which were positive or negative, whichever is greater;
3. 1-tail and 2-tail probabilities that the larger number of shifts or a still larger number could have occurred by chance if both positive and negative shifts were equally likely.

14. *SYMTST*

Performs a test of goodness-of-fit between cardinal sample data and an assumed symmetric (about the mean) population. Outputs include:

1. 1-tail and 2-tail probabilities of drawing a sample at least as asymmetric as the sample actually drawn from a symmetric population.

15. *T-TEST*

Performs a T-TEST on the difference between means of two samples containing cardinal data. Outputs include:

1. Computed value of T;
2. 1-tail and 2-tail probabilities that T values at least as large in absolute value as the computed T value could have been generated if the samples were drawn from the same population with respect to mean;
3. The outputs of U-TEST.

16. *TOTEST*

Performs a T-TEST on the significance of a mean difference between matched pairs of cardinal sample observations. Outputs include:

1. The computed value of T;
2. 1-tail and 2-tail probabilities that T-values at least as large in absolute value as the computed T-value could have been generated if the true mean difference were zero;
3. The outputs of WILCXN;
4. The outputs of SINTST.

17. *U-TEST*

Performs a Mann-Whitney U-TEST on the difference between medians of two samples containing ordinal or cardinal data. Outputs include:

1. Computed value of U;

2. 1-tail and 2-tail probabilities that U values at least as extreme as the computed U value could have been generated if the samples were drawn from the same population with respect to median.

18. *WILCXN*

Performs a Wilcoxon matched-pairs, signed-ranks test on the ranked differences between matched pairs of either ordinal or cardinal sample observations. Outputs include:

1. Computed value of the Wilcoxon statistic T associated with the smaller sum of ranks;
2. 1-tail and 2-tail probabilities that T values at least as small as the computed T value could have been generated by matched samples drawn from the same population;
3. The outputs of SINTST.

APPENDIX C

AN EXAMPLE OF ON-LINE ANALYSIS

This appendix illustrates one of the routines performing on-line analysis. The printout depicts a user attempting to fit (by least squares) a variety of linear functions to some data which he has collected and typed into the computer. Messages sent by the user to the computer are typed in lower-case characters. Messages returned by the computer are typed in upper-case characters. The number printed out at the end of the routine indicates that the entire analysis required 21.606 seconds of machine time for reading, computing results, and printing out conclusions plus 25.133 additional seconds to execute various housekeeping tasks involved in time-sharing. The entire analysis required less than 10 minutes of the user's time.

```
r xecute linfit
W 1328.4


YOU HAVE CALLED FOR A GENERAL ROUTINE TO FIT A LINEAR FUNCTION
THROUGH COLLECTED DATA. LEAST SQUARES IS THE TECHNIQUE BY WHICH
A BEST FIT IS ACHIEVED.


ASSUMPTIONS UNDERLYING THIS ROUTINE ARE -

     1. ALL DATA MUST BE MEANINGFUL ON AT LEAST
        AN INTERVAL SCALE.
     2. NO ASSUMPTIONS NEED BE MADE ABOUT UNDERLYING
        PROBABILITY DISTRIBUTIONS, SAMPLING METHODS, ETC.


LIMITATIONS ON THIS ROUTINE INCLUDE -

     1. NO MORE THAN 20 INDEPENDENT VARIABLES.
     2. NO MORE THAN 250 OBSERVATIONS PER VARIABLE.
     3. NO MISSING DATA ARE PERMITTED.


THE GENERAL FUNCTION WHICH WILL BE FITTED TO YOUR DATA IS -

          Y = A0 + (A1)(X1) + (A2)(X2) + ... + (AN)(XN)

WHERE     Y IS THE DEPENDENT VARIABLE
          X1, X2, ... , XN ARE THE INDEPENDENT VARIABLES
          A0, A1, ... , AN ARE THE LEAST-SQUARES FITTING CONSTANTS.


TYPE IN THE NUMBER OF INDEPENDENT VARIABLES. TYPE
IN A SINGLE NUMBER WITH AN EXPLICIT DECIMAL POINT.

5.        (independent variable number typed in)


TYPE IN THE NUMBER OF OBSERVATIONS ON EACH VARIABLE.
TYPE IN A SINGLE NUMBER WITH AN EXPLICIT DECIMAL POINT.

10.       (observation number typed in)


NOW TYPE IN YOUR  10 OBSERVATIONS ON THE DEPENDENT VARIABLE Y.
TYPE ALL  10 NUMBERS IN A SINGLE COLUMN WITH EXPLICIT DECIMAL POINTS.

2.7
4.6
3.8    )
9.6    |
5.3    |
1.8    }  (data input; with eighth item corrected)
5.2    |
4.8#9  )
8.7
8.3
```

```
REVIEW THE  10 ENTRIES YOU TYPED IN THE ABOVE COLUMN, AND VERIFY
THAT THEY ARE ALL NUMERICALLY CORRECT. IF ALL ARE CORRECT, PUSH
CARRIAGE RETURN. IF ONE OR MORE ARE INCORRECT, TYPE IN A 9.
```

(carriage return pushed at this point)

```
NOW TYPE IN YOUR  10 OBSERVATIONS ON THE INDEPENDENT VARIABLE X( 1).
TYPE ALL  10 NUMBERS IN A SINGLE COLUMN WITH EXPLICIT DECIMAL POINTS.

3.6
5.4
2.6
8.9
7.6
5.7
5.7
8.9
5.8
6.9@9.6
```

(data input; with tenth item corrected)

```
REVIEW THE  10 ENTRIES YOU TYPED IN THE ABOVE COLUMN, AND VERIFY
THAT THEY ARE ALL NUMERICALLY CORRECT. IF ALL ARE CORRECT, PUSH
CARRIAGE RETURN. IF ONE OR MORE ARE INCORRECT, TYPE IN A 9.
```

(carriage return pushed at this point)

```
NOW TYPE IN YOUR  10 OBSERVATIONS ON THE INDEPENDENT VARIABLE X( 2).
TYPE ALL  10 NUMBERS IN A SINGLE COLUMN WITH EXPLICIT DECIMAL POINTS.

8.6
5.8
97.###9.7
6.4
2.3
7.6
4.3
3.2@2.1
3.5
12#.2
```

(data input; with third, eighth, and tenth items corrected)

```
REVIEW THE  10 ENTRIES YOU TYPED IN THE ABOVE COLUMN, AND VERIFY
THAT THEY ARE ALL NUMERICALLY CORRECT. IF ALL ARE CORRECT, PUSH
CARRIAGE RETURN. IF ONE OR MORE ARE INCORRECT, TYPE IN A 9.
```

(carriage return pushed at this point)

```
NOW TYPE IN YOUR  10 OBSERVATIONS ON THE INDEPENDENT VARIABLE X( 3).
TYPE ALL  10 NUMBERS IN A SINGLE COLUMN WITH EXPLICIT DECIMAL POINTS.

5.6
4.7
8.6
3.4
8.9
8.7
2.3
4.3
5.6
7.3
```

(data input; third item is wrong)

```
REVIEW THE  10 ENTRIES YOU TYPED IN THE ABOVE COLUMN, AND VERIFY
THAT THEY ARE ALL NUMERICALLY CORRECT. IF ALL ARE CORRECT, PUSH
CARRIAGE RETURN. IF ONE OR MORE ARE INCORRECT, TYPE IN A 9.

9.     (mode change typed in)

NOW TYPE IN THE ENTRY NUMBER (E.G., ENTRY NO. 2 IN THE ABOVE
COLUMN) AND THE NEW VALUE OF THE FIRST ENTRY TO BE CORRECTED.
TYPE THESE TWO NUMBERS IN A SINGLE COLUMN WITH DECIMAL POINTS.

3.           }
6.8          } (third item number and correct data typed in)

PUSH CARRIAGE RETURN IF ALL CORRECTIONS HAVE BEEN MADE.
TYPE IN 9 IF FURTHER CORRECTIONS ARE TO BE MADE.

       (carriage return pushed at this point)

NOW TYPE IN YOUR  10 OBSERVATIONS ON THE INDEPENDENT VARIABLE X( 4).
TYPE ALL  10 NUMBERS IN A SINGLE COLUMN WITH EXPLICIT DECIMAL POINTS.

5.6
4.5
8.7          }
2.3          }
8.9          }
6.7          } (data input; all items correct)
2.3          }
5.4          }
1.9
4.7

REVIEW THE  10 ENTRIES YOU TYPED IN THE ABOVE COLUMN, AND VERIFY
THAT THEY ARE ALL NUMERICALLY CORRECT. IF ALL ARE CORRECT, PUSH
CARRIAGE RETURN. IF ONE OR MORE ARE INCORRECT, TYPE IN A 9.

       (carriage return pushed at this point)

NOW TYPE IN YOUR  10 OBSERVATIONS ON THE INDEPENDENT VARIABLE X( 5).
TYPE ALL  10 NUMBERS IN A SINGLE COLUMN WITH EXPLICIT DECIMAL POINTS.

1.0
2.1
3.4          }
2.9          }
5.1          }
4.9          } (data input; all items correct)
6.7          }
5.9          }
8.7
7.9

REVIEW THE  10 ENTRIES YOU TYPED IN THE ABOVE COLUMN, AND VERIFY
THAT THEY ARE ALL NUMERICALLY CORRECT. IF ALL ARE CORRECT, PUSH
CARRIAGE RETURN. IF ONE OR MORE ARE INCORRECT, TYPE IN A 9.

       (carriage return pushed at this point)

THIS COMPLETES DATA INPUT.
```

```
NOW INDICATE BY TYPING IN THE APPROPRIATE DIGIT (SEE CODE BELOW)
WHICH TYPE OF ANALYSIS YOU WOULD LIKE TO PERFORM ON THE DATA.

     1. STRAIGHT REGRESSION MODEL AS SHOWN ABOVE.
     2. REGRESSION MODEL WITHOUT CONSTANT-ADDED (A0) TERM.
     3. REGRESSION MODEL WITH ALL NON-NEGATIVE CONSTANTS.
     4. REGRESSION MODEL WITHOUT CONSTANT-ADDED (A0) TERM
        AND WITH ALL REMAINING CONSTANTS NON-NEGATIVE.
     5. REGRESSION MODEL WITHOUT CONSTANT-ADDED (A0) TERM
        AND WITH REMAINING CONSTANTS NON-NEGATIVE ADDING TO 1.

TYPE IN ONE OF THE ABOVE CODE DIGITS.
```

1.   *(code typed in)*

```
RESULTS FOLLOW -

A( 0)=    3.1166
A( 1)=     .5316
A( 2)=     .0718
A( 3)=     .1221
A( 4)=    -.5483
A( 5)=     .1450

SQUARE ROOT OF PROPORTIONAL VARIANCE REDUCED =    .7806


PUSH CARR. RETURN TO PERFORM ANOTHER KIND OF ANALYSIS ON THE SAME DATA.
TYPE IN 1 TO PERFORM THE ENTIRE ANALYSIS AGAIN ON FRESH DATA.
TYPE IN 2 TO STOP COMPLETELY.
```

(carriage return pushed at this point)

```
REFERRING TO THE CODE NUMBERS DISPLAYED ABOVE, SELECT A TYPE OF
ANALYSIS, AND TYPE IN THE APPROPRIATE DIGIT.
```

2.   *(code typed in)*

```
RESULTS FOLLOW -

A( 1)=     .7166
A( 2)=     .2744
A( 3)=     .0781
A( 4)=    -.4637
A( 5)=     .2877

SQUARE ROOT OF PROPORTIONAL VARIANCE REDUCED =    .7732


PUSH CARR. RETURN TO PERFORM ANOTHER KIND OF ANALYSIS ON THE SAME DATA.
TYPE IN 1 TO PERFORM THE ENTIRE ANALYSIS AGAIN ON FRESH DATA.
TYPE IN 2 TO STOP COMPLETELY.
```

(carriage return pushed at this point)

```
REFERRING TO THE CODE NUMBERS DISPLAYED ABOVE, SELECT A TYPE OF
ANALYSIS, AND TYPE IN THE APPROPRIATE DIGIT.

3.     (code typed in)

RESULTS FOLLOW -

A( 0)=     0.
A( 1)=      .5994
A( 2)=      .0696
A( 3)=     0.
A( 4)=     0.
A( 5)=      .2733

SQUARE ROOT OF PROPORTIONAL VARIANCE REDUCED =    .6573


PUSH CARR. RETURN TO PERFORM ANOTHER KIND OF ANALYSIS ON THE SAME DAT
TYPE IN 1 TO PERFORM THE ENTIRE ANALYSIS AGAIN ON FRESH DATA.
TYPE IN 2 TO STOP COMPLETELY.

       (carriage return pushed at this point)

REFERRING TO THE CODE NUMBERS DISPLAYED ABOVE, SELECT A TYPE OF
ANALYSIS, AND TYPE IN THE APPROPRIATE DIGIT.

4.     (code typed in)

RESULTS FOLLOW -

A( 1)=      .5994
A( 2)=      .0696
A( 3)=     0.
A( 4)=     0.
A( 5)=      .2733

SQUARE ROOT OF PROPORTIONAL VARIANCE REDUCED =    .6573


PUSH CARR. RETURN TO PERFORM ANOTHER KIND OF ANALYSIS ON THE SAME DAT
TYPE IN 1 TO PERFORM THE ENTIRE ANALYSIS AGAIN ON FRESH DATA.
TYPE IN 2 TO STOP COMPLETELY.

       (carriage return pushed at this point)

REFERRING TO THE CODE NUMBERS DISPLAYED ABOVE, SELECT A TYPE OF
ANALYSIS, AND TYPE IN THE APPROPRIATE DIGIT.

5.     (code typed in)
```

```
RESULTS FOLLOW -

A( 1)=     .5578
A( 2)=     .0991
A( 3)=    0.
A( 4)=    0.
A( 5)=     .3431

SQUARE ROOT OF PROPORTIONAL VARIANCE REDUCED =   .6482


PUSH CARR. RETURN TO PERFORM ANOTHER KIND OF ANALYSIS ON THE SAME DATA.
TYPE IN 1 TO PERFORM THE ENTIRE ANALYSIS AGAIN ON FRESH DATA.
TYPE IN 2 TO STOP COMPLETELY.


2.                                   (stop command typed in)
  EXIT CALLED. PM MAY BE TAKEN.
R 21.666+25.133
```

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
Massachusetts Institute of Technology
Project MAC

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP
None

3. REPORT TITLE

On-Line Analysis for Social Scientists

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Technical Report, Sloan School of Management, November 1966

5. AUTHOR(S) *(Last name, first name, initial)*

Miller, James R.

6. REPORT DATE
May 1967

7a. TOTAL NO. OF PAGES
32

7b. NO. OF REFS
4

8a. CONTRACT OR GRANT NO.
Office of Naval Research, Nonr-4102(01)

b. PROJECT NO.
NR 048-189

c.
RR 003-09-01

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
MAC-TR-40

9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

None

12. SPONSORING MILITARY ACTIVITY
Advanced Research Projects Agency
3D-200 Pentagon
Washington, D. C. 20301

13. ABSTRACT

A library of computer routines has been compiled to facilitate the analysis of social science research data. Many of these routines are designed to test statistical hypotheses.

These routines are designed primarily for social science researchers who do not possess extensive prior training in mathematics, statistics, or computer operations. They provide a rapid, flexible, and immediately accessible method of testing preliminary hypotheses and hunches on small to intermediate amounts of data. They also provide a useful pedagogical tool for training students in practical data and analysis.

Detailed instructions for gaining access to the routines are provided in Appendix A of this paper. References to standard statistical texts are also provided so that the user may obtain more detailed information concerning the assumptions underlying each routine and the criteria for selecting them.

14. KEY WORDS

Computers
Data analysis
Machine-aided cognition
Multiple-access computers
On-line computers
Real-time computers
Social science research
Time-sharing
Time-shared computers

UNCLASSIFIED
Security Classification