

AD-A063 145

NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF
A USER'S GUIDE TO THE OA3660 APL WORKSPACE.(U)
OCT 78 F R RICHARDS
NPS55-78-028

F/G 9/2

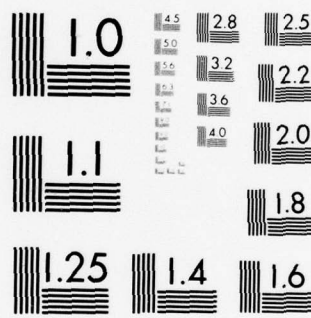
UNCLASSIFIED

NL

1 OF 2

AD
A063145





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD A063145

DDC FILE COPY.

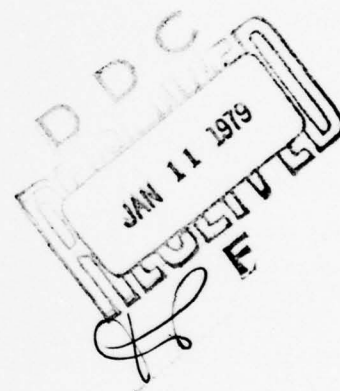
LEVEL

A

2

NPS55-78-028

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THIS DOCUMENT IS BEST QUALITY PRACTICABLE.
THE COPY FURNISHED TO DDC CONTAINED A
SIGNIFICANT NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.

A USER'S GUIDE
TO THE OA3660 APL WORKSPACE

by

F. Russell Richards

October 1978

Approved for public release; distribution unlimited.

79 01 09 027

Naval Postgraduate School
Monterey, California

Rear Admiral T. F. Dedman
Superintendent

Jack R. Borsting
Provost

Reproduction of all or part of this report is authorized.

This report was prepared by:

F. Russell Richards

F. Russell Richards, Associate Professor
Department of Operations Research

Reviewed by:

Released by:

Michael G. Sovereign
Michael G. Sovereign, Chairman
Department of Operations Research

William M. Tolles
William M. Tolles
Dean of Research

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DDC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER (14) NPS55-78-028	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) A User's Guide to the OA3660 APL Workspace.	5. TYPE OF REPORT & PERIOD COVERED (9) Technical Repts	
7. AUTHOR(s) (10) F. Russell Richards	8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE (11) October 1978	13. NUMBER OF PAGES 113
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified	
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE		
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. (12) 44-3 p.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) APL Interactive Programming Data Analysis Exploratory Data Analysis		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Instructions are given for the use of the APL Public Library Workspace, 2 OA3660, which was developed as an aid to interactive exploratory data analysis. The OA3660 workspace is accessible to all users of the computer time sharing system at the Naval Postgraduate School. The workspace contains various data analysis functions, data, and complete internal documentation. This report provides a primer on APL, documentation on each function contained in the workspace, examples of the use of each function, and program listings.		

DD FORM 1 JAN 73 1473

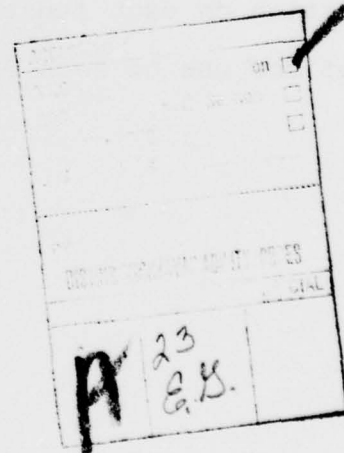
EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 01 09 027 251 450

TABLE OF CONTENTS

	Page
ABSTRACT	2
I. INTRODUCTION	3
II. APL PRIMER	
A. Getting Started: Login	3
B. Generating and Storing Data	10
C. Arithmetic Operations	20
D. Workspace Management	29
E. Error Recovery	34
III. THE OA3660 WORKSPACE	36
A. General Documentation	36
B. Function Descriptions and Examples....	44
APPENDIX A.....	79
APPENDIX B	92
REFERENCES	108



A USER'S GUIDE
TO THE OA3660 APL WORKSPACE

by

F. Russell Richards
Naval Postgraduate School
Monterey, CA 93940

ABSTRACT

Instructions are given for the use of the APL Public Library Workspace, 2 OA3660, which was developed as an aid to interactive exploratory data analysis. The OA3660 workspace is accessible to all users of the computer time sharing system at the Naval Postgraduate School. The workspace contains various data analysis functions, data, and complete internal documentation. This report provides a primer on APL, documentation on each function contained in the workspace, examples of the use of each function, and program listings.

A USER'S GUIDE
TO THE OA3660 APL WORKSPACE

by

F. Russell Richards
Naval Postgraduate School
Monterey, CA 93940

I. INTRODUCTION

This report provides documentation for the OA3660 APL workspace that was developed during the summer and fall quarters of 1977 in conjunction with the offering of the OA3660 course, Data Analysis, at the Naval Postgraduate School. The OA3660 workspace contains APL functions, variables, and data sets that are useful for interactive exploratory data analysis. The APL functions include selected procedures from the STAT101 public library [6], and from Donald R. McNeil's textbook, Interactive Data Analysis: A Practical Primer [5] (with some modifications). The OA3660 workspace also includes functions written by myself and students in my class and various utility functions extracted from other public library workspaces. Documentation for each function is contained in this report and in the OA3660 Public Library Workspace.

The data sets include data contained in examples and exercises in the above mentioned book, data from John Tukey's Exploratory Data Analysis [8], data generated from test scores in the OA3660 class, and data extracted from a few other textbooks. The data are cross referenced with the sources in a variable named DATAMAP.

This report is intended for the user who has some experience with APL and who is taking a course in Data Analysis or one who is already familiar with the basic techniques of interactive exploratory data analysis as described in Tukey [8], McNeil [5], and Mosteller and Tukey [7]. This report will not attempt to explain the analysis techniques themselves, nor will it discuss detailed APL concepts. It will give a brief description of the construction of data arrays and transformations of data since those are key elements in interactive exploratory data analysis. It also includes a brief discussion of basic APL commands such as logging into the system, loading workspaces, etc. For further details on APL and the NPS time-sharing system, CP/CMS, the reader is referred to the Naval Postgraduate School Technical Note No. 0141-33, APL(CMS)-An Introduction [1]; the IBM report #GH20-0906-1, APL\360-OS and APL\360-DOS User's Manual [2]; APL-An Interactive Programming Language [3] by Gilman and Rose; and APL Programming and Computer Techniques [4] by Katzan.

Each function contained in the OA3660 workspace is discussed; the function syntax is given; the function parameters (if any) are described; APL listings of the functions are given; and at least one example is included to illustrate the use of each function. An attempt has been made to assure that every function in the workspace is completely debugged. However, should any problems be experienced, the user is urged to notify me of the problems. Users are also encouraged to submit to me interesting data and APL functions that are useful for interactive exploratory data analysis.

II. APL PRIMER

This chapter is written primarily for the user who is not familiar with the APL programming language. We attempt to provide only that material that the user needs to know about APL to use successfully the OA3660 APL Public Library. Therefore, we discuss in this chapter login procedures, basic APL workspace management, error recovery procedures, creation and storage of data, data transformations, function syntax, and logout procedures. A user conversant with APL should proceed to Chapter III.

A good understanding of APL would enhance the user's facility for working with data arrays and performing interactive exploratory data analysis. Therefore, we encourage the user to seek out more detailed information on APL. References [1,2,3,4] are all recommended.

A. Getting Started: Login

In order to use the OA3660 APL Public Library, the user must have access to a computer terminal that has APL capability linked to the Naval Postgraduate School IBM 360/67 computer. There are many different types of remote terminals available for use,* and each has its unique features of operation. Therefore, the user should check out the operating

*The IBM 2741 terminals with a special APL typing ball, the Intertec terminals, and some of the CRT terminals have APL capability. In addition, there are several school owned portable terminals with APL capability.

instructions for each terminal to augment the general instructions given here.

If the terminal is wired directly to the computer, the user need only turn it on to access the NPS CP/CMS Time Sharing System. If not, the user must link to the computer via an acoustic hookup. First, turn the terminal on to the correct settings. Then dial the appropriate telephone number for connection to CP/CMS.* When the shrill audible tone is received, place the telephone receiver into the acoustic coupler which is connected to the terminal. If all of the terminal settings are correct, and if the time sharing system is in operation, the user should receive the message "CP-67 ONLINE". At this point, the user must log onto the system by typing (with the non-APL character set):

LOGIN XXXXPYY (CR)

where XXXX is the user's identification number assigned by the computer center, P indicates a private user ID (type G if a general user without private disk space), YY is the terminal number, and (CR) indicates a carriage return.# If the identification number is valid, the system will request the user's password.

*The telephone number and the terminal settings depend on the terminal being used. For ASCII terminals (nearly all terminals at NPS other than the IBM 2741 terminals) dial either x2611 (with speed setting at 110) or x3025 (with speed setting at 300). For EBCDIC terminals (IBM 2741), dial x2701.

#On some terminals the user must depress simultaneously the keys CONTROL and S in place of a carriage return even though there may be a key so labelled. We will use (CR) to indicate a carriage return for all terminals.

The assigned password should be typed followed by a carriage return:

PASSWORD (CR)

If the password is incorrect, the system will reject the login and ask the user to start over. If the password is correct, the system will request the user's four-digit project number to be followed by a four-character cost center code. The project number must be assigned by the computer center. The cost center code is the user's section identifier or department code.

PPPPCCCC (CR)

If the project number is acceptable the system will respond with a ready message indicating that the user is in the CMS subsystem of the CP/CMS Time Sharing System. Otherwise, the user must repeat the entire procedure.

Once the user has entered CMS he should switch on the APL character set* and type

APL (CR)

* Switching to the APL character set may consist of flipping a switch, changing a type ball or impact print set, or issuing a sequence of program instructions depending on the type of terminal.

to enter the APL subsystem. The system will then respond with the message:

A * P * L \ N * P * S

LIBRARY DOCUMENTATION SYSTEM...)LOAD 1 LIBDOC TYPE DESCRIBE.*

The user is now in APL, and all of its powerful features are available to him. He may use APL in the calculator mode somewhat as he would use a hand held calculator; he may define his own functions to perform a sequence of operations; or he may access the public libraries which consist of commonly used preprogrammed functions that perform a variety of useful computations.

In the next section we describe briefly the use of APL in the calculator mode to generate, manipulate and store data. In later sections we describe the use of APL to transform the data, and the use of APL Public Library Workspaces. The user is directed to references [1,2,3,4] for information about writing functions in APL, and for more detailed information about primitive operations in APL.

* If any symbol other than the right parenthesis) appears before the word LOAD, the system is not properly translating the character set into the required APL characters. If this happens, the user should type the symbol that appears (represented here by #) followed by the word OFF as follows:

#OFF (CR)

This will get the user off the system at which time he should consult with the computer center staff as to the special requirements for his terminal. Do not proceed in APL until the character ")" appears before the word LOAD in the message typed when APL is entered.

B. Generating and Storing Data

Generation of data arrays is very simple in APL because one need not be concerned with format. One must, however, become familiar with the APL syntax. First is the assignment symbol \leftarrow which plays a role somewhat like the $=$ sign in FORTRAN. It means to take the expression on the right and assign it to the variable named on the left. Suppose, for example, one wants to generate a vector of data named X^* consisting of the four observations: 3.15 12.57 8 6.003. One need only type:

$X \leftarrow 3.15 \ 12.57 \ 8 \ 6.003$ (CR)

with the decimal typed if needed and with one or more spaces serving as delimiters for the separate observed values. (Real numbers may be expressed in scientific format by use of the E notation. $2.5E5$ means 2.5×10^5 or 250000.) Extra spaces before and after the assignment arrow, \leftarrow , are not needed, but they will not hurt anything. Additional values can be added to the vector by using the catenate operator (the comma)[#] as follows:

* Just about any name can be used for variables as long as the first character is alphabetic and the other characters are alphabetic or numeric. The character Δ is also acceptable.

One must take care to strike the characters for the APL character set. Several of the APL characters are identical in appearance on the terminal keyboards to non-APL characters, but different results are obtained. Among the characters that appear the same are $) \ ' \ (\ - \ + \ \times \ \div \ / \ : \ ; \ *$.

```
X ← X, 5.76 ^4 1.47 (CR)
```

This statement says to concatenate the three indicated values to the end of the old data vector X and to call the resulting vector X. The entire vector can be viewed by typing X followed by the (CR)

```
X (CR)
3.15 ^12.57 8 6.003 5.76 ^4 1.47
```

In the above vector, the negative signs are typed using the negative symbol (upper shift 2), not the minus sign (upper shift +).

If one is entering many data values into a vector he may require several lines of input to do so. Additional values could be inserted, as above, using the concatenate operator. However, a simpler way is to type a comma followed by the quad symbol □ (upper shift L) and a carriage return at the end of a line if more values are to be continued on succeeding lines. The next line will automatically begin with the quad symbol prompting the user to continue entering values. This is illustrated below:

```
DATA ← 4 6 8 12 6 ^2 ^4 0 0 1 22 36 29 18, □ (CR)
□: 4 2 12 9 ^3 ^17 9 9 2 ^1 4
```

If a value is typed incorrectly and discovered before the (CR) is struck, the value can be corrected by backspacing to the incorrect entry and hitting the line feed key.* This will erase the incorrect value and everything to the right of the value. Then type the correct values.

Data correction or modification after a line has been terminated by a carriage return can be accomplished several ways. One way is to determine the index of an incorrect entry and assign a new value to that specific element of the data array. For example, suppose one wants to change the third element from 8 to 4.81 leaving all the other elements alone. This can be done by typing:

X[3] ← 4.81 (CR)

Multiple corrections can be made simultaneously as follows:

X[3, 4, 5] ← 4.81 5.9 -3.26 (CR)

The X vector would then contain:

3.15 -12.57 4.81 5.9 -3.26 1.47.

The index operator \downarrow (upper shift I) is convenient for determining the index of a given value (or indices of a set of values) in an array of data. If X is the data array and B is a set of values in X, then $X \downarrow B$ will generate the set of

* On some terminals the line feed key is labelled LF. On others, the attention or break key must be struck.

indices of the values of B in X. If a value appears in more than one place in X, only the subscript of its first appearance is given. For example, if $X \leftarrow 3.5 \text{ } ^{-}12.57 \text{ } 4.81 \text{ } 5.9 \text{ } ^{-}3.26 \text{ } ^{-}4 \text{ } 1.47$ the index of the value 5.9 is found as follows:

$X \text{ } ^{-} 5.9$

4

Let $B \leftarrow ^{-}12.57 \text{ } ^{-}3.26$. The indices of B in X are given by:

$X \text{ } ^{-} B$

2 5

This operation could be useful for altering selected values in an array. To illustrate, assume that we want to change the values $^{-}12.57$ and $^{-}3.26$ in X to 80 and 83. The operation below will accomplish this change:

$X[X \text{ } ^{-}12.57 \text{ } ^{-}3.26] \leftarrow 80 \text{ } 83$

X

3.15 80 4.81 5.9 83 $^{-}4$ 1.47

Additional values can be inserted at any position in the vector, and values can be deleted from the vector by using the take " \uparrow ", drop " \downarrow ", and catenate " $\text{ } ^{-}$ " operations. The take operation with syntax

$r \uparrow X,$

selects the first r elements from the vector X if r is positive and the last $|r|$ elements if r is negative. The drop operation with syntax

$r \uparrow X$

deletes the first r elements of X if r is positive and the last $|r|$ elements if r is negative. If $|r|$ is greater than the number of elements in X , $r \uparrow X$ will insert zeros to the right of the elements of X until $|r|$ elements are obtained and $r \uparrow X$ will result in the empty vector. The examples below demonstrate the take and drop operations.

X

3.15 $\bar{12.57}$ 4.81 5.9 $\bar{3.26}$ $\bar{4}$ 1.47

$3 \uparrow X$

3.15 $\bar{12.57}$ 4.81

$\bar{2} \uparrow X$

$\bar{4}$ 1.47

$9 \uparrow X$

3.15 $\bar{12.57}$ 4.81 5.9 $\bar{3.26}$ $\bar{4}$ 1.47 3.15 $\bar{12.57}$

$2 \downarrow X$

4.81 5.9 $\bar{3.26}$ $\bar{4}$ 1.47

$\bar{4} \downarrow X$

3.15 $\bar{12.57}$ 4.81

$8 \downarrow X$

The take and drop operators can be used in conjunction with the catenate operator to edit data vectors. The examples below illustrate some ways this can be done.

Let

$X \leftarrow 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 10.$

Create a new vector Y consisting of the first four elements of X followed by the elements 11 12 and then the last three elements of X ; i.e. $Y = 1 \ 2 \ 3 \ 4 \ 11 \ 12 \ 8 \ 9 \ 10.$

$Y \leftarrow (4 \uparrow X), 11 \ 12, (\bar{3} \uparrow X) \quad \textcircled{\text{CR}}$

or $Y \leftarrow (4 \uparrow X), 11 \ 12, (7 \uparrow X) \quad \textcircled{\text{CR}}$

Let $W \leftarrow 0 \ \bar{1} \ \bar{2} \ \bar{3} \ \bar{4} \ \bar{5}.$ Create a new vector Y consisting of the first eight elements of X , followed by the number 25, followed by the middle four elements of W .

$Y \leftarrow (8 \uparrow X), 25, (1 \uparrow (5 \uparrow W)) \quad \textcircled{\text{CR}}$

A few comments about the above operation are in order. APL always operates from right to left except when parentheses are used to override the standard order of operation. Thus, $(5 \uparrow W)$ is the first operation executed. This results in the vector $0 \ \bar{1} \ \bar{2} \ \bar{3} \ \bar{4}.$ The next operation is $(1 \uparrow 0 \ \bar{1} \ \bar{2} \ \bar{3} \ \bar{4})$ which gives $\bar{1} \ \bar{2} \ \bar{3} \ \bar{4}.$ This is catenated to 25 yielding $25 \ \bar{1} \ \bar{2} \ \bar{3} \ \bar{4}$ which is then catenated to $(8 \uparrow X)$ giving $1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 25 \ \bar{1} \ \bar{2} \ \bar{3} \ \bar{4}.$

In data analysis one often wants to compare different groups of data or to relate one group to another group. Many of the functions in OA 3660 require the data arrays be matrices whose columns represent the different groups and whose rows represent the various observations for each group. Let us see how a data matrix can be created. Suppose for example that one wants to create the matrix:

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{pmatrix}$$

The easiest way to create A is to create a vector consisting of the eight elements and then reshape the vector into the desired 4×2 array. The reshape operator ρ can be used to reshape a vector into any specified size. The matrix A is created by the operation:

$$A \leftarrow (4,2) \rho \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \quad \textcircled{\text{CR}}$$

The general syntax for the reshape operation is $A \leftarrow (N,M) \rho B$ where B is a vector, or scalar, M is the number of columns of the array A , and N is the number of rows of A . The first M elements of B will be used for row 1, the second M for row 2, etc. If there are less than $M \cdot N$ elements of B , the elements will be repeated from the beginning as many as are needed. If

B contains more than $M \cdot N$ elements, only the first $M \cdot N$ will be used. Try a few examples to see what reshape does.

Remember that the reshape operator fills up matrices row by row. Suppose that the user entered the data into the vector by columns; e.g. $B \leftarrow 1 \ 3 \ 5 \ 7 \ 2 \ 4 \ 6 \ 8$. How can the 4×2 matrix A be generated? $(4,2) \rho B$ will not work since that would give

$$\begin{pmatrix} 1 & 3 \\ 5 & 7 \\ 2 & 4 \\ 6 & 8 \end{pmatrix}$$

However, if one requested $(2,4) \rho B$ one would obtain

$$\begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \end{pmatrix}.$$

What we need now is to transpose this to obtain A. The transpose operator is \mathbb{Q} (upper shift 0 overstruck with upper shift /).^{*} This sequence of operations can be combined as follows:

$$A \leftarrow \mathbb{Q}(2,4) \rho B \quad (\text{CR})$$

The operator ρ is also useful for determining the dimension or shape of an arbitrary variable. When used for this purpose it is called the shape operator. The syntax is

$$\rho Z \quad (\text{CR})$$

^{*}Type shift 0 (oh), backspace, and type upper shift /.

Table 2.1 illustrates the results of the shape operator when applied to selected variables. The first entry is the scalar 3. APL considers all scalars to have no dimension (not dimension 0). Thus, ρZ is the empty vector. In the next to last case, the variable Z is defined to be $\iota 0$ (iota 0). (This represents a second use of the index operator ι where the operator has only a right argument. If K is any nonnegative integer, ιK is the vector consisting of the first K positive integers. For example, $\iota 3$ would be the vector 1 2 3.) The use of the index operator in Table 2.1 is a rather special one. APL interprets $\iota 0$ as a vector with 0 elements, hence the result 0 for ρZ . This probably is puzzling to the reader, but it is really quite handy. The reader should simply commit to memory the facts that scalars have no dimension and the empty vector has dimension 0. In the last entry of Table 2.1, Z is defined as the 1×3 array consisting of 1 2 3. This array appears to be identical to the vector $Y \leftarrow 1 \ 2 \ 3$, but APL makes a distinction between the two.

VARIABLE	ρZ
$Z \leftarrow 3$	-
$Z \leftarrow 2 \ 3 \ 5$	3
$Z \leftarrow (2,2) \rho \ 1 \ 2 \ 3 \ 4$	2 2
$Z \leftarrow \iota 0$	0
$Z \leftarrow (1,3) \rho \ 2 \ 3 \ 5$	1 3

Table 2.1: THE SHAPE OPERATOR

The last APL operators discussed in this section are the grade up and grade down functions. These two functions, denoted \uparrow and \downarrow ,* give the indices of the elements in a vector in order of magnitude: ascending order for grade up and descending order for grade down. This is very useful for data analysis since much of our work requires us to sort the data. These operations accomplish this easily. Here are some examples:

```

      A ← 9 12 6 4 -2 3 7 -5 11
      ⍷A
8 5 6 4 3 7 1 9 2
      ⍶A
2 9 1 7 3 4 6 5 8

```

In the grade up of A the first element, 8, says that the eighth value of A should be taken first; the second element, 5, says that the fifth element of A should be taken next, etc., to sort A in increasing order. Similarly for grade down to sort in decreasing order. The sorted arrays can be written as follows:

```

      A[⍷A]
-5 -2 3 4 6 7 9 11 12
      A[⍶A]
12 11 9 7 6 4 3 -2 -5

```

* These symbols are upper shift H and G, respectively, overstruck with upper shift M.

We have only scratched the surface of APL features useful for generating, modifying and storing data. The collection of operations discussed here will, nevertheless, enable a user to do most of the things that he requires to generate, correct, modify and store data. In the next section we discuss arithmetic operations in APL so that the user can perform calculations and transform data arrays.

C. Arithmetic Operations

In this section we discuss some of the basic arithmetic operations. We restrict attention to those operations most frequently used in data analysis.

The symbols used for the basic operations of adding, subtracting, multiplying, and dividing are the standard ones used in mathematics: (+, -, ×, ÷).^{*} However, in APL these operations are more powerful than their equivalents in most programming languages since the operations can be applied to entire vectors or matrices. When used with vectors or arrays, the operations are applied componentwise. This requires that the vectors or arrays have the same size. An exception is made for the case where one of the arguments is a scalar. When this happens APL adds,

^{*}The subtraction sign is upper shift +. Do not confuse this with the negative sign which is upper shift 2.

subtracts, multiplies or divides the scalar to each element of the vector or array. Of course, the operation can be applied to two or more scalars as with any other programming language. The examples below illustrate the basic operations:

```

      A ← 3  5  7  8      (CR)
      B ← -1 -2  1  2      (CR)
      A + B      (CR)
2  3  8  10
      A - B      (CR)
4  7  6  6
      A × B      (CR)
-3 -10  7  16
      A ÷ B      (CR)
-3 -2.5  7  4
      2 × A      (CR)
6  10  14  16
      3 + B      (CR)
2  1  4  5
      C ← (2,2)pA      (CR)
      C      (CR)
3  5
7  8

```

$$D \leftarrow (2,2) \rho B \quad (\text{CR})$$

$$D \quad (\text{CR})$$

$$\begin{matrix} \sim 1 & \sim 2 \end{matrix}$$

$$\begin{matrix} 1 & 2 \end{matrix}$$

$$C + B \quad (\text{CR})$$

$$\begin{matrix} 2 & 3 \end{matrix}$$

$$\begin{matrix} 8 & 10 \end{matrix}$$

$$C \div 2 \quad (\text{CR})$$

$$\begin{matrix} 1.5 & 2.5 \end{matrix}$$

$$\begin{matrix} 3.5 & 4.0 \end{matrix}$$

$$3 \times 4 + 5 \quad (\text{CR})$$

$$27$$

$$12 \div 3 \times 2 \quad (\text{CR})$$

$$2$$

The last two examples are included to re-emphasize the right-to-left order of operations in APL. Thus, 5 is added to 4 to give 9 which is then multiplied by 3 to yield 27. Similarly, in the last case, 2 is multiplied by 3 to give 6 which is then divided into 12 to yield 2. Unlike some other languages there is no hierarchy of operations other than right to left. Of course, parentheses may be used as in algebra to change the priority of operation.

$$(3 \times 4) + 5 \quad (\text{CR})$$

$$17$$

$$(12 \div 3) \times 2 \quad (\text{CR})$$

$$8$$

In addition to the four basic operations, data analysis frequently requires power, log, reciprocal, and exponential transformations. As with the four basic operations these can be applied to scalars, vectors, or matrices.

Table 2.2 presents the syntax for these operations.

Transformation	Syntax
X^a	$X * a$
$\ln X$	$\textcircled{O}X^1$
$1/X$	$\div X$
e^X	$*X$

Table 2.2. SYNTAX FOR DATA TRANSFORMATIONS

Here we give examples of these transformations:

```

X ← 1 2 3 4 5 (CR)
X * 0.5 (CR)
1 1.41421 1.73205 2 2.23607
X * 2 (CR)
1 4 9 16 25
X * ^-3 (CR)
1 0.125 0.037037 0.015625 0.008

```

¹The symbol for the logarithm is the upper shift O (oh) overstruck with the asterisk (upper shift P).

⊗X (CR)
 0 0.693147 1.098612 1.386294 1.609438

÷X (CR)
 1 0.5 0.333333 0.25 0.2

X * ÷⁻³ (CR)
 1 0.793701 0.693361 0.629961 0.584804

(Recall that APL operates from right to left so that the operator above raises each element in X to the $-1/3$ power.)

*X (CR)
 2.71828 7.389056 20.085537 54.598150 148.413159

The logarithm of a number N to an arbitrary base B can be determined by typing B ⊗ N; e.g.,

10 ⊗ 1000
 3

Three other useful arithmetic operations for scalars, vectors, and matrices are the ceiling \lceil , floor \lfloor , and absolute value $|$. All of these can be used with a single right hand argument or with both left and right arguments.

The examples below illustrate the use of these operators:

Monadic (one argument)

$\lceil 2.75 \ 6 \ 0.08 \ ^{-3.6}$
 3 6 1 ⁻³ (Gives smallest integer \geq argument.)

$\lfloor 2.75 \ 6 \ 0.08 \ ^{-3.6}$
 2 6 0 ⁻⁴ (Gives largest integer \leq argument.)

$| 2.75 \ 6 \ 0.08 \ ^{-3.6}$
 2.75 6 0.08 3.6 (Gives absolute value of each element.)

Dyadic (two arguments)

3 5 [2 8

3 8 (Gives the maximum of each component.)

3 5 | 2 8

2 5 (Gives the minimum of each component.)

3 | 7 6 8 -2

1 0 2 1 (Gives the remainder after dividing each element by 3.)*

Clearly, the floor and ceiling operators are useful for rounding values. One can select any number of significant digits. If one wants to round values to the nearest integer, one should type $\lfloor 0.5 + N$. If, say, four significant digits are wanted, one could type $(\lfloor 0.5 + N \times 1E4) \div 1E4$. The examples below illustrate the use of \lfloor for rounding:

$N \leftarrow 0.0835126 \quad 12.51877623 \quad 1.33333333 \quad 5.25$

$\lfloor .5 + N$

0 13 1 5

$(\lfloor .5 + N \times 1E4) \div 1E4$

0.0835 12.5188 1.3333 5.2500

Another operator, the reduction operator $/$, allows the operations discussed above to be applied to all the elements of a vector or to the rows or columns of a matrix. The syntax is

* All remainders of $K|N$ are expressed as positive integers in the set $0, 1, \dots, K-1$. If a remainder is negative, K is added to it.

f/A where f can be any of the arithmetic operations discussed above and A is a vector or a matrix. The result of this operation when applied to a vector A of size n is the scalar $A[1] f A[2] f \cdots f A[n]$. When applied to a matrix M having r rows and c columns, the result is the vector B of dimension r where $B[i] = A[i;1] f A[i;2] f \cdots f A[i;c]$ for $i = 1, 2, \dots, r$ (column reduction). The matrix M can be reduced over its rows by typing $f/[1]M$. The examples below illustrate the reduction operation:

```

+ /1 3 7 4
15      (the sum of all the elements)

x/1 3 7 4
84      (the product of all the elements)

[/1 3 7 4
7      (the largest element)

M ← (3,2)ρ3 1 2 6 5 4
M
3 1
2 6
5 4

-/M
2 -4 1      (differences of elements in rows 1, 2, and 3)

L/M
1 2 4      (minimum values in rows 1, 2, and 3)

```



```

+/[1]M
10 1    (sums of elements in columns 1 and 2)

[/[1]M
5 6     (maximum values in columns 1 and 2)

```

Finally, we describe the inner product operation and the matrix inverse. The inner product, like the reduction operation can be applied with any general APL operators. The syntax is $Af.gB$, where A and B are vectors or matrices (which must satisfy certain size restrictions) and f and g are any general APL operators. For vector arguments, A and B must be the same size, say n , and the result is given by $f/A[1]gB[1] \ A[2]gB[2] \ \dots \ A[n]gB[n]$. For matrix arguments the number of columns of A must be the same as the number of rows of B . The result is a matrix of size $n \times m$ where $\rho A = (n,k)$ and $\rho B = (k,m)$. The (i,j) th element of the result is $f/A[i;1]gB[1;j] \ A[i;2]gB[2;j] \ \dots \ A[i;k]gB[k;j]$. The reader should recognize the operation above as matrix multiplication when f is $+$ and g is \times . The examples below illustrate the inner product.

```

X ← 1 4 3 2
Y ← 2 3 1 2
X+.×Y
21 ((1×2) + (4×3) + (3×1) + (2×2))
X×.+Y
336 ((1+2) × (4+3) × (3+1) × (2+2))

```

A ← 3 2 ρ 1 6

A

1 2

3 4

5 6

B ← 2 3 ρ 3 5 2 3 1 1

B

3 5 2

3 1 1

A + . × B

7 7 4 (the matrix product of A and B)

21 19 10

33 31 16

The domino operator \boxtimes (type \square , backspace, \div) is used to solve for the matrix inverse of a nonsingular square matrix. The inverse of a nonsingular square matrix C is found by typing $\boxtimes C$. For example,

C ← 2 2 ρ 1 1 0 2

C

1 1

0 2

$\boxtimes C$

1 -0.5

0 0.5

C + . × $\boxtimes C$

1 0

0 1

D. Workspace Management

In the previous sections we have seen how data are input and variables are transformed. In this section we describe how the user manages his APL workspace so that he can save data for use from one session to the next; he can load public library workspaces; and he can use functions available in the public library workspaces. For information about writing functions the user should see references [2,3,4].

When a user types APL he is put into a clear APL workspace. In this clear workspace data can be created and any of the operations described in the earlier chapters can be performed. If the user wishes to maintain the data or results for future use he must save the workspace. This is done by typing

```
)SAVE WSPACE
```

where WSPACE is an arbitrary name (first letter alphabetic, eleven characters or less) that the user selects for the workspace. This private workspace can subsequently be loaded into the user's active workspace by typing

```
)LOAD WSPACE
```

This causes the active workspace to be cleared and a copy of the named workspace to be written into the active workspace. Alternatively, if the user does not want to clear out the contents of the active workspace to bring in the named workspace, he can type

`) COPY WSPACE`

This will simply augment the existing contents of the active workspace with the contents of workspace WSPACE. However, if an existing variable or function in the active workspace has the same name as a function or variable in workspace WSPACE the latter will replace the former.

Any modifications, additions, deletions, corrections, etc. that the user makes to a copy of a workspace will affect the active workspace, but will not affect the permanent copy of the workspace maintained on his private files unless the user saves the so modified active workspace. This can be accomplished by typing `)SAVE WSPACE` (or simply `)SAVE` if workspace WSPACE were LOADED).

The user can determine the name of the active workspace at any time by typing `)WSID` (for WorkSpace IDentification). Similarly, he can change the name of the active workspace by typing `)WSID NEWNAME`. A list of all the user's private APL workspaces is obtained by typing `)LIB`. Entire workspaces can be permanently destroyed by typing `)DROP WSPACE`. (Be careful with this one!)

Within an active workspace, a user can obtain a list of all of the functions contained in the workspace by typing `)FNS`. The functions will be listed alphabetically so that all functions from those beginning with a certain letter, LETTER, onward can be obtained by typing `)FNS LETTER`. Similarly, a list of

variables in the workspace can be obtained by typing `)VARS` or `)VARS LETTER`. Typeout can be terminated at any point by hitting the `BREAK` or the `ATTN` key. Variables and/or functions can be deleted from the active workspace by typing `)ERASE LIST` where `LIST` is a single function or variable or a list of functions and variables to be deleted. Names in the list should be separated by one or more blanks. Two other system commands allow the user some control over his APL environment. These commands control the number of digits printed out and the width of a typed line. They are `)DIGITS N` and `)WIDTH N`, where `N` is the number of digits to be printed or the desired line width, respectively.

In addition to his own private APL workspaces, the user also has access to all of the APL public libraries available at NPS. These libraries have numbers between 1 and 999 and are intended to hold workspaces of general interest. See [1] for a list of the public library numbers. The contents of a public library can be displayed using the `)LIB` command followed by the library number. For example,

)LIB 1

A-DISK	R/0	
IOFNS	17.25	7/09
PLOTFORM	17.25	7/09
NEWS	15.54	8/27
WSFNS	17.25	7/09
TEXTEDIT	17.26	7/09
FORMAT	17.26	7/09
CATALOG	15.17	7/14
MAILBOX	12.50	7/14
MULTIPLO	14.06	7/18
FILEFNS	17.18	8/14

The list contains the workspace name and the time and date that it was last modified.

A copy of a public library workspace can be put into the user's active workspace by typing)LOAD n WSNAME or)COPY n WSNAME where n is the library number and WSNAME the workspace name. (Recall the differences in LOAD and COPY; LOAD will first clear the contents of the active workspace.) Selected functions and variables from a workspace (either public or private) can be copied by typing

or)COPY n WSNAME OBJECT
)COPY n WSNAME GROUP

where OBJECT is a single variable or function that the user wants to copy into his active workspace and GROUP is a group of variables and/or functions that has previously been defined for WSNAME.*

When a public library workspace is brought into the user's active workspace, documentation can usually be obtained by typing the word DESCRIBE. It is somewhat standard procedure to document public library workspaces with a DESCRIBE variable giving general information and with each function in the workspace being documented by a "HOW" variable. Type the name of the function followed by HOW with no intervening spaces. For example, documentation for the function STEMLEAF in public library 2 OA3660 is obtained by typing STEMLEAFHOW. "HOW" variables generally describe the function syntax, parameters, input/output requirements, etc. Of course, the user can determine if a workspace contains this sort of internal documentation by typing)VARS.

When the user has completed his work session he can logout by typing)OFF. If he wants to get out of APL but not logout, he should type)OFF CMS. Then, he should switch back to the standard keyboard.

*The group structure for the OA3660 workspace is discussed in Chapter III.

E. Error Recovery

This section describes recovery procedures that the user can employ if he receives error messages during execution of functions contained in a workspace. Usually, error messages received when functions in public library workspaces are executed are the result of improper syntax or problems with the shapes of the arguments. As soon as an error is detected, execution of the function is suspended, the number of the line containing the error is typed, a caret is inserted at the position in the line that the error occurred, and an explanatory error message is printed out. When the function is suspended the values of all variables determined up to the point of suspension can be obtained by simply typing the variable name. All of this makes error discovery and correction quite simple. From the point of suspension the user can branch to any line of the function by typing `→ n`, where `n` is the desired line number. If `n` is omitted a branch is made outside of the function. Many types of calculations can be performed while a function is suspended, including the execution of other functions. However, the function will remain suspended until a branch is made as described above. Suspended functions tend to clutter up the user's workspace so that suspended functions should not be left pending. A list of all suspended functions can be displayed at any time by typing `)SI` (state indicator). If several functions are listed as being suspended, the user should type as many branch arrows (`→`), one per line, as there are asterisks displayed in the list.

Since the functions contained in the public library workspaces have been tested extensively, most of the errors that are encountered result from improper function syntax or the use of arguments that are not conformable or of improper size. The documentation contained in the next chapter shows the proper syntax for each function, and gives the requirements placed on the arguments.

III. THE OA3660 WORKSPACE

This chapter provides documentation for the public library workspace, 2 OA3660. General workspace documentation is contained in the DESCRIBE variable and in the lists of functions, variables, and groups. Short writeups are given for each function in the workspace via "HOW" variables, and examples of the use of each function are provided. Data are contained in the workspace to provide easy illustration of the functions. The data are described in the DATAMAP variable. Finally, the actual APL programs are displayed.

A. General Documentation

The OA3660 APL workspace is contained in public library 2. Therefore, the user must type)LOAD 2 OA3660 or)COPY 2 OA3660 to create a copy of the OA3660 workspace in his active workspace area. The functions and variables contained in OA3660 are displayed by typing)FNS and)VARS as shown below.

```
      )FNS
AND      ANOVA   A3R      A3RSR   BOXPLOT CHISQUARE   CODEARS
COMPARE  CONDENSE CONTINGENCY CORRELATION  FILL      FMT
INPUT    KS      LINE     LIT      LSLINE  MEDPOLISH  MSTATS  NUMSUM
ONEH     ONE3    PARTIAL  REGRESS  SCAT     SHOWRES  SPLIT   STATISTICS
STEMLEAF SUMSQ   TWICE    UTCOND
```

```

)VARS
AGES      ANDHOW  ANOVAHOW      A3RHOW  A3RSRHOW      BOXPLOTHOW
CHICKWTS  CHISQUAREHOW  COAL    CODERESHOW      COMPAREHOW
CONDENSEHOW  CONSUMPTION  CONTINGENCYHOW  CORRELATIONHOW  CR
CRIMES  DATAMAP  DEATHS  DEP      DEPTH  DESCRIBE      DISCOVERIES
DRAPER  EPSILON  GRADES1  GRADES2  HYDROPLANTS      INSECTS ASHOW
LANDAREAS  LENGTH  LINEHOW  LSLINEHOW      MEDPOLISHHOW      MISS
MORTALITY  MSTATSHOW  NDIVX   NDIVY   NGAP    NORMEFFECTS
NUM        NUMSUMHOW      ONEHHOW  ONE3HOW  OPTION  PARTIALHOW
PRECIPITATION  PRESRATING  REGRESSHOW      RESIDS  RIVERS
RSELECT  SCALE  SCATHOW  SHOWRESHOW      SPLITHOW      STATISTICSHOW
STEMLEAFHOW  TESTX   TESTY   TWICEHOW      USPOP  VOLCANO
WARPBREAKS  WID     WIDTH  AINTERCEPT  CA

```

Notice that there is a variable called DESCRIBE. This variable gives general documentation about the workspace.

```

)THE DESCRIBE VARIABLE PROVIDES GENERAL DOCUMENTATION
)FOR THE ENTIRE WORKSPACE.  SIMPLY TYPE 'DESCRIBE' TO
)OBTAIN THE INFORMATION.

```

DESCRIBE

0A3660-DATA ANALYSIS

DATE: JANUARY 1978

PROGRAMMER: F. RUSSELL RICHARDS

THIS WORKSPACE CONTAINS FUNCTIONS AND VARIABLES USEFUL FOR EXPLORATORY INTERACTIVE DATA ANALYSIS. FOR INFORMATION ON THE USE OF THE FUNCTIONS TYPE THE FUNCTION NAME FOLLOWED BY 'HOW'. FOR EXAMPLE, TYPE STEMLEAFHOW FOR DOCUMENTATION ON THE FUNCTION STEMLEAF. SOME FUNCTIONS ARE USED ONLY AS SUBPROGRAMS FOR OTHER FUNCTIONS AND WILL NOT HAVE A 'HOW' VARIABLE. FOR MORE DETAILED DOCUMENTATION SEE THE NPS TECHNICAL REPORT, A USER'S GUIDE TO THE 0A3660 APL WORKSPACE, BY F. RUSSELL RICHARDS OR DONALD A. MCNEIL'S BOOK, INTERACTIVE DATA ANALYSIS.

THE WORKSPACE ALSO CONTAINS VARIOUS SMALL DATA SETS FOR ILLUSTRATION OF THE FUNCTIONS. TYPE DATAMAP TO OBTAIN IDENTIFICATION OF THE DATA ARRAYS.

THE USER CAN SAVE SPACE IN HIS ACTIVE WORKSPACE BY COPYING ONLY THOSE FUNCTIONS AND VARIABLES NEEDED TO PERFORM THE NECESSARY TASKS. FUNCTIONS AND VARIABLES ARE CONVENIENTLY COLLECTED INTO GROUPS WHICH ARE IDENTIFIED IN THE '407' VARIABLES. FOR EXAMPLE, IF THE USER WANTS TO GENERATE STEMLEAF OR BOXPLOT DISPLAYS HE CAN COPY THE DISPLAY GROUP AS FOLLOWS:

```
)COPY 2 0A3660 DISPLAY
```

THE GROUP CONTAINS NOT ONLY THE FUNCTIONS BUT ALSO ALL REQUIRED PARAMETERS AND DOCUMENTATION.

There is also a variable named DATAMAP which describes the data contained in OA3660 and provides a reference to the data source. This is shown below.

THE VARIABLE DATAMAP PROVIDES INFORMATION ABOUT THE DATA SETS CONTAINED IN THE WORKSPACE. IT GIVES THE SOURCE OF EACH DATA SET AND THE SIZE OF THE DATA ARRAY. SIMPLY TYPE 'DATAMAP' FOR DATA DOCUMENTATION.

DATAMAP

THIS WORKSPACE CONTAINS SEVERAL DATA SETS THAT CAN BE USED TO ILLUSTRATE THE FUNCTIONS OR TO SERVE AS HOMEWORK PROBLEMS. THE ENTIRE COLLECTION OF DATA ARRAYS IS CONTAINED IN THE GROUP NAMED DATA. THE DATA CAME FROM THE FOLLOWING SOURCES:

(1) DRAPER, W.R. AND H. SMITH, APPLIED REGRESSION ANALYSIS, WILEY AND SONS.

(2) MCNEIL, D.R., INTERACTIVE DATA ANALYSIS, ADDISON-VESLEY.

(3) MILLER, I. AND J.E. FREUND, PROBABILITY AND STATISTICS FOR ENGINEERS, PRENTICE-HALL.

(4) RICHARDS, F.R., CLASSROOM GRADES.

(5) TUKEY, JOHN, EXPLORATORY DATA ANALYSIS, ADDISON-VESLEY.

THE TABLE BELOW LISTS EACH DATA ARRAY, THE DIMENSION, THE SOURCE (USING AUTHOR ABBREVIATIONS), AND PAGE OR CHAPTER NUMBERS.

DATA	DATA	SOURCE
AGES	42	MC(P.17)
CHICKWTS	14 6	MC(P.30)
COAL	49	TU
CONSUMPTION	5 5	MC(P.100)
CRIMES	50 4	MC(P.132)
DEATHS	5 4	MC(P.94)
DISCOVERIES	100	MC(P.121)
DRAPERX	13 4	DS(P.178)
DRAPERY	13	DS(P.178)
GRADES1	15 5	RI
GRADES2	15 5	RI
HYDROPLANTS	34	TU(CH.3)
INSECTS	12 6	MC(P.12,36)
LANDAREAS	48	MC(P.9)
MORTALITY	5 4	TU(CH.16)
PRECIPITATION	69	MC(P.3)
PRESKRATING	114	MC(P.126)
RIVERS	141	MC(P.14)
TESTX	10 2	MF(P.253)
TESTY	10	MF(P.253)
USPOP	19 2	TU(CH.8)
VOLCANO	219	TU
WARPBREAKS	9 6	MC(P.28)

Finally, notice that there are "HOW" variables for most of the functions contained in the workspace. (The only functions without "HOW" variables are utility functions which are used by other functions, but which are transparent to the user.) These variables provide documentation on the use of the functions. For example, documentation on the function SCAT is obtained by typing SCATHOW.

SCATHOW

SYNTAX: SCAT ARRAY

PARAMETERS:

- (1) WID- CONTROLS THE HORIZONTAL SIZE OF THE DISPLAY (DEFAULT=30 CHARACTERS).
- (2) DEP- CONTROLS THE VERTICAL SIZE OF THE DISPLAY (DEFAULT=15 LINES).
- (3) NDIVX, NDIVY- NUMBER OF UNITS ON X- AND Y-AXES, RESPECTIVELY (DEFAULT=4,4).

GROUP: RELATIONS, SMOOTH, COMPARISONS

DESCRIPTION: SCAT PRODUCES A SCATTER PLOT OF THE DATA CONTAINED IN ARRAY. THE ARGUMENT ARRAY CAN BE A VECTOR OR A MATRIX WITH AS MANY AS 9 COLUMNS. IF A VECTOR OF SIZE N, THOSE VALUES ARE PLOTTED VS. THE INTEGERS 1 TO N; IF A MATRIX, THE SECOND, THIRD, ETC. COLUMNS ARE PLOTTED VS. COLUMN 1 ON THE SAME AXES. DIVISIONS ON THE AXES OF THE PLOT ARE NOT EXPLICITLY PRINTED, EXCEPT AT THE EXTREMES OF THE PLOT. THE USER CAN CONTROL THE RESOLUTION OF THE PLOT BY MODIFYING THE PARAMETERS WID, DEP, NDIVX AND NDIVY. PRINTING TIME INCREASES DRAMATICALLY WITH RESOLUTION; THEREFORE, OTHER PLOT PROGRAMS SHOULD BE USED IF HIGH RESOLUTION IS DESIRED. FOR A SINGLE GROUP OF DATA, THE NUMBER N ($2 \leq N \leq 9$) WILL BE PRINTED IF N POINTS LIE CLOSE TOGETHER ON THE DISPLAY. FOR MULTIPLE PLOTS ON THE SAME DISPLAY, THE LETTER A REPRESENTS GROUP 1, B REPRESENTS GROUP 2, ETC. FOR 2 POINTS CLOSE TOGETHER THE LETTER WILL BE PRINTED WITH AN UNDERSCORE. THE DISPLAY CANNOT HANDLE 3 OR MORE POINTS CLOSE TOGETHER IF THERE ARE MULTIPLE PLOTS.

The format of SCATHOW is followed for every function. The function syntax, the user controlled parameters, the group (or groups) containing the functions, the subroutines used by the

function, and a brief description of the use of the function are displayed.

The functions and variables are grouped into APL groups for ease of handling and to conserve space. The group structure is described below.

```
)GRPS
COMPARISONS    DATA    DISPLAY    DOC    ESSENTIALS    GETT
RELATIONS      SMOOTH   STATS     TWOWAY
```

```
)GRP COMPARISONS
CONDENSE      CONDENSEHOW    NUM    MISS    UTCOND    FMT    SCAT
SCATHOW LINE  LINEHOW NDIVX   NDIVY   WID    DEP    DEPTH    GAP
COMPARE COMPAREHOW    FILL    NUMSUM    NUMSUMHOW    KSELECT AND
ANDHOW LSLINE LSLINEHOW
```

```
)GRP DATA
AGES    CHICKNTS    COAL    CONSUMPTION    CRIMES    DEATHS
DISCOVERIES    DRAPERX    DRAPERY    GRADES1    GRADES2    HYDROPLANTS
INSECTS LANDAREAS    MORTALITY    PRECIPITATION    PRESRATING
RIVERS    TESTX    TESTY    USPOP    WARPBREAKS    VOLCANO
```

```
)GRP DISPLAY
STEMLEAF      STEMLEAFHOW    WIDTH    SCALE    LIT    BOXPLOT
BOXPLOTHOW    LENGTH    FILL    NUMSUM    NUMSUMHOW    MISS    FIT
CH
```

```
)GRP DOC
ANDHOW ANOVHOW    A3RHOW    A3RSRHOW    BOXPLOTHOW
CHISQUAREHOW    CODERESHOW    COMPAREHOW    CONDENSEHOW
CONTINGENCYHOW    CORRELATIONHOW    DATAMAP    DESCRIBE    KSHOW
LINEHOW LSLINEHOW    MEDPOLISHHOW    MSTATHOW    NUMSUMHOW
ONEHOW ONE3HOW    PARTIALHOW    REGRESSHOW    SCATHOW SHOWRESHOW
SPLITHOW    STATISTICSHOW    STEMLEAFHOW    TWICEHOW
```

```

)GRP ESSENTIALS
AND      ANOVA      A3R      A3RSR      BOXPLOT  CHISQUARE      CODERES
COMPARE  CONDENSE      CONTINGENCY  CORRELATION  CR      DEP
DEPTH    EPSILON  FILL      FMT      INPUT    KS      LENGTH  LIVE      LIT
LSLINE   MEDPOLISH      MISS      NDIVX     NDIVY     NORMEFFECTS      191
NUNSUM   ONEH      ONE3      OPTION    REGRESS  RESIDS  SCALE    SCAT
SHOWRES  SPLIT    STEMLEAF      SUMSQ    TWICE    UTCOND  WID      17011
CR        MSTATS    PARTIAL STATISTICS      RSELECT  BINTERCEPT

```

```

)GRP GFIT
KS        KSHOW      CHISQUARE      CHISQUAREHOW

```

```

)GRP RELATIONS
LINE      LINEHOW  LSLINE  LSLINEHOW  REGRESS  REGRESSHOW  190
ANDHOW    RSELECT  SCAT    SCATHOW  WID      DEP      NDIVX     NDIVY
BINTERCEPT  FMT      CR

```

```

)GRP SMOOTH
SPLIT     ONE3      ONEH      A3R      A3RSR      TWICE    SCAT      SPLITHOW
A3RHOW    A3RSRHOW      ONE3HOW  ONEHHOW  TWICEHOW      SCATHOW  NDIVX
NDIVY     WID      DEP

```

```

)GRP STATS
CORRELATION  CR      FMT      MSTATS  PARTIAL STATISTICS      CR
CORRELATIONHOW  MSTATSHOW      PARTIALHOW      STATISTICSHOW

```

```

)GRP TOWAY
MEDPOLISH      MEDPOLISHHOW  LINE      LINEHOW  LSLINE  LSLINEHOW
RESIDS  EPSILON  NORMEFFECTS  SHOWRES  CODERES  RSELECT  SHOWRESHOW
CODERESHOW      AND      ANDHOW  CONTINGENCY  CONTINGENCYHOW  ANOVA
ANOVAHOW      OPTION  INPUT  SUMSQ

```

The user can conserve space in his active work area by selectively loading or copying only the required functions or groups. We have attempted to anticipate the type of analyses that the user will attempt, and the group structure has been selected with the objective to combine functions and parameters which naturally go hand-in-hand. A specific group can be selected by typing

)COPY 2 OA3660 GROUPNAME

where GROUPNAME is any one of the groups listed above. Two of the groups, ESSENTIALS and DOC, are especially useful. The ESSENTIALS group contains all of the functions and default values of all of the required parameters. Documentation and data are not included. The DOC group contains all of the documentation (DESCRIBE, DATAMAP, and "HOW" variables). The user may want to load the entire OA3660 workspace, check on some of the documentation, and then erase the group DOC to make room for user generated data and other functions or variables. The DOC group can be erased by typing)ERASE DOC. Similarly, any group can be erased by typing)ERASE GROUPNAME.

The following sections give brief descriptions (the writeups contained in the "HOW" variables) and examples of the functions contained in the OA3660 library. Each function description gives the syntax, a list of the parameters, a list of the groups which contain the function, a list of subroutines (other functions) used by the function, and a narrative about the use of the function. The narrative tells what the function does, describes any restrictions on the function arguments, and indicates how the function parameters affect the output. The order of presentation of the functions follows what appears to me to be a natural sequence of exposure to tools of data analysis.

The HOW variables are listed alphabetically in Appendix A, and the APL program listings are given in Appendix B.

B. Function Descriptions and Examples

```
      )GRP DISPLAY
STEMLEAF      STEMLEAFHOW      WIDTH  SCALE  LIT      BOXPLOT
BOXPLOTHOW    LENGTH  FILL      NUNSUM  NUNSUMHOW  MISS    FMT
CR
```

STEMLEAFHOW

SYNTAX: STEMLEAF VECTOR

GROUP: DISPLAY

PARAMETERS:

- (1) WIDTH- CONTROLS THE WIDTH (CHARACTERS PER LINE) OF THE DISPLAY (DEFAULT=70).
- (2) SCALE- VARIES THE DEPTH (STEM INTERVAL) OF THE DISPLAY IN UNITS OF 1, 2, OR 0.5 TIMES A POWER OF 10. SELECT AN INTEGER FROM 1 TO 3 (DEFAULT=1).

SUBPROGRAM: LIT

DESCRIPTION: STEMLEAF GENERATES A STEM AND LEAF DISPLAY OF A VECTOR OF OBSERVATIONS. THE FUNCTION AUTOMATICALLY SCALES THE DATA USING A SCALING ROUTINE BASED ON THE RANGE AND SIZE OF THE DATA VECTOR. BY VARYING THE SCALE PARAMETER THE USER CAN CHANGE THE SIZE OF THE STEM, BUT IT WILL NOT NECESSARILY BE SCALED BY THE AMOUNT SPECIFIED. TRY VARIOUS VALUES LIKE 1, 2, AND 3 TO SEE WHICH CHOICE YIELDS THE BEST DISPLAY. IF A LEAF CONTAINS TOO MANY CHARACTERS TO BE PRINTED ON A LINE, THE LINE WILL BE TRUNCATED AND THE NUMBER OF TRUNCATED CHARACTERS WILL BE WRITTEN AT THE END OF THE LINE. IF THE STEM INTERVAL IS TWO TIMES A POWER OF TEN, THE LEAVES MAY CONTAIN THE CHARACTERS A,B,C,..... IN ADDITION TO THE DIGITS 0,1,....,9. IN THIS CASE, THE CHARACTERS A TO C REPRESENT THE NUMBERS 10 TO 19.

SCALE

1

```

      STEHLER VOLCANO
00|25666799
01|0001366799
02|0011222444556667788999
03|01122445556667899
04|0111233333444578899999
05|0011222344556666677799
06|001144556666777889
07|00001112334555678889
08|122223335679
09|000123344556779
10|0112233445689
11|0112334669
12|11244456
13|03478
14|00
15|667
16|25
17|29
18|5
19|03379

```

SCALE+.5

```

      STEHLER VOLCANO
00|25666799AAAEDGGH**
02|00112224445566677889999ABBCCEEEEEFGGGHI**
04|0111233333444578899999AABBCCEEEFGGGGGGGHHH
06|001144556666777889AAAABBBCCDEEEFGHIII-
08|122223335679AAAABCCDEEEFGHH*
10|0112233445689ABBCDEEGG*
12|11244456ADENI
14|00GGH
16|25C-
18|5ADDH*

```

A CHANGING SCALE TO .5 REDUCED THE NUMBER OF STEPS.
 A INCREASING SCALE WILL ALWAYS RESULT IN MORE STEPS ON
 A THE SAME NUMBER.

BOXPLOTCHW
 SYNTAX: BOXPLOT VECTOR
 PARAMETER:

LENGTH- CONTROLS THE HORIZONTAL SIZE OF THE DISPLAY
 (DEFAULT=50 CHARACTERS).

GROUP: DISPLAY

SUBPROGRAM: FILL

DESCRIPTION: BOXPLOT GENERATES A BOX PLOT DISPLAY FOR A VECTOR OF DATA. A RECTANGULAR BOX WITH ENDS CORRESPONDING TO LOWER AND UPPER QUANTILES IS PRESENTED WITH THE MEDIAN MARKED WITH AN ASTERISK. WHISKERS ARE DRAWN ON EACH SIDE OF THE BOX WITH CROSSES MARKING THE LOWEST AND HIGHEST DATA VALUES WITHIN AN INTERQUARTILE DISTANCE OF THE QUANTILES. DATA VALUES OUTSIDE THE CROSSES (OUTLIERS) ARE MARKED WITH CIRCLES AND THOSE MORE THAN 1.5 INTERQUARTILE DISTANCES GET HEAVY CIRCLES.

```

2                                199
XXXXXXXXXX [-----] XXXXXXXXXXXXXXXX OOOO C @@@@
              *
              [-----]
              22
  
```

AS WE CAN SQUEEZE THE BOXPLOT IN OR STRETCH IT OUT BY
 A CHANGING LENGTH.

LENGTH=30

```

2                                199
XXXX [-----] XXXXX OOOOO@@@
      *
      [-----]
      32  32
  
```

LENGTH=70
 BOXPLOT VOLCANO

```

2                                199
XXXXXXXXXXXX [-----] XXXXXXXXXXXXXXXX O O O O * @@@@
              *
              [-----]
              3 2
  
```

NUMSUMHOW
 SYNTAX: NUMSUM VECTOR OR NUMSUM ARRAY
 PARAMETER: MISS= NUMBER USED TO INDICATE MISSING VALUE IN THE DATA
 ARRAY (DEFAULT=-99999).
 GROUP: COMPARISONS
 SUBPROGRAM: FMT
 DESCRIPTION: NUMSUM OPERATES ON EITHER A VECTOR OR AN ARRAY BUT,
 IN ALL CASES, THE DATA ARE TREATED AS A SINGLE BATCH. NUMSUM
 PRODUCES A NUMERICAL SUMMARY WHICH GIVES THE SAMPLE SIZE (AFTER
 DELETION OF MISSING VALUES), THE EIGHTHS, THE EXTREMES, THE
 SPREADS, AND THE MIDPOINTS IN TABULAR FORM. THIS SUMMARY IS
 USEFUL FOR TESTING THE SYMMETRY OF A DATA SET, AND TO EVALUATE
 THE EFFECTIVENESS OF DATA TRANSFORMATIONS IN PRODUCING SYMMETRY.

NUMSUM VOLCANO

NUMERICAL SUMMARY

SAMPLE SIZE = 219

MIDPTS	LOQ/8/MIN	MEDIAN	UPQ/8/MAX	SPREADS
65.00		65.00		
66.50	37.00		96.00	59.00
72.50	24.00		121.00	97.00
100.50	2.00		199.00	197.00

A THE STEMLEAF, THE BOXPLOT, AND THE NUMERICAL SUMMARY ALL
 A SUGGEST THAT THE VOLCANO DISTRIBUTION IS POSITIVELY
 A SKEWED. TRY A LOG TRANSFORM TO SEE IF THAT WILL MAKE
 A THE DISTRIBUTION MORE NEARLY SYMMETRIC.

NUMSUM • VOLCANO

NUMERICAL SUMMARY

SAMPLE SIZE = 219

MIDPTS	LOQ/8/MIN	MEDIAN	UPQ/8/MAX	SPREADS
4.17		4.17		
4.09	3.61		4.56	0.95
3.99	3.18		4.80	1.62
2.99	0.69		5.29	4.60

A THERE IS NOW A DECREASING TREND IN THE MIDPTS COLUMN.
 A THAT SUGGESTS THAT THE LOG IS TOO EXTREME A TRANSFORMATION.
 A TRY SQUARE ROOT.

NUMSUM VOLCANO*.5

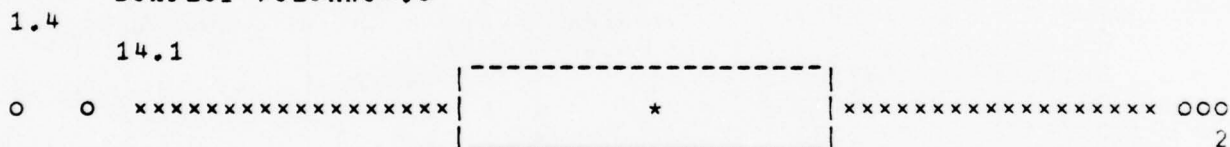
NUMERICAL SUMMARY

SAMPLE SIZE = 219

MIDPTS	LOQ/8/MIN	MEDIAN	UPQ/8/MAX	SPREADS
8.06		8.06		
7.94	6.08		9.80	3.72
7.95	4.90		11.00	6.10
7.76	1.41		14.11	12.69

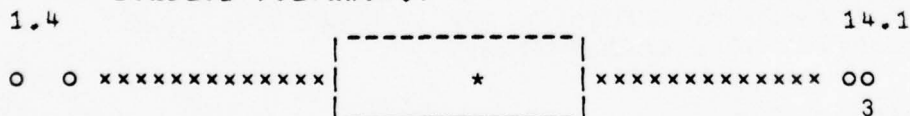
A THAT LOOKS A LOT BETTER. LET US OBTAIN THE BOXPLOT
 A AND STEMLEAF PLOT OF THE SQUARE ROOT TRANSFORMED DATA.

BOXPLOT VOLCANO*.5



LENGTH=50

BOXPLOT VOLCANO*.5



A THE BOXPLOT LOOKS A LOT MORE SYMMETRIC. NOW FOR THE
 A STEMLEAF PLOT:

STEMLEAF VOLCANO*.5

01		4
02		24446
03		0022236
04		001445566777999
05		00111223344456677889999
06		00012223444566666668999
07		000001111222333445555555777788
08		001111112222234444445556777788889
09		011111112334555566677778889
10		00011112223445556667889
11		000111224567788
12		55578
13		146899
14		01

```

)GRP COMPARISONS
CONDENSE          CONDENSEHOW  NUM      MISS      UTCOND  FMT      SCAT
SCATHOW LINE      LINEHOW NDIVX  NDIVY     WID      DEP      DEPTH  WGAP
COMPARE COMPAREHOW      FILL    NUMSUM  NUMSUMHOW      RSELECT AND
ANDHOW  LSLINE  LSLINEHOW

```

ANDHOW

SYNTAX: X AND Y

GROUPS: RELATIONS, TWOWAY, COMPARISONS

DESCRIPTION: THE FUNCTION 'AND' IS USED TO CREATE A NEW DATA ARRAY CONSISTING OF X AUGMENTED BY Y AS ADDITIONAL COLUMNS. X AND Y CAN BE SCALARS, VECTORS, OR MATRICES BUT CANNOT BOTH BE SCALARS. IF AN ARGUMENT IS A SCALAR, A COLUMN IS GENERATED EACH ELEMENT OF WHICH IS THE SCALAR. IF THE ARGUMENTS ARE VECTORS OR MATRICES, THEY MUST BE CONFORMABLE.

A THE FUNCTION 'AND' IS USEFUL FOR DATA ENTRY AND MANIPULATION.

```

X+3 4 5
Y+9 10 12
X AND Y
3 9
4 10
5 12
Z+3 2p16
X AND Z
3 1 2
4 3 4
5 5 6
50 AND Z
50 1 2
50 3 4
50 5 6
50 AND X
50 3
50 4
50 5

```

A EXCEPT WHEN ONE ARGUMENT IS A SCALAR, THE ARGUMENTS OF A AND MUST BE CONFORMABLE IN SIZE.

```

W+4 6 8 10
X AND W
ARGUMENTS OF AND ARE NOT CONFORMABLE.
W AND Z
ARGUMENTS OF AND ARE NOT CONFORMABLE.

```

COMPAREHOW

SYNTAX: COMPARE MATRIX

PARAMETERS:

- (1) DEPTH- VERTICAL HEIGHT OF DISPLAY (DEFAULT=20 LINES).
- (2) MISS- NUMBER USED TO INDICATE MISSING VALUES (DEFAULT = -99999)
- (3) NGAP- NUMBER OF HORIZONTAL SPACES BETWEEN THE BOXPLOT DISPLAYS (DEFAULT=3).

GROUP: COMPARISONS

SUBPROGRAM: FILL

DESCRIPTION: COMPARE OPERATES ON AN N BY K MATRIX TO PRODUCE K VERTICAL BOX PLOTS PLACED NEXT TO EACH OTHER TO ALLOW VISUAL COMPARISON OF THE CENTERS, SPREADS, AND OUTLIERS OF THE BATCHES (COLUMNS) OF THE MATRIX. THE USER MUST FILL UP THE MATRIX SO THAT THERE ARE N OBSERVATIONS FOR EACH BATCH. THE PARAMETER, MISS, SHOULD BE USED TO FILL IN MISSING VALUES. THE USER SHOULD ASSURE THAT MISS IS DIFFERENT FROM ALL VALID ENTRIES IN THE DATA ARRAY.

DEPTH

20

MISS

-99999

NGAP

3

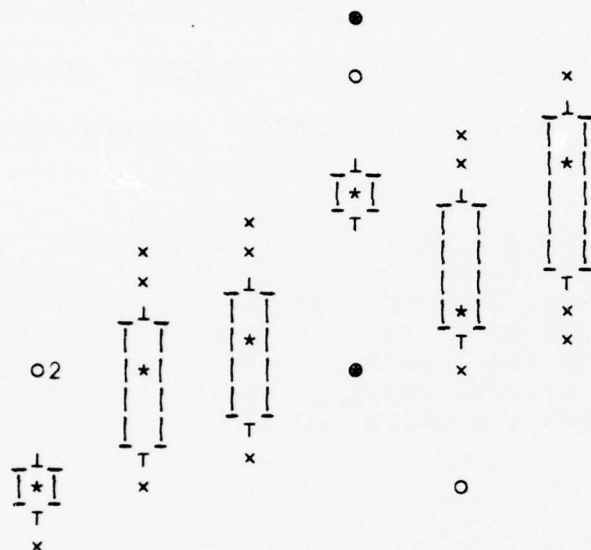
A WE WILL DEMONSTRATE COMPARE WITH THE CHICKWT DATA.
A SINCE WE USED 0 TO INDICATE MISSING VALUES IN THAT
A DATA, WE MUST FIRST ALTER THE PARAMETER MISS.

MISS=0

COMPARE CHICKWTS

423

108



- A THE DIAGRAM REVEALS RATHER LARGE DIFFERENCES IN THE
- A MIDSPREADS OF THE CHICKWT DATA. WE SHOULD TRANSFORM
- A THE DATA TO TRY TO ACHIEVE HOMOGENIETY IN THE SPREADS.
- A FIRST LETS GET A NUMERICAL SUMMARY OF THE DIFFERENT COLUMNS.

CONDENSEHOW

SYNTAX: CONDENSE MATRIX OR R+CONDENSE MATRIX

PARAMETERS:

- (1) NUM= CONTROLS WHAT STATISTICS ARE INCLUDED IN THE SUMMARY. NUM∈[1,2,5,7] (DEFAULT=2).
 NUM=1 GIVES MEDIAN OF EACH COLUMN OF THE MATRIX
 NUM=2 GIVES MEDIAN AND INTERQUARTILE RANGES
 NUM=5 GIVES MIN, QUANTILES, MAX, AND SIZE
 NUM=7 GIVES MIN, EIGHTHS, MAX AND SIZE
- (2) MISS= NUMBER USED TO CODE MISSING VALUES (DEFAULT=99999).

GROUP: COMPARISONS

SUBPROGRAMS: UTCOND, FMT

DESCRIPTION: CONDENSE GENERATES SUMMARY STATISTICS FOR EACH GROUP OF DATA REPRESENTED BY THE COLUMNS OF THE ARGUMENT MATRIX. SINCE THE ARGUMENT IS A MATRIX, THE USER MUST FILL UP THE MISSING VALUES IN THE MATRIX WHENEVER THE GROUPS (COLUMNS) HAVE DIFFERENT NUMBERS OF OBSERVATIONS. THE PARAMETER, MISS, SHOULD BE USED TO FILL THE MATRIX SINCE THE FUNCTION WILL RECOGNIZE THOSE VALUES AS MISSING DATA AND WILL IGNORE THEM IN ALL CALCULATIONS. THE RESULT FROM CONDENSE CAN BE USED AS THE ARGUMENT OF OTHER FUNCTIONS SUCH AS SCAT OR LINE IF AND ONLY IF NUM=2.

2 NUM
MISS
0

CONDENSE CHICKWTS
151.5 43
221 83.5
248 72
328 33
263 83
342 102

A THE ABOVE DATA ARE THE MEDIANS AND MIDSPREADS FOR EACH
A OF THE 6 BATCHES (COLUMNS) OF CHICKWT DATA. ZERO WAS
A ENTERED INTO THE PARAMETER MISS EARLIER. THE MIDSPREADS
A SHOW QUITE A DISPERSION, FROM AS SMALL AS 151.5 TO AS LARGE
A AS 342.

A WE CAN GET MORE INFORMATION ABOUT THE INDIVIDUAL COLUMNS
A OF CHICKWTS BY CHANGING NUM AS FOLLOWS:

NUM+5
CONDENSE CHICKWTS

MIN	LOQ	MEDIAN	UPQ	MAX	SIZE
108.00	136.00	151.50	179.00	227.00	10.00
141.00	175.00	221.00	258.50	309.00	12.00
158.00	199.00	248.00	271.00	329.00	14.00
226.00	307.50	328.00	340.50	423.00	12.00
153.00	242.00	263.00	325.00	380.00	11.00
216.00	271.50	342.00	373.50	404.00	12.00

A IN SEEKING OUT AN APPROPRIATE TRANSFORMATION TO EVEN OUT
A THE MIDSPREADS, IT IS OFTEN USEFUL TO PLOT THE LOG OF THE
A MIDSPREADS VS. THE LOG OF THE MEDIANS.
A IN MANY CASES WE NEED TO LOOK AT A SCATTER PLOT OF OUR
A DATA, OR AT SCATTER PLOTS OF FUNCTIONS OF OUR DATA.
A LET US INTRODUCE A FUNCTION SCAT THAT WILL PROVIDE THE
A NEEDED SCATTER PLOT.

SCATHOW

SYNTAX: SCAT ARRAY

PARAMETERS:

- (1) WID= CONTROLS THE HORIZONTAL SIZE OF THE DISPLAY (DEFAULT=30 CHARACTERS).
- (2) DEP= CONTROLS THE VERTICAL SIZE OF THE DISPLAY (DEFAULT=15 LINES).
- (3) NDIVX, NDIVY= NUMBER OF UNITS ON X- AND Y-AXES, RESPECTIVELY (DEFAULT=4,4).

GROUP: RELATIONS, SMOOTH, COMPARISONS

DESCRIPTION: SCAT PRODUCES A SCATTER PLOT OF THE DATA CONTAINED IN ARRAY. THE ARGUMENT ARRAY CAN BE A VECTOR OR A MATRIX WITH AS MANY AS 9 COLUMNS. IF A VECTOR OF SIZE N, THOSE VALUES ARE PLOTTED VS. THE INTEGERS 1 TO N; IF A MATRIX, THE SECOND, THIRD, ETC. COLUMNS ARE PLOTTED VS. COLUMN 1 ON THE SAME AXES. DIVISIONS ON THE AXES OF THE PLOT ARE NOT EXPLICITLY PRINTED, EXCEPT AT THE EXTREMES OF THE PLOT. THE USER CAN CONTROL THE RESOLUTION OF THE PLOT BY MODIFYING THE PARAMETERS WID, DEP, NDIVX AND NDIVY. PRINTING TIME INCREASES DRAMATICALLY WITH RESOLUTION; THEREFORE, OTHER PLOT PROGRAMS SHOULD BE USED IF HIGH RESOLUTION IS DESIRED. FOR A SINGLE GROUP OF DATA, THE NUMBER N ($2 \leq N \leq 9$) WILL BE PRINTED IF N POINTS LIE CLOSE TOGETHER ON THE DISPLAY. FOR MULTIPLE PLOTS ON THE SAME DISPLAY, THE LETTER A REPRESENTS GROUP 1, B REPRESENTS GROUP 2, ETC. FOR 2 POINTS CLOSE TOGETHER THE LETTER WILL BE PRINTED WITH AN UNDERSCORE. THE DISPLAY CANNOT HANDLE 3 OR MORE POINTS CLOSE TOGETHER IF THERE ARE MULTIPLE PLOTS.

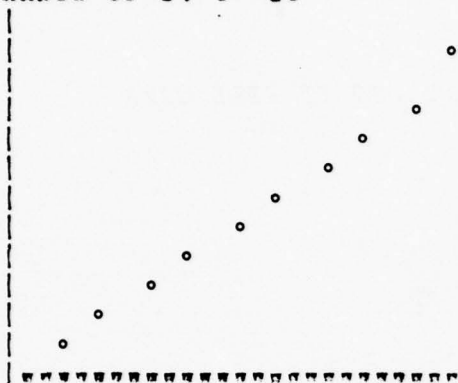
WID
30
DEP
15
NDIVX
4
NDIVY
4

X+110
Y1+2 3 5 6 8 10 11 13 15 18
Y2+20 20 17 24 21 26 31 30 37 33

```

SCAT X AND Y1
RANGE OF X: 0 10
RANGE OF Y: 0 20

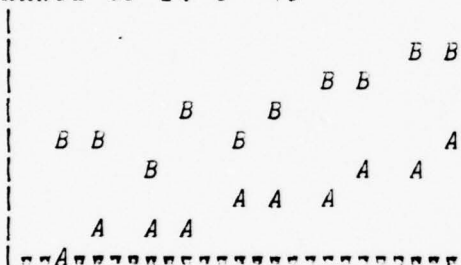
```



```

SCAT X AND Y1 AND Y2
RANGE OF X: 0 10
RANGE OF Y: 0 40

```

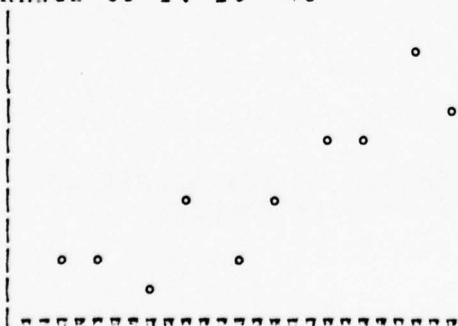


A IF THE ARGUMENT OF SCAT IS A VECTOR, SCAT WILL CREATE
 A ABSCISSA VALUES TO BE THE POSITIVE INTEGERS 1 TO pARGUMENT.

```

SCAT Y2
RANGE OF X: 0 10
RANGE OF Y: 15 40

```

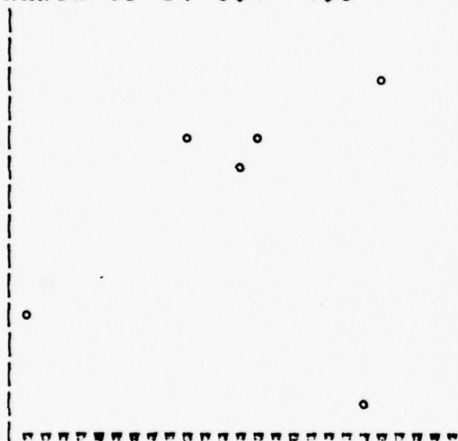


A BACK TO THE PROBLEM OF NONHOMOGENEOUS MIDSPREADS IN THE
 A CHICKWT DATA. LET US LOOK AT A SCATTER PLOT OF THE
 A LOG MIDSPREADS V. LOG MEDIANS.

A FIRST WE RESET PARAMETER NUM TO 2, SO IT WILL GIVE
 A MEDIANS AND MIDSPREADS.

NUM+2

SCAT CONDENSE CHICKWTS
 RANGE OF X: 5 6
 RANGE OF Y: 3.4 4.8



A THE SCATTER PLOT SUGGESTS A RELATIONSHIP BETWEEN MIDSPREADS
 A AND MEDIANS. PERHAPS SOME ROOT OF THE DATA WOULD MAKE
 A THE MIDSPREADS MORE HOMOGENEOUS. TRY SQUARE ROOT.

CONDENSE CHICKWTS*.5

12.30368569	1.717184371
14.86363273	2.851055496
15.74801575	2.355341653
18.11001266	0.9195158937
16.21727474	2.471407191
18.4912651	2.852064545

A THAT IS BETTER (RATIO OF MAX MIDSPREAD TO MIN MIDSPREAD
 A IS ABOUT 1.5. MORE WORK NEEDS TO BE DONE THOUGH.

)GRP RELATIONS
 LINE LINEHOW LSLINE LSLINEHOW REGRESS REGRESSHOW AND
 ANDHOW RSELECT SCAT SCATHOW WID DEP NDIVX NDIVY
 ΔINTERCEPT FMT CR

LINEHOW

SYNTAX: LINE ARRAY CR Z+LINE ARRAY

PARAMETER:

RSELECT- SELECTS THE OUTPUT MATRIX OF RESIDUALS.

RSELECT=1 GIVES ABSCISSA VS. RESIDUALS (DEFAULT=1).

RSELECT≠1 GIVES FITTED VALUES VS. RESIDUALS

GROUP: RELATIONS

DESCRIPTION: LINE FITS A STRAIGHT LINE TO A SET OF (X,Y) POINTS BY DIVIDING THE POINTS INTO 3 REGIONS AND USING THE MEDIANS OF THE X AND Y VALUES IN THE OUTER REGIONS TO DETERMINE THE SLOPE. THE INTERCEPT IS THE MEDIAN OF THE DIFFERENCES $Y - \text{SLOPE} \times X$. LINE ALSO COMPUTES THE RESIDUALS AND GIVES AN N BY 2 MATRIX CONTAINING EITHER THE X-VALUES VS. RESIDUALS OR THE FITTED VALUES VS. RESIDUALS DEPENDING ON THE PARAMETER, RSELECT. IF THE USER DOES NOT WANT RESIDUALS TYPED OUT, HE SHOULD USE THE SYNTAX:

Z+LINE ARRAY

THE ARGUMENT ARRAY CAN BE EITHER A VECTOR OR A 2-COLUMN MATRIX. IF THE ARGUMENT IS A VECTOR OF SIZE N, THE X-VARIABLE IS CONSTRUCTED TO BE THE FIRST N POSITIVE INTEGERS. IF THE ARGUMENT IS A 2-COLUMN MATRIX, THE FIRST COLUMN IS TAKEN TO BE THE SET OF X-VALUES AND THE SECOND COLUMN THE SET OF Y-VALUES. LINE CAN BE USED IN CONJUNCTION WITH THE FUNCTION SCAT TO PRODUCE THE SLOPE, THE INTERCEPT, AND A PLOT OF THE RESIDUALS. TO DO THIS, ENTER:

SCAT LINE ARRAY

X=2 3 4 3 5 6 7 6 7 5 9 8 10 11 12 14
 pX

16

ARRAY←X AND Y

LINE ARRAY

SLOPE: 1.375 Y=INTERCEPT: 0.0625

2	0.1875
3	-0.1875
4	-1.5625
3	0.8125
5	1.0625
6	-2.3125
7	-5.6875
6	-0.3125
7	-0.6875
5	3.0625
9	-0.4375
8	3.9375
10	2.1875
11	-1.1875
12	1.4375
14	0.6875

RSELECT←0

LINE ARRAY

SLOPE: 1.375 Y=INTERCEPT: 0.0625

2.8125	0.1875
4.1875	-0.1875
5.5625	-1.5625
4.1875	0.8125
6.9375	1.0625
8.3125	-2.3125
9.6875	-5.6875
8.3125	-0.3125
9.6875	-0.6875
6.9375	3.0625
12.4375	-0.4375
11.0625	3.9375
13.8125	2.1875
15.1875	-1.1875
16.5625	1.4375
19.3125	0.6875

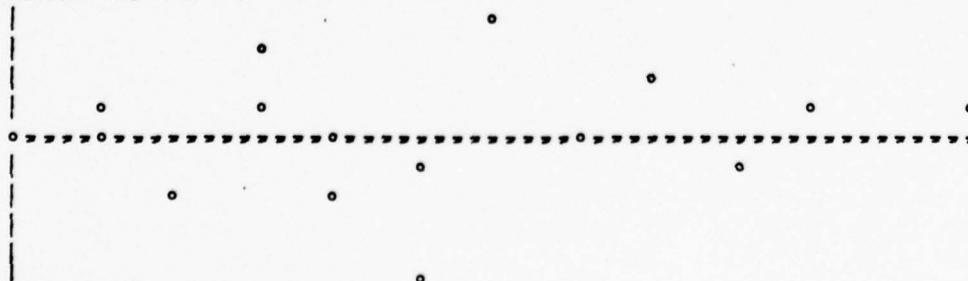
RSELECT←1

SCAT LINE ARRAY

SLOPE: 1.375 Y=INTERCEPT: 0.0625

RANGE OF X: 2 14

RANGE OF Y: -6 4



Y
3 4 4 5 8 6 4 8 9 10 12 15 16 14 13 20
pY

16

LINE Y

SLOPE: 1.090909091 Y-INTERCEPT: 0.6818181818

1	1.227272727
2	1.136363636
3	0.045454545
4	-0.045454545
5	1.863636364
6	-1.227272727
7	-4.318181818
8	-1.409090909
9	-1.5
10	-1.590909091
11	-0.6818181818
12	1.227272727
13	1.136363636
14	-1.954545455
15	0.954545455
16	1.863636364

X←116

ARRAY←X AND Y

LINE ARRAY

SLOPE: 1.090909091 Y-INTERCEPT: 0.6818181818

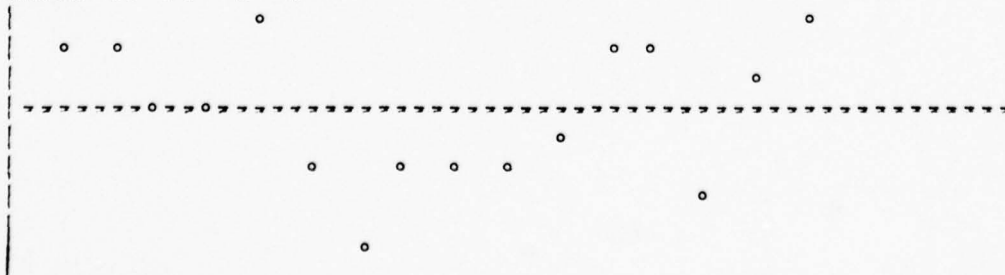
1	1.227272727
2	1.136363636
3	0.045454545
4	-0.045454545
5	1.863636364
6	-1.227272727
7	-4.318181818
8	-1.409090909
9	-1.5
10	-1.590909091
11	-0.6818181818
12	1.227272727
13	1.136363636
14	-1.954545455
15	0.954545455
16	1.863636364

SCAT LINE Y

SLOPE: 1.090909091 Y-INTERCEPT: 0.6818181818

RANGE OF X: 0 20

RANGE OF Y: -6 2



```

RSELECT+1
Z+LINE Y2
SLOPE: 1.857142857 Y-INTERCEPT: 17.42857143

```

```

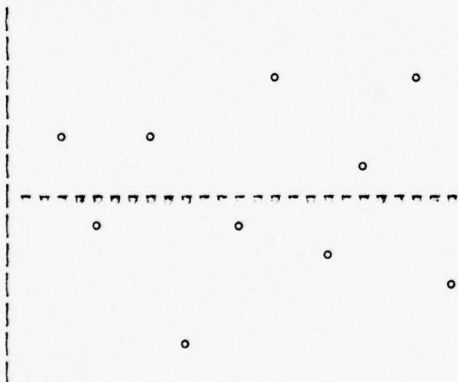
LSLINEHOW
SYNTAX: LSLINE ARRAY      OR      Z+LSLINE ARRAY
GROUP: RELATIONS
DESCRIPTION: LSLINE DETERMINES THE LEAST SQUARES SOLUTION,
Y=YI+SLOPE*X. THE ARGUMENT ARRAY CAN BE A VECTOR OR A 2-COLUMN
MATRIX. IF THE ARGUMENT IS A VECTOR, THE X-VALUES ARE TAKEN TO
BE THE INTEGERS 1 TO N WHERE N=ARRAY. IF THE ARGUMENT IS A
2-COLUMN MATRIX, COLUMN 1 CONSISTS OF THE X-VALUES AND COLUMN 2
THE Y-VALUES. RESIDUALS ARE DETERMINED AND STORED IN A MATRIX
WHOSE FIRST COLUMN IS THE SET OF X-VALUES AND WHOSE SECOND COLUMN
IS THE SET OF RESIDUALS. IF THE USER WANTS TO SUPPRESS PRINTOUT
OF THE RESIDUALS, HE MUST ASSIGN THE RESULTS OF LSLINE TO A
VARIABLE; I.E., TYPE Z+LSLINE ARRAY.

```

```

SCAT LSLINE X AND Y2
SLOPE: 1.933333333 Y-INTERCEPT: 16.86666667
RANGE OF X: 0 10
RANGE OF Y: -4 4

```



```

AIF THE USER WANTS TO REGRESS A RESPONSE VARIABLE ON
A A SET OF CARRIERS, HE CAN USE THE PROGRAM REGRESS.
A THIS LEAST SQUARES MULTIPLE REGRESSION PROGRAM IS
A VERY POWERFUL AND FLEXIBLE.

```

REGRESSHOW

SYNTAX: Z←Y REGRESS X

PARAMETER:

ΔINTERCEPT. DETERMINES WHETHER OR NOT AN INTERCEPT TERM IS TO BE INCLUDED. ΔINTERCEPT=1 GIVES AN INTERCEPT TERM, AND ΔINTERCEPT=0 GIVES NO INTERCEPT. (DEFAULT IS 1.)

GROUP: RELATIONS

SUBPROGRAMS: FMT AND SCAT

DESCRIPTION: REGRESS DOES A MULTIPLE REGRESSION ANALYSIS RELATING THE DEPENDENT VARIABLE Y TO A SET OF CARRIERS X. THE LEFT ARGUMENT Y IS A VECTOR OF SIZE N. THE RIGHT ARGUMENT X IS AN N BY K MATRIX CONSISTING OF N OBSERVATIONS ON EACH OF K VARIABLES OR A VECTOR OF SIZE N IF K=1. OUTPUT CONSISTS OF AN ANOVA TABLE, R-SQUARE, STD. ERROR, REGRESSION COEFFICIENTS (THE FIRST COEFFICIENT IS THE CONSTANT TERM IF ΔINTERCEPT=1.), T-STATISTICS, VARIANCE-COVARIANCE MATRIX, DURBIN-WATSON STATISTIC, AND A VECTOR OF PREDICTED Y VALUES AND RESIDUALS. THERE IS AN OPTION THAT ALLOWS THE USER TO INPUT A VECTOR OF X VALUES AND USE THE REGRESSION EQUATION TO FORECAST Y VALUES. THE USER CAN ALSO OBTAIN A SCATTER PLOT OF THE RESIDUALS. WHEN EXECUTION TERMINATES, THE PREDICTED Y VALUES AND THE RESIDUALS RESIDE IN THE N BY 2 MATRIX Z.

ΔINTERCEPT

1

ρDRAPERX

13 4

ρLETS REGRESS A RESPONSE VARIABLE DRAPERY ON THE FOUR CARRIERS IN THE MATRIX DRAPERX.

Z+DRAPERY REGRESS DRAPERX

ANOVA

SOURCE	DF	SUM SQUARES	MEAN SQUARE	F-RATIO
REGRESSION	4	2.6679E3	6.6697E2	1.1148E2
RESIDUAL	8	4.7864E1	5.9830E0	
TOTAL	12	2.7158E3		

R SQUARE: 0.9823756203

STD ERROR: 2.446007963

COEFFICIENTS T STATISTICS

62.4054	0.8906
1.5511	2.0827
0.5102	0.7049
0.1019	0.135
-0.1441	-0.2032

DO YOU WANT A PRINTOUT OF THE VARIANCE-COVARIANCE MATRIX?

Y

VARIANCE-COVARIANCE MATRIX:

4.9099E+3	5.0507E+1	5.0603E+1	5.1660E+1	4.9597E+1
5.0507E+1	5.5468E-1	5.1266E-1	5.5425E-1	5.0529E-1
5.0603E+1	5.1266E-1	5.2387E-1	5.2570E-1	5.1213E-1
5.1660E+1	5.5425E-1	5.2570E-1	5.6959E-1	5.1688E-1
4.9597E+1	5.0529E-1	5.1213E-1	5.1688E-1	5.0275E-1

DURBIN-WATSON: 2.052596933

DO YOU WANT TO FORECAST A VALUE FOR Y?

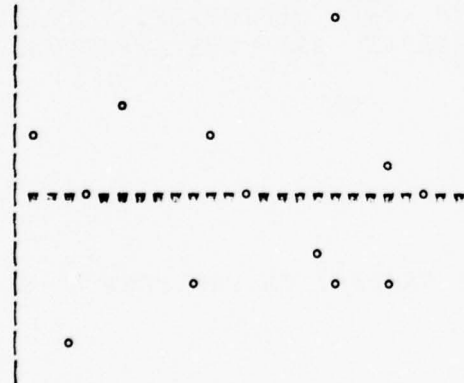
20 30 15 40

DO YOU WANT TO SCAT RESIDUALS VS. PREDICTED Y?

Y

RANGE OF X: 70 120

RANGE OF Y: -4 4



```

)GRP SMOOTH
SPLIT  ONE3  ONEH  A3R  A3RSR  TWICE  SCAT  SPLITHOW
A3RHOW  A3RSRHOW  ONE3HOW  ONEHHOW  TWICEHOW  SCATHOW  NDIVX
NDIVY  WID  DEP

```

```

ONE3HOW
SYNTAX:  ONE3 VECTOR
GROUP:  SMOOTH
DESCRIPTION:  ONE3 SMOOTHS A VECTOR OF DATA USING ONE PASS OF
RUNNING MEDIANS OF 3 WITH TUKEY'S END-POINT RULE. WHEN SMOOTHING
(X,Y) PAIRS, THE USER SHOULD USE AS THE ARGUMENT VECTOR THE Y
VALUES ORDERED ACCORDING TO THE MAGNITUDE OF THE X VALUES:

```

VECTOR+Y[AX]

IT IS ASSUMED THAT THE X VALUES ARE EQUISPACED. REPEATED
SMOOTHING BY MEDIANS OF 3 CAN BE ACCOMPLISHED WITH A3R. OUTPUT
CONSISTS OF THE SMOOTHED SEQUENCE. A PLOT OF THE SMOOTHED
SEQUENCE IS GIVEN BY TYPING:

SCAT ONE3 VECTOR

```

      X
3  2  3  7  4  3  2  15  4  5  3  6

```

```

ONE3 X
2  3  3  4  4  3  3  4  5  4  5  3

```

```

SPLITHOW
SYNTAX:  SPLIT A3R VECTOR
GROUP:  SMOOTH
DESCRIPTION:  SPLIT DOES ONE PASS AT DIVIDING MESAS (PAIRS OF
ADJACENT POINTS WITH A COMMON VALUE WHICH IS A LOCAL MAX OR MIN)
USING TUKEY'S END-POINT RULE. IT IS USED IN CONJUNCTION WITH
A3R.

```

```

SPLIT ONE3 X
2  3  3  3  3  4  3  4  5  4  5  3

```

```

A3RHOW
SYNTAX:  A3R VECTOR
GROUP:  SMOOTH
SUBPROGRAM:  ONE3
DESCRIPTION:  A3R DOES REPEATED SMOOTHINGS OF RUNNING MEDIANS OF 3
UNTIL THERE ARE NO CHANGES IN THE SMOOTHED SEQUENCE FROM ONE
ITERATION TO THE NEXT. SEE ONE3HOW.

```


A3R X
 3 3 3 4 4 3 3 4 4 4 4 4

A3RSRHOW
 SYNTAX: A3RSR VECTOR
 GROUP: SMOOTH
 SUBPROGRAMS: A3R, ONE3, SPLIT
 DESCRIPTION: A3RSR DOES REPEATED SMOOTHINGS BY RUNNING MEDIANS
 OF 3 FOLLOWED BY SPLITTING MESAS AND REPEATING UNTIL CONVERGENCE.
 SEE A3R AND SPLIT.

A3RSR X
 3 3 3 3 3 4 4 4 4 4 4 4

ONEHHOW
 SYNTAX: ONEH VECTOR
 GROUP: SMOOTH
 DESCRIPTION: ONEH DOES A SINGLE HANNING OF A SEQUENCE OF DATA.
 IT IS USED IN CONJUNCTION WITH THE OTHER SMOOTHING FUNCTIONS AS A
 FINAL TOUCH-UP TO A SEQUENCE OF SMOOTHED DATA. THE I-TH RESPONSE
 IS REPLACED BY $0.25 \times Y[I-1] + 0.5 \times Y[I] + 0.25 \times Y[I+1]$. SEE A3R AND
 A3RSR.

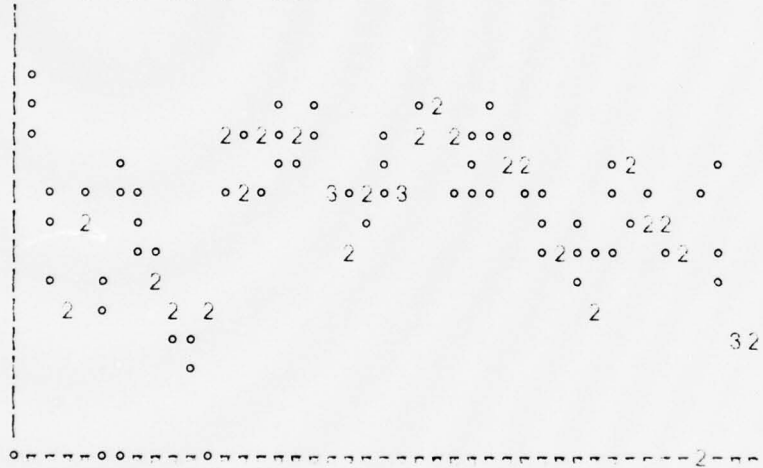
ONEH X
 3 2.5 3.75 5.25 4.5 3 5.5 9 7 4.25 4.25 6

TWICEHOW
 SYNTAX: TWICE VECTOR
 GROUP: SMOOTH
 SUBPROGRAMS: ONE3, SPLIT, A3RSR
 DESCRIPTION: TWICE SMOOTHS A SEQUENCE OF DATA, THEN SMOOTHS THE
 RESIDUALS AND ADDS THE SMOOTHED RESIDUALS BACK TO THE SMOOTHED
 DATA TO OBTAIN THE FINAL SMOOTH. THE SMOOTHING IS DONE USING
 A3RSR.

TWICE X

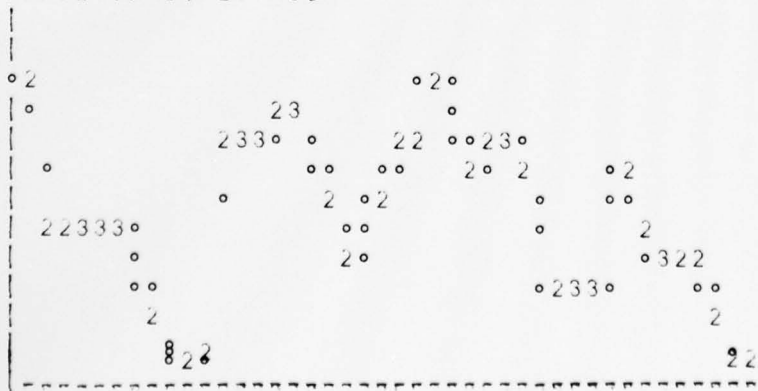
ANON LET US LOOK AT SOME PLOTS WITH SOME MORE INTERESTING
A DATA THAT SHOWS THE EFFECTS OF SMOOTHING.
WE WILL USE THE PRESRATING DATA.

WID+45
DEP+20
SCAT PRESRATING

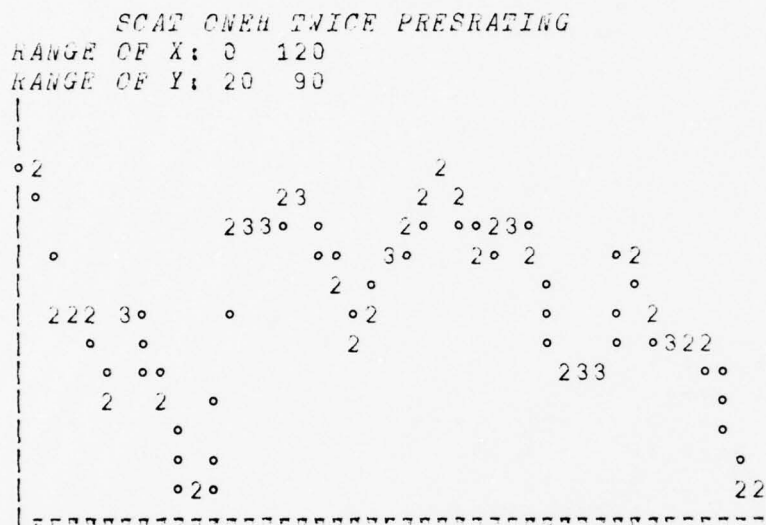


IT IS HARD TO TELL MUCH FROM THIS. LETS SEE IF SHOOTING
A HELPS ANY.

SCAT A3RSH PRESKATING



THAT IS A LOT SMOOTHER. THE PEAKS AND VALLEYS ARE A LOT
 CLEARER NOW. AS A FINAL TOUCHUP, LETS USE TWICE TO
 SMOOTH THE RESIDUALS AND ADD BACK TO THE SMOOTHED SEQUENCE,
 AND THEN USE ONEH AS A FINAL POLISH.



```

)GRP TWOWAY
MEDPOLISH      MEDPOLISHHOW  LINE   LINEHOW  LSLINE  LSLINEHOW
RESIDS  EPSILON NORMEFFECTS  SHOWRES CODERES RSELECT SHOWRESHOW
CODERESHOW    AND      ANDHOW CONTINGENCY  CONTINGENCYHOW ANOVA
ANOVAHOW      OPTION  INPUT  SUMSQ

```

```

MEDPOLISHHOW
SYNTAX: MEDPOLISH MATRIX      OR      Z+MEDPOLISH MATRIX
PARAMETERS:

```

- (1) RESIDS- CONTROLS OUTPUT OF RESIDUALS. IF 1, RESIDUALS ARE PRODUCED IN A TWO-WAY TABLE; IF 2, COMPARISON VALUES, $(RE \cdot CE) \cdot TV$ AND RESIDUALS ARE PRODUCED IN A TWO COLUMN MATRIX (DEFAULT=1).
- (2) EPSILON- PROPORTION BY WHICH THE SUM OF THE ABSOLUTE VALUES OF THE RESIDUALS MUST BE REDUCED AT EACH ITERATION TO CONTINUE POLISHING (DEFAULT=0.01).
- (3) NORMEFFECTS- DETERMINES WHETHER OR NOT NORMALIZED EFFECTS ARE OUTPUT. THE DEFAULT OF 0 SUPPRESSES OUTPUT OF NORMALIZED VALUES. NORMEFFECTS=1 CAUSES NORMALIZED VALUES TO BE PRINTED.

GROUP: TWOWAY

DESCRIPTION: MEDPOLISH ITERATIVELY SWEEPS OUT MEDIANS FROM THE ROWS AND COLUMNS OF A TWO-WAY TABLE TO YIELD THE MODEL:

$$OBS = MEDIAN + ROW\ EFFECT + COLUMN\ EFFECT + RESIDUAL$$

IT YIELDS RESIDUALS FOR TESTING THE ADEQUACY OF THE MODEL (SEE SHOWRES AND CODERES). THE FUNCTION CONTINUES POLISHING UNTIL THE STOPPING RULE CONTROLLED BY EPSILON IS ACTIVATED. THE OVERALL MEDIAN, THE ROW- AND COLUMN-EFFECTS, AND THE SUM OF ABSOLUTE VALUES OF THE RESIDUALS ARE GIVEN. THE USER CAN SUPPRESS RESIDUAL PRINTOUT BY ASSIGNING THE FUNCTION TO A VARIABLE, I.E., BY WRITING:

Z+ MEDPOLISH MATRIX

```

RESIDS
1
EPSILON
0.01
NORMEFFECTS
0

```

DEATHS

11.7	8.7	15.4	8.4
18.1	11.7	24.3	13.6
26.9	20.3	37	19.3
41	30.9	54.6	35.1
66	54.3	71.1	50

Z+MEDPOLISH DEATHS

315.4
40.3
38.25
37.575
TYPICAL VALUE: 25.6
ROW EFFECTS: -15.3625 -10.1 -1.4 -12.6375 32.875
COLUMN EFFECTS: 2.7 -3.9 12.625 -3.1375

THE RESIDUALS ARE STORED IN Z. THE FIRST VALUES
PRINTED OUT ARE THE SUMS OF THE ABSOLUTE VALUES
OF THE RESIDUALS. ITERATION CEASES WHEN TWO
CONSECUTIVE VALUES ARE WITHIN EPSILON.

2			
-1.2375	2.3625	-7.4625	1.3
-0.1	0.1	-3.825	-1.2375
0	0	0.175	-1.7625
0.0625	-3.4375	3.7375	0
4.825	-0.275	0	-5.3375

+ / + / | Z
37.2375

SHOWRESHOW
SYNTAX: SHOWRES MATRIX OR SHOWRES MEDPOLISH MATRIX
SUBPROGRAM: AND
GROUP: TOWAY
DESCRIPTION: SHOWRES AIDS IN THE ANALYSIS OF RESIDUALS FROM
MEDIAN POLISHING A TWO-WAY TABLE. THE USER SHOULD SET RESIDS=1
IN MEDPOLISH IF SHOWRES IS TO BE USED. THE OUTPUT OF SHOWRES IS
A MATRIX CONSISTING OF THE SYMBOLS COO* \times X DEPENDING ON THE SIZES
OF THE RESIDUALS. LET R REPRESENT THE RESIDUAL, M THE MIDSPREAD
OF ALL RESIDUALS, L THE LOWER QUANTILE, AND U THE UPPER QUANTILE.
THE CODES ARE PRINTED AS FOLLOWS:

O: $R < L - 1.5M$
O: $L - 1.5M \leq R \leq L + M$
O: $L + M \leq R < U + M$
X: $U + M \leq R \leq U + 1.5M$
X: $R > U + 1.5M$

SHOWRES Z

```

o o o o
o o o o
o o o o
o o x o
X o o o

```

CODERESHOW

SYNTAX: CODERES MATRIX OR CODESRES MEDPOLISH MATRIX

SUBPROGRAM: AND

GROUP: TOWAY

DESCRIPTION: CODERES PRODUCES A DIAGNOSTIC ARRAY FOR ANALYZING THE RESIDUALS FROM MEDIAN POLISHING OF A TWO-WAY TABLE. THE USER SHOULD SET RESIDS=1 IN MEDPOLISH WHEN CODERES IS USED. THE OUTPUT MATRIX CONSISTS OF THE SYMBOLS -o+ DEPENDING ON THE SIZE OF THE RESIDUALS.

```

-: VALUE BELOW LOWER QUANTILE
o: VALUE BETWEEN QUANTILES
+: VALUE LARGER THAN UPPER QUANTILES

```

CODERES Z

```

o + - +
o o - +
o o o -
o - + o
+ o o -

```

A LETS SEE WHAT HAPPENS WHEN RESIDS=2.

RESIDS+2

Z+MEDPOLISH DEATHS

315.4

40.3

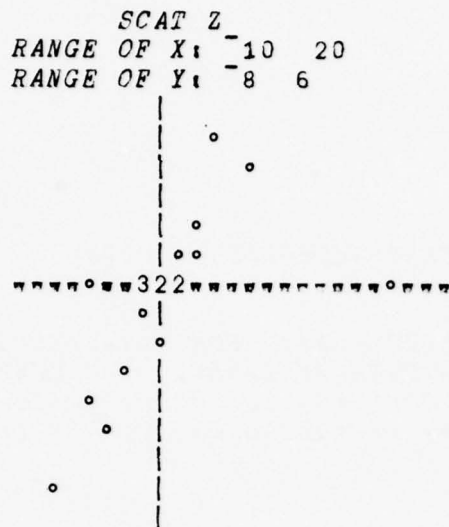
38.25

37.575

TYPICAL VALUE: 25.6

ROW EFFECTS: -15.3625 -10.1 -1.4 12.6375 32.875

COLUMN EFFECTS: 2.7 -3.9 12.625 -3.1375



A THERE SEEMS TO BE A RELATIONSHIP BETWEEN THE RESIDUALS
 A AND THE COMPARISON VALUES, (RE*.XCE)*TV.

A+LINE Z
 SLOPE: 0.6216761995 Y-INTERCEPT: -0.1348011389

ANOVAHOW
 SYNTAX: ANOVA
 PARAMETER:

OPTION= SELECTS THE TYPE OF ANOVA PERFORMED. OPTION=1
 GIVES A ONE-WAY ANOVA AND OPTION=2 GIVES A TWO-WAY
 ANOVA (TREATMENTS WITH BLOCKING). (DEFAULT=1)

GROUP: TOWAY

SUBPROGRAMS: INPUT, SUMSQ, FMT

DESCRIPTION: ANOVA DOES A ONE-WAY OR A TWO-WAY ANALYSIS OF
 VARIANCE DEPENDING ON THE VALUE OF THE OPTION PARAMETER. ANOVA
 WILL INTERACT WITH THE USER TO OBTAIN THE REQUIRED INFORMATION
 AND DATA. WHEN ENTERING DATA, SEPARATE THE DATA POINTS WITH AT
 LEAST ONE BLANK AND USE NO OTHER DELIMITERS. OUTPUT CONSISTS OF
 AN ANOVA TABLE, ESTIMATES OF THE OVERALL MEAN, AND ESTIMATES OF
 THE TREATMENT (AND, IF OPTION=2, BLOCK) EFFECTS.

OPTION

1

ANOVA

ENTER NUMBER OF TREATMENTS.

□:

4

ENTER VECTOR OF NUMBER OF OBS. FOR TREATMENTS 1 TO 4

□:

3 3 3 3

ENTER 3 OBSERVATIONS FOR TREATMENT 1

□:

45 46 51

ENTER 3 OBSERVATIONS FOR TREATMENT 2

□:

42 44 50

ENTER 3 OBSERVATIONS FOR TREATMENT 3

□:

36 41 48

ENTER 3 OBSERVATIONS FOR TREATMENT 4

□:

49 47 54

ANOVA TABLE

SOURCE	DF	SS	MS	F
TREATMENT	3	110.92	36.97	1.92
ERROR	8	154.00	19.25	
TOTAL	11	264.92		

R-SQUARE = 0.419

OVERALL MEAN = 46.08

TREATMENT EFFECTS 1.25 -0.75 -4.42 3.92

A LET US TAKE THE SAME DATA AND DO A TWO WAY ANOVA.

OPTION+2

ANOVA
 ENTER NUMBER OF TREATMENTS.
☐ 4
 ENTER VECTOR OF NUMBER OF OBS. FOR TREATMENTS 1 TO 4
☐ 3 3 3 3
 ENTER 3 OBSERVATIONS FOR TREATMENT 1
☐ 45 46 51
 ENTER 3 OBSERVATIONS FOR TREATMENT 2
☐ 42 44 50
 ENTER 3 OBSERVATIONS FOR TREATMENT 3
☐ 36 41 48
 ENTER 3 OBSERVATIONS FOR TREATMENT 4
☐ 40 47 54

ANOVA TABLE				
SOURCE	DF	SS	MS	F
TREATMENT	3	60.67	20.22	5.56
BLOCKS	2	204.17	102.08	28.05
ERROR	6	21.83	3.64	
TOTAL	11	286.67		

R-SQUARE = 0.924
 OVERALL MEAN = 45.33
 TREATMENT EFFECTS 2.00 0.00 -3.67 1.67
 BLOCK EFFECTS -4.58 -0.83 5.42

CONTINGENCYHOW
SYNTAX: CONTINGENCY MATRIX

GROUP: TOWAY

DESCRIPTION: CONTINGENCY TAKES A TWO-WAY TABLE AND PERFORMS A
TEST OF INDEPENDENCE OF ROWS AND COLUMNS. THE CHI-SQUARE
STATISTIC AND ITS DEGREES OF FREEDOM ARE OUTPUT.

CONTINGENCY DEATHS
CHI-SQUARE VALUE= 2.920832627 DF= 12

- A COMPARE THE ABOVE VALUE WITH THOSE IN CHI-SQUARE
- A TABLES FOR 12 DF TO DETERMINE IF THE HYPOTHESIS
- A STATING THAT THE ROWS AND COLUMNS ARE INDEPENDENT
- A CAN BE REJECTED AT THE SELECTED SIGNIFICANCE LEVEL.

)GRP STATS
 CORRELATION CR FMT MSTATS PARTIAL STATISTICS CH
 CORRELATIONHOW MSTATSHOW PARTIALHOW STATISTICSHOW

STATISTICSHOW
 SYNTAX: STATISTICS VECTOR
 GROUP: STATS
 SUBPROGRAM: FMT
 DESCRIPTION: STATISTICS DETERMINES THE MEAN, VARIANCE, STANDARD
 DEVIATION, COEFFICIENT OF VARIATION, LOWER AND UPPER QUANTILES,
 MEDIAN, TRIMEAN, MIDMEAN, MIDRANGE, RANGE, MEAN ABSOLUTE
 DEVIATION, INTERQUARTILE RANGE, SKEWNESS, AND KURTOSIS FOR THE
 DATA IN VECTOR.

STATISTICS VOLCANO
 MEAN: 70.24657534
 VARIANCE: 1850.562775
 STD. DEV.: 43.01816796
 COEFF. OF VARIATION: 0.6123881165
 LOWER QUANTILE: 37
 UPPER QUANTILE: 96
 MEDIAN: 65
 TRIMEAN: 65.75
 MIDMEAN: 64.47747748
 RANGE: 197
 MIDRANGE: 100.5
 MEAN ABSOLUTE DEVIATION: 33.68493151
 INTERQUARTILE RANGE: 59
 COEFF. OF SKEWNESS: 0.8325767059
 COEFF. OF KURTOSIS: 0.4281831164

CORRELATIONHOW
 SYNTAX: R+CORRELATION W
 GROUP: STATS
 SUBPROGRAM: FMT
 DESCRIPTION: CORRELATION DETERMINES THE SIMPLE PEARSON PRODUCT
 MOMENT CORRELATIONS BETWEEN EACH PAIR OF VARIABLES REPRESENTED BY
 THE C COLUMNS OF W. THE OUTPUT IS A C BY C CORRELATION MATRIX.

CORRELATION GRADES1						
	1.00	2.00	3.00	4.00	5.00	
1.00	1.00	0.34	-0.50	0.25	0.64	
2.00	0.34	1.00	-0.07	0.54	0.68	
3.00	0.50	-0.07	1.00	0.52	0.64	
4.00	0.25	0.54	0.52	1.00	0.87	
5.00	0.64	0.68	0.64	0.87	1.00	

A IN THE CORRELATION OUTPUT, THE TOP ROW AND THE FIRST
 A COLUMN ARE LABELS FOR THE CORRELATIONS. FOR EXAMPLE,
 A THE CORRELATION BETWEEN THE SECOND AND FOURTH COLUMNS
 A OF GRADES1 IS 0.54.

pGRADES1

15 5

MSTATSHOW

SYNTAX: MSTATS W

GROUP: STATS

SUBPROGRAM: FMT

DESCRIPTION: MSTATS DETERMINES CORRELATIONS, MEANS, STANDARD
 DEVIATIONS, LOWER QUANTILES, MEDIANS, AND UPPER QUANTILES FOR THE
 COLUMNS OF THE ARGUMENT MATRIX W. THE CORRELATIONS ARE DISPLAYED
 IN AN N BY N MATRIX, WHERE N IS THE NUMBER OF COLUMNS OF W.

THE OUTPUT FROM MSTATS IS MUCH LIKE THAT OF CORRELATION

MSTATS GRADES1						
	1.00	2.00	3.00	4.00	5.00	
1.00	1.00	0.34	-0.50	0.25	0.64	
2.00	0.34	1.00	-0.07	0.54	0.68	
3.00	0.50	-0.07	1.00	0.52	0.64	
4.00	0.25	0.54	0.52	1.00	0.87	
5.00	0.64	0.68	0.64	0.87	1.00	
MEANS	0.86	0.81	0.88	0.81	0.84	
STD.DEV	0.09	0.11	0.10	0.10	0.07	
L. QRTLS	0.82	0.69	0.82	0.74	0.78	
MEDIANS	0.85	0.83	0.88	0.79	0.83	
U. QRTLS	0.94	0.92	0.98	0.91	0.90	

PARTIALHON
 SYNTAX: PARTIAL W
 GROUP: STATS
 SUBPROGRAMS: CORRELATION, FMT
 DESCRIPTION: PARTIAL DETERMINES PARTIAL CORRELATIONS OF ANY SPECIFIED ORDER FOR THE COLUMNS OF THE MATRIX W. THE USER WILL BE ASKED FOR THE COLUMNS TO BE PARTIALED OUT. THE USER'S RESPONSES MUST BE INTEGERS BETWEEN 1 AND C, WHERE C IS THE NUMBER OF COLUMNS OF W.

PARTIAL GRADES1
 ENTER COLUMNS OF MATRIX TO BE PARTIALED OUT.

1:

	1				
		1.00	-2.00	3.00	4.00
1.00		1.00	-0.29	0.50	0.64
2.00		-0.29	1.00	0.47	0.49
3.00		0.50	0.47	1.00	0.95
4.00		0.64	0.49	0.95	1.00

A BE CAREFUL WITH INTERPRETING THE LABELS IN THE OUTPUT
 A OF PARTIAL. ABOVE, LABEL 1 DOES NOT MEAN COLUMN ONE,
 A BUT THE FIRST COLUMN REMAINING AFTER THE SELECTED VARIABLES
 A HAVE BEEN PARTIALED OUT. SIMILARLY, 2 REFERS TO THE
 A SECOND REMAINING COLUMN.

PARTIAL GRADES1
 ENTER COLUMNS OF MATRIX TO BE PARTIALED OUT.

1:

	2	3		
		1.00	-2.00	3.00
1.00		1.00	-0.45	0.23
2.00		-0.45	1.00	0.77
3.00		0.23	0.77	1.00

A THE ABOVE EXAMPLE SHOWS THAT MORE THAN ONE VARIABLE
 A CAN BE PARTIALED OUT. LABEL 2 ABOVE REFERS TO THE
 A SECOND COLUMN REMAINING AFTER THE SELECTED VARIABLES
 A HAVE BEEN PARTIALED OUT. IN THIS EXAMPLE 2 REFERS TO
 A COLUMN 4.

KS)GRP GFIT CHISQUARE CHISQUAREHOW
 KSHOW

CHISQUAREHOW

SYNTAX: T CHISQUARE X

GROUP: GFIT

DESCRIPTION: CHISQUARE COMPARES A THEORETICAL DISCRETE PROBABILITY MASS FUNCTION T HAVING N POSSIBLE VALUES WITH AN EMPIRICAL FREQUENCY FUNCTION X OVER THE SAME N VALUES TO TEST HOW WELL THE PROBABILITIES T FIT THE DATA X. THE CHI-SQUARE GOODNESS-OF-FIT STATISTIC IS PRINTED OUT WITH THE NUMBER OF DEGREES OF FREEDOM.

A THE THEORETICAL PROBABILITY MASS FUNCTION T AND THE
A EMPIRICAL FREQUENCY FUNCTION X MUST BE OF THE SAME
A SIZE; I.E., $p_T = p_X$. THE ACTUAL OBSERVED VALUES ARE
A IMPLICIT IN THE POSITION OF THE VECTORS.

T←.1 .2 .3 .2 .1
X← 5 20 38 32 12

A THE VALUES IN T REPRESENT PROBABILITIES. THE VALUES IN
A X REPRESENT OBSERVED FREQUENCIES. THE ACTUAL OBSERVATIONS
A ARE NOT STATED EXPLICITLY, BUT IT IS UNDERSTOOD THAT
A THE POSITIONS OF X AND T REFER TO THE SAME VALUES. FOR
A EXAMPLE, IN THE CASE ABOVE, THE FIRST POSITION MAY REFER
A TO THE VALUE 1, THE SECOND 0, ETC.

T CHISQUARE X
CHI-SQUARE VALUE = 9.620872274 DF = 4

A TO DETERMINE IF ONE SHOULD REJECT THE HYPOTHESIS THAT
A THE DATA WERE DRAWN FROM A POPULATION WITH PROBABILITY
A MASS FUNCTION T, COMPARE THE VALUE ABOVE WITH THOSE
A IN CHI-SQUARE TABLES WITH 4 DF.

KSHOW
 SYNTAX: F KS G
 GROUP: GFIT
 DESCRIPTION: KS DOES A KOLMOGOROV-SMIRNOV GOODNESS OF FIT TEST
 OF THE EQUALITY OF TWO DISCRETE EMPIRICAL PROBABILITY
 DISTRIBUTIONS. THE ARGUMENTS F AND G ARE EACH VECTORS OF
 OBSERVATIONS. THEY NEED NOT BE SORTED, NOR MUST THEY BE THE SAME
 SIZE. OUTPUT OF KS ARE THE VALUES:
 MAX $F(X)-G(X)$ AND MIN $F(X)-G(X)$

F+3 2 5 9 10 16 3 5 7 19 20 12 6 3 8 8 9 7 7 7 7 2 10 7

G+4 2 10 20 18 3 6 7 4 2 1 10 8 6 3 31 7

F KS G
 MAX $F(X)-G(X)$ = 0.1274509804
 MIN $F(X)-G(X)$ = -0.3137254902

A TO DETERMINE IF THE TWO SAMPLES OF DATA WERE TAKEN FROM
 A THE SAME POPULATION (OR POPULATIONS WITH THE SAME CDF'S)
 A COMPARE THE MAX AND MIN VALUES WITH THE DELTA STATISTICS
 A IN TABLES FOR THE TWO SAMPLE KS TEST.

APPENDIX A

ANDHOW

SYNTAX: X AND Y

GROUPS: RELATIONS, TWOWAY, COMPARISONS

DESCRIPTION: THE FUNCTION 'AND' IS USED TO CREATE A NEW DATA ARRAY CONSISTING OF X AUGMENTED BY Y AS ADDITIONAL COLUMNS. X AND Y CAN BE SCALARS, VECTORS, OR MATRICES BUT CANNOT BOTH BE SCALARS. IF AN ARGUMENT IS A SCALAR, A COLUMN IS GENERATED EACH ELEMENT OF WHICH IS THE SCALAR. IF THE ARGUMENTS ARE VECTORS OR MATRICES, THEY MUST BE CONFORMABLE.

ANOVAHOW

SYNTAX: ANOVA

PARAMETER:

OPTION- SELECTS THE TYPE OF ANOVA PERFORMED. OPTION=1 GIVES A ONE-WAY ANOVA AND OPTION=2 GIVES A TWO-WAY ANOVA (TREATMENTS WITH BLOCKING). (DEFAULT=1)

GROUP: TWOWAY

SUBPROGRAMS: INPUT, SUMSQ, FMT

DESCRIPTION: ANOVA DOES A ONE-WAY OR A TWO-WAY ANALYSIS OF VARIANCE DEPENDING ON THE VALUE OF THE OPTION PARAMETER. ANOVA WILL INTERACT WITH THE USER TO OBTAIN THE REQUIRED INFORMATION AND DATA. WHEN ENTERING DATA, SEPARATE THE DATA POINTS WITH AT LEAST ONE BLANK AND USE NO OTHER DELIMITERS. OUTPUT CONSISTS OF AN ANOVA TABLE, ESTIMATES OF THE OVERALL MEAN, AND ESTIMATES OF THE TREATMENT (AND, IF OPTION=2, BLOCK) EFFECTS.

A3RHOW

SYNTAX: A3R VECTOR

GROUP: SMOOTH

SUBPROGRAM: ONE3

DESCRIPTION: A3R DOES REPEATED SMOOTHINGS OF RUNNING MEDIANS OF 3 UNTIL THERE ARE NO CHANGES IN THE SMOOTHED SEQUENCE FROM ONE ITERATION TO THE NEXT. SEE ONE3HOW.

A3RSRHOW

SYNTAX: A3RSR VECTOR

GROUP: SMOOTH

SUBPROGRAMS: A3R, ONE3, SPLIT

DESCRIPTION: A3RSR DOES REPEATED SMOOTHINGS BY RUNNING MEDIANS OF 3 FOLLOWED BY SPLITTING MESAS AND REPEATING UNTIL CONVERGENCE. SEE A3R AND SPLIT.

BOXPLOT HOW

SYNTAX: BOXPLOT VECTOR

PARAMETER:

LENGTH= CONTROLS THE HORIZONTAL SIZE OF THE DISPLAY
(DEFAULT=50 CHARACTERS).

GROUP: DISPLAY

SUBPROGRAM: FILL

DESCRIPTION: BOXPLOT GENERATES A BOX PLOT DISPLAY FOR A VECTOR OF DATA. A RECTANGULAR BOX WITH ENDS CORRESPONDING TO LOWER AND UPPER QUANTILES IS PRESENTED WITH THE MEDIAN MARKED WITH AN ASTERISK. WHISKERS ARE DRAWN ON EACH SIDE OF THE BOX WITH CROSSES MARKING THE LOWEST AND HIGHEST DATA VALUES WITHIN AN INTERQUARTILE DISTANCE OF THE QUANTILES. DATA VALUES OUTSIDE THE CROSSES (OUTLIERS) ARE MARKED WITH CIRCLES AND THOSE MORE THAN 1.5 INTERQUARTILE DISTANCES GET HEAVY CIRCLES.

CHISQUARE HOW

SYNTAX: T CHISQUARE X

GROUP: GFIT

DESCRIPTION: CHISQUARE COMPARES A THEORETICAL DISCRETE PROBABILITY MASS FUNCTION T HAVING N POSSIBLE VALUES WITH AN EMPIRICAL FREQUENCY FUNCTION X OVER THE SAME N VALUES TO TEST HOW WELL THE PROBABILITIES T FIT THE DATA X. THE CHI-SQUARE GOODNESS-OF-FIT STATISTIC IS PRINTED OUT WITH THE NUMBER OF DEGREES OF FREEDOM.

CODERESHOW

SYNTAX: CODERES MATRIX OR CODESRES MEDPOLISH MATRIX

SUBPROGRAM: AND

GROUP: TWOWAY

DESCRIPTION: CODERES PRODUCES A DIAGNOSTIC ARRAY FOR ANALYZING THE RESIDUALS FROM MEDIAN POLISHING OF A TWO-WAY TABLE. THE USER SHOULD SET RESIDS=1 IN MEDPOLISH WHEN CODERES IS USED. THE OUTPUT MATRIX CONSISTS OF THE SYMBOLS -o+ DEPENDING ON THE SIZE OF THE RESIDUALS.

=: VALUE BELOW LOWER QUANTILE

o: VALUE BETWEEN QUANTILES

+: VALUE LARGER THAN UPPER QUANTILES

COMPAREHOW

SYNTAX: COMPARE MATRIX

PARAMETERS:

- (1) DEPTH- VERTICAL HEIGHT OF DISPLAY (DEFAULT=20 LINES).
- (2) MISS- NUMBER USED TO INDICATE MISSING VALUES (DEFAULT = -99999)
- (3) NGAP- NUMBER OF HORIZONTAL SPACES BETWEEN THE BOXPLOT DISPLAYS (DEFAULT=3).

GROUP: COMPARISONS

SUBPROGRAM: FILL

DESCRIPTION: COMPARE OPERATES ON AN N BY K MATRIX TO PRODUCE K VERTICAL BOX PLOTS PLACED NEXT TO EACH OTHER TO ALLOW VISUAL COMPARISON OF THE CENTERS, SPREADS, AND OUTLIERS OF THE BATCHES (COLUMNS) OF THE MATRIX. THE USER MUST FILL UP THE MATRIX SO THAT THERE ARE N OBSERVATIONS FOR EACH BATCH. THE PARAMETER, MISS, SHOULD BE USED TO FILL IN MISSING VALUES. THE USER SHOULD ASSURE THAT MISS IS DIFFERENT FROM ALL VALID ENTRIES IN THE DATA ARRAY.

CONDENSEHOW

SYNTAX: CONDENSE MATRIX OR R<CONDENSE MATRIX

PARAMETERS:

- (1) NUM- CONTROLS WHAT STATISTICS ARE INCLUDED IN THE SUMMARY. NUM \in [1,2,5,7] (DEFAULT=2).
NUM=1 GIVES MEDIANS OF EACH COLUMN OF THE MATRIX
NUM=2 GIVES MEDIANS AND INTERQUARTILE RANGES
NUM=5 GIVES MIN, QUARTILES, MAX, AND SIZE
NUM=7 GIVES MIN, EIGHTHS, MAX AND SIZE
- (2) MISS- NUMBER USED TO CODE MISSING VALUES (DEFAULT=-99999).

GROUP: COMPARISONS

SUBPROGRAMS: UTCOND, FMT

DESCRIPTION: CONDENSE GENERATES SUMMARY STATISTICS FOR EACH GROUP OF DATA REPRESENTED BY THE COLUMNS OF THE ARGUMENT MATRIX. SINCE THE ARGUMENT IS A MATRIX, THE USER MUST FILL UP THE MISSING VALUES IN THE MATRIX WHENEVER THE GROUPS (COLUMNS) HAVE DIFFERENT NUMBERS OF OBSERVATIONS. THE PARAMETER, MISS, SHOULD BE USED TO FILL THE MATRIX SINCE THE FUNCTION WILL RECOGNIZE THOSE VALUES AS MISSING DATA AND WILL IGNORE THEM IN ALL CALCULATIONS. THE RESULT FROM CONDENSE CAN BE USED AS THE ARGUMENT OF OTHER FUNCTIONS SUCH AS SCAT OR LINE IF AND ONLY IF NUM=2.

CONTINGENCYHOW

SYNTAX: CONTINGENCY MATRIX

GROUP: TWOWAY

DESCRIPTION: CONTINGENCY TAKES A TWO-WAY TABLE AND PERFORMS A TEST OF INDEPENDENCE OF ROWS AND COLUMNS. THE CHI-SQUARE STATISTIC AND ITS DEGREES OF FREEDOM ARE OUTPUT.

KSHOW

SYNTAX: F KS G

GROUP: GFIT

DESCRIPTION: KS DOES A KOLMOGOROV-SMIRNOV GOODNESS OF FIT TEST OF THE EQUALITY OF TWO DISCRETE EMPIRICAL PROBABILITY DISTRIBUTIONS. THE ARGUMENTS F AND G ARE EACH VECTORS OF OBSERVATIONS. THEY NEED NOT BE SORTED, NOR MUST THEY BE THE SAME SIZE. OUTPUT OF KS ARE THE VALUES:

MAX $F(X)-G(X)$ AND MIN $F(X)-G(X)$

LINEHOW

SYNTAX: LINE ARRAY OR Z+LINE ARRAY

PARAMETER:

RSELECT= SELECTS THE OUTPUT MATRIX OF RESIDUALS.

RSELECT=1 GIVES ABSCISSA VS. RESIDUALS (DEFAULT=1).

RSELECT≠1 GIVES FITTED VALUES VS. RESIDUALS

GROUP: RELATIONS

DESCRIPTION: LINE FITS A STRAIGHT LINE TO A SET OF (X,Y) POINTS BY DIVIDING THE POINTS INTO 3 REGIONS AND USING THE MEDIANS OF THE X AND Y VALUES IN THE OUTER REGIONS TO DETERMINE THE SLOPE. THE INTERCEPT IS THE MEDIAN OF THE DIFFERENCES $Y-SLOPE \times X$. LINE ALSO COMPUTES THE RESIDUALS AND GIVES AN N BY 2 MATRIX CONTAINING EITHER THE X-VALUES VS. RESIDUALS OR THE FITTED VALUES VS. RESIDUALS DEPENDING ON THE PARAMETER, RSELECT. IF THE USER DOES NOT WANT RESIDUALS TYPED OUT, HE SHOULD USE THE SYNTAX:

Z+LINE ARRAY

THE ARGUMENT ARRAY CAN BE EITHER A VECTOR OR A 2-COLUMN MATRIX. IF THE ARGUMENT IS A VECTOR OF SIZE N, THE X-VARIABLE IS CONSTRUCTED TO BE THE FIRST N POSITIVE INTEGERS. IF THE ARGUMENT IS A 2-COLUMN MATRIX, THE FIRST COLUMN IS TAKEN TO BE THE SET OF X-VALUES AND THE SECOND COLUMN THE SET OF Y-VALUES. LINE CAN BE USED IN CONJUNCTION WITH THE FUNCTION SCAT TO PRODUCE THE SLOPE, THE INTERCEPT, AND A PLOT OF THE RESIDUALS. TO DO THIS, ENTER:

SCAT LINE ARRAY

LSLINEHOW

SYNTAX: LSLINE ARRAY OR Z←LSLINE ARRAY

GROUP: RELATIONS

DESCRIPTION: LSLINE DETERMINES THE LEAST SQUARES SOLUTION, $Y=YI+SLOPE \times X$. THE ARGUMENT ARRAY CAN BE A VECTOR OR A 2-COLUMN MATRIX. IF THE ARGUMENT IS A VECTOR, THE X-VALUES ARE TAKEN TO BE THE INTEGERS 1 TO N WHERE $N=PAARRAY$. IF THE ARGUMENT IS A 2-COLUMN MATRIX, COLUMN 1 CONSISTS OF THE X-VALUES AND COLUMN 2 THE Y-VALUES. RESIDUALS ARE DETERMINED AND STORED IN A MATRIX WHOSE FIRST COLUMN IS THE SET OF X-VALUES AND WHOSE SECOND COLUMN IS THE SET OF RESIDUALS. IF THE USER WANTS TO SUPPRESS PRINTOUT OF THE RESIDUALS, HE MUST ASSIGN THE RESULTS OF LSLINE TO A VARIABLE; I.E., TYPE Z←LSLINE ARRAY.

MEDPOLISHHOW

SYNTAX: MEDPOLISH MATRIX OR Z←MEDPOLISH MATRIX

PARAMETERS:

- (1) RESIDS- CONTROLS OUTPUT OF RESIDUALS. IF 1, RESIDUALS ARE PRODUCED IN A TWO-WAY TABLE; IF 2, COMPARISON VALUES, $(RE \circ . \times CE) : TV$ AND RESIDUALS ARE PRODUCED IN A TWO COLUMN MATRIX (DEFAULT=1).
- (2) EPSILON- PROPORTION BY WHICH THE SUM OF THE ABSOLUTE VALUES OF THE RESIDUALS MUST BE REDUCED AT EACH ITERATION TO CONTINUE POLISHING (DEFAULT=0.01).
- (3) NORMEFFECTS- DETERMINES WHETHER OR NOT NORMALIZED EFFECTS ARE OUTPUT. THE DEFAULT OF 0 SUPPRESSES OUTPUT OF NORMALIZED VALUES. NORMEFFECTS=1 CAUSES NORMALIZED VALUES TO BE PRINTED.

GROUP: TOWAY

DESCRIPTION: MEDPOLISH ITERATIVELY SWEEPS OUT MEDIANS FROM THE ROWS AND COLUMNS OF A TWO-WAY TABLE TO YIELD THE MODEL:

$$OBS = MEDIAN + ROW \text{ EFFECT} + COLUMN \text{ EFFECT} + RESIDUAL$$

IT YIELDS RESIDUALS FOR TESTING THE ADEQUACY OF THE MODEL (SEE SHOWRES AND CODERES). THE FUNCTION CONTINUES POLISHING UNTIL THE STOPPING RULE CONTROLLED BY EPSILON IS ACTIVATED. THE OVERALL MEDIAN, THE ROW- AND COLUMN-EFFECTS, AND THE SUM OF ABSOLUTE VALUES OF THE RESIDUALS ARE GIVEN. THE USER CAN SUPPRESS RESIDUAL PRINTOUT BY ASSIGNING THE FUNCTION TO A VARIABLE, I.E., BY WRITING:

Z← MEDPOLISH MATRIX

CORRELATIONHOW

SYNTAX: R+CORRELATION W

GROUP: STATS

SUBPROGRAM: FMT

DESCRIPTION: CORRELATION DETERMINES THE SIMPLE PEARSON PRODUCT MOMENT CORRELATIONS BETWEEN EACH PAIR OF VARIABLES REPRESENTED BY THE C COLUMNS OF W. THE OUTPUT IS A C BY C CORRELATION MATRIX.

MSTATSHOW

SYNTAX: MSTATS W

GROUP: STATS

SUBPROGRAM: FMT

DESCRIPTION: MSTATS DETERMINES CORRELATIONS, MEANS, STANDARD DEVIATIONS, LOWER QUANTILES, MEDIANS, AND UPPER QUANTILES FOR THE COLUMNS OF THE ARGUMENT MATRIX W. THE CORRELATIONS ARE DISPLAYED IN AN N BY N MATRIX, WHERE N IS THE NUMBER OF COLUMNS OF W.

PARTIALHOW

SYNTAX: PARTIAL W

GROUP: STATS

SUBPROGRAMS: CORRELATION, FMT

DESCRIPTION: PARTIAL DETERMINES PARTIAL CORRELATIONS OF ANY SPECIFIED ORDER FOR THE COLUMNS OF THE MATRIX W. THE USER WILL BE ASKED FOR THE COLUMNS TO BE PARTIALLED OUT. THE USER'S RESPONSES MUST BE INTEGERS BETWEEN 1 AND C, WHERE C IS THE NUMBER OF COLUMNS OF W.

NUMSUMHOW

SYNTAX: NUMSUM VECTOR OR NUMSUM ARRAY

PARAMETER:

MISS= NUMBER USED TO INDICATE MISSING VALUE IN THE DATA
ARRAY (DEFAULT=-99999).

GROUP: COMPARISONS

SUBPROGRAM: FMT

DESCRIPTION: NUMSUM OPERATES ON EITHER A VECTOR OR AN ARRAY BUT, IN ALL CASES, THE DATA ARE TREATED AS A SINGLE BATCH. NUMSUM PRODUCES A NUMERICAL SUMMARY WHICH GIVES THE SAMPLE SIZE (AFTER DELETION OF MISSING VALUES), THE EIGHTHS, THE EXTREMES, THE SPREADS, AND THE MIDPOINTS IN TABULAR FORM. THIS SUMMARY IS USEFUL FOR TESTING THE SYMMETRY OF A DATA SET, AND TO EVALUATE THE EFFECTIVENESS OF DATA TRANSFORMATIONS IN PRODUCING SYMMETRY.

ONEHHOW

SYNTAX: ONEH VECTOR

GROUP: SMOOTH

DESCRIPTION: ONEH DOES A SINGLE HANNING OF A SEQUENCE OF DATA. IT IS USED IN CONJUNCTION WITH THE OTHER SMOOTHING FUNCTIONS AS A FINAL TOUCH-UP TO A SEQUENCE OF SMOOTHED DATA. THE I-TH RESPONSE IS REPLACED BY $0.25 \times Y[I-1] + 0.5 \times Y[I] + 0.25 \times Y[I+1]$. SEE A3R AND A3RSR.

ONE3HOW

SYNTAX: ONE3 VECTOR

GROUP: SMOOTH

DESCRIPTION: ONE3 SMOOTHS A VECTOR OF DATA USING ONE PASS OF RUNNING MEDIANS OF 3 WITH TUKEY'S END-POINT RULE. WHEN SMOOTHING (X,Y) PAIRS, THE USER SHOULD USE AS THE ARGUMENT VECTOR THE Y VALUES ORDERED ACCORDING TO THE MAGNITUDE OF THE X VALUES:

VECTOR+Y[AX]

IT IS ASSUMED THAT THE X VALUES ARE EQUISPACED. REPEATED SMOOTHING BY MEDIANS OF 3 CAN BE ACCOMPLISHED WITH A3R. OUTPUT CONSISTS OF THE SMOOTHED SEQUENCE. A PLOT OF THE SMOOTHED SEQUENCE IS GIVEN BY TYPING:

SCAT ONE3 VECTOR

REGRESSHOW

SYNTAX: Z←Y REGRESS X

PARAMETER:

ΔINTERCEPT. DETERMINES WHETHER OR NOT AN INTERCEPT TERM IS TO BE INCLUDED. ΔINTERCEPT=1 GIVES AN INTERCEPT TERM, AND ΔINTERCEPT=0 GIVES NO INTERCEPT. (DEFAULT IS 1.)

GROUP: RELATIONS

SUBPROGRAMS: FMT AND SCAT

DESCRIPTION: REGRESS DOES A MULTIPLE REGRESSION ANALYSIS RELATING THE DEPENDENT VARIABLE Y TO A SET OF CARRIERS X. THE LEFT ARGUMENT Y IS A VECTOR OF SIZE N. THE RIGHT ARGUMENT X IS AN N BY K MATRIX CONSISTING OF N OBSERVATIONS ON EACH OF K VARIABLES OR A VECTOR OF SIZE N IF K=1. OUTPUT CONSISTS OF AN ANOVA TABLE, R-SQUARE, STD. ERROR, REGRESSION COEFFICIENTS (THE FIRST COEFFICIENT IS THE CONSTANT TERM IF ΔINTERCEPT=1.), T-STATISTICS, VARIANCE-COVARIANCE MATRIX, DURBIN-WATSON STATISTIC, AND A VECTOR OF PREDICTED Y VALUES AND RESIDUALS. THERE IS AN OPTION THAT ALLOWS THE USER TO INPUT A VECTOR OF X VALUES AND USE THE REGRESSION EQUATION TO FORECAST Y VALUES. THE USER CAN ALSO OBTAIN A SCATTER PLOT OF THE RESIDUALS. WHEN EXECUTION TERMINATES, THE PREDICTED Y VALUES AND THE RESIDUALS RESIDE IN THE N BY 2 MATRIX Z.

STATISTICSHOW

SYNTAX: STATISTICS VECTOR

GROUP: STATS

SUBPROGRAM: FMT

DESCRIPTION: STATISTICS DETERMINES THE MEAN, VARIANCE, STANDARD DEVIATION, COEFFICIENT OF VARIATION, LOWER AND UPPER QUANTILES, MEDIAN, TRIMEAN, MIDMEAN, MIDRANGE, RANGE, MEAN ABSOLUTE DEVIATION, INTERQUARTILE RANGE, SKEWNESS, AND KURTOSIS FOR THE DATA IN VECTOR.

SCATHOW

SYNTAX: SCAT ARRAY

PARAMETERS:

(1) WID- CONTROLS THE HORIZONTAL SIZE OF THE DISPLAY (DEFAULT=30 CHARACTERS).

(2) DEP- CONTROLS THE VERTICAL SIZE OF THE DISPLAY (DEFAULT=15 LINES).

(3) NDIVX, NDIVY- NUMBER OF UNITS ON X- AND Y-AXES, RESPECTIVELY (DEFAULT=4,4).

GROUP: RELATIONS, SMOOTH, COMPARISONS

DESCRIPTION: SCAT PRODUCES A SCATTER PLOT OF THE DATA CONTAINED IN ARRAY. THE ARGUMENT ARRAY CAN BE A VECTOR OR A MATRIX WITH AS MANY AS 9 COLUMNS. IF A VECTOR OF SIZE N, THOSE VALUES ARE PLOTTED VS. THE INTEGERS 1 TO N; IF A MATRIX, THE SECOND, THIRD, ETC. COLUMNS ARE PLOTTED VS. COLUMN 1 ON THE SAME AXES. DIVISIONS ON THE AXES OF THE PLOT ARE NOT EXPLICITLY PRINTED, EXCEPT AT THE EXTREMES OF THE PLOT. THE USER CAN CONTROL THE RESOLUTION OF THE PLOT BY MODIFYING THE PARAMETERS WID, DEP, NDIVX AND NDIVY. PRINTING TIME INCREASES DRAMATICALLY WITH RESOLUTION; THEREFORE, OTHER PLOT PROGRAMS SHOULD BE USED IF HIGH RESOLUTION IS DESIRED. FOR A SINGLE GROUP OF DATA, THE NUMBER N ($2 \leq N \leq 9$) WILL BE PRINTED IF N POINTS LIE CLOSE TOGETHER ON THE DISPLAY. FOR MULTIPLE PLOTS ON THE SAME DISPLAY, THE LETTER A REPRESENTS GROUP 1, B REPRESENTS GROUP 2, ETC. FOR 2 POINTS CLOSE TOGETHER THE LETTER WILL BE PRINTED WITH AN UNDERSCORE. THE DISPLAY CANNOT HANDLE 3 OR MORE POINTS CLOSE TOGETHER IF THERE ARE MULTIPLE PLOTS.

SHOWRESHOW
 SYNTAX: SHOWRES MATRIX OR SHOWRES MEDPOLISH MATRIX
 SUBPROGRAM: AND
 GROUP: TOWAY
 DESCRIPTION: SHOWRES AIDS IN THE ANALYSIS OF RESIDUALS FROM
 MEDIAN POLISHING A TWO-WAY TABLE. THE USER SHOULD SET RESIDS=1
 IN MEDPOLISH IF SHOWRES IS TO BE USED. THE OUTPUT OF SHOWRES IS
 A MATRIX CONSISTING OF THE SYMBOLS 000X DEPENDING ON THE SIZES
 OF THE RESIDUALS. LET R REPRESENT THE RESIDUAL, M THE MIDSPREAD
 OF ALL RESIDUALS, L THE LOWER QUANTILE, AND U THE UPPER QUANTILE.
 THE CODES ARE PRINTED AS FOLLOWS:
 0: $R < L - 1.5M$
 o: $L - 1.5M \leq R \leq L - M$
 °: $L - M < R < U + M$
 x: $U + M \leq R \leq U + 1.5M$
 X: $R > U + 1.5M$

SPLITHOW
 SYNTAX: SPLIT A3R VECTOR
 GROUP: SMOOTH
 DESCRIPTION: SPLIT DOES ONE PASS AT DIVIDING MESAS (PAIRS OF
 ADJACENT POINTS WITH A COMMON VALUE WHICH IS A LOCAL MAX OR MIN)
 USING TUKEY'S END-POINT RULE. IT IS USED IN CONJUNCTION WITH
 A3R.

STEMLEAFHOW

SYNTAX: STEMLEAF VECTOR

GROUP: DISPLAY

PARAMETERS:

(1) WIDTH- CONTROLS THE WIDTH (CHARACTERS PER LINE) OF THE DISPLAY (DEFAULT=70).

(2) SCALE- VARIES THE DEPTH (STEM INTERVAL) OF THE DISPLAY IN UNITS OF 1, 2, OR 0.5 TIMES A POWER OF 10.

SELECT AN INTEGER FROM 1 TO 3 (DEFAULT=1).

SUBPROGRAM: LIT

DESCRIPTION: STEMLEAF GENERATES A STEM AND LEAF DISPLAY OF A VECTOR OF OBSERVATIONS. THE FUNCTION AUTOMATICALLY SCALES THE DATA USING A SCALING ROUTINE BASED ON THE RANGE AND SIZE OF THE DATA VECTOR. BY VARYING THE SCALE PARAMETER THE USER CAN CHANGE THE SIZE OF THE STEM, BUT IT WILL NOT NECESSARILY BE SCALED BY THE AMOUNT SPECIFIED. TRY VARIOUS VALUES LIKE 1,2, AND 3 TO SEE WHICH CHOICE YIELDS THE BEST DISPLAY. IF A LEAF CONTAINS TOO MANY CHARACTERS TO BE PRINTED ON A LINE, THE LINE WILL BE TRUNCATED AND THE NUMBER OF TRUNCATED CHARACTERS WILL BE WRITTEN AT THE END OF THE LINE. IF THE STEM INTERVAL IS TWO TIMES A POWER OF TEN, THE LEAVES MAY CONTAIN THE CHARACTERS A,B,C,...,J IN ADDITION TO THE DIGITS 0,1,...,9. IN THIS CASE, THE CHARACTERS A TO J REPRESENT THE NUMBERS 10 TO 19.

TWICEHOW

SYNTAX: TWICE VECTOR

GROUP: SMOOTH

SUBPROGRAMS: ONE3, SPLIT, A3RSR

DESCRIPTION: TWICE SMOOTHS A SEQUENCE OF DATA, THEN SMOOTHS THE RESIDUALS AND ADDS THE SMOOTHED RESIDUALS BACK TO THE SMOOTHED DATA TO OBTAIN THE FINAL SMOOTH. THE SMOOTHING IS DONE USING A3RSR.

DEFAULT VALUES OF THE PARAMETERS:

	DEP
15	DEPTH
20	EPSILON
0.01	LENGTH
50	MISS
-99999	NDIVX
4	NDIVY
4	NGAP
3	NORMEFFECTS
0	NUM
2	OPTION
1	RESIDS
1	SCALE
1	RSELECT
1	WID
30	WIDTH
70	ΔINTERCEPT
1	CH

APPENDIX B

AD-A063 145

NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF
A USER'S GUIDE TO THE OA3660 APL WORKSPACE. (U)
OCT 78 F R RICHARDS
NPS55-78-028

F/G 9/2

UNCLASSIFIED

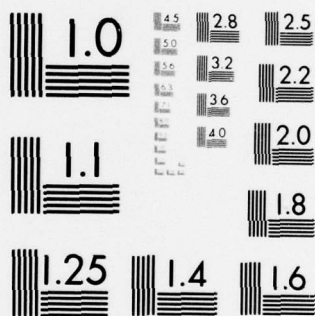
NL

2 OF 2

AD
A063145



END
DATE
FILMED
3-79
DDC



```

VAND[ ]V
V L←A AND B;C;D
[1] →(((2<ppA)∨3<ppB),0≠ppB)/ 17 3
[2] B←,b
[3] →(((3=ppB)∧1≠1ppB),2=ppA)/ 17 7
[4] A←,A
[5] →(∧/((pA)≠1,D),1≠D+1p-2φpB)/16
[6] A←(((D×pA)LD(pA),1)pA
[7] →(1≠ppB)/9
[8] B←(((pB)∧(1=ppB)×1ppA),1)pB
[9] →(∧/D≠1,1ppA),1≠D+1p-2φpB)/ 16 11
[10] B←(((3=ppB)p1),(1ppA),1pφpB)pB
[11] →(3=ppB)/14
[12] L←(((C+1pφpA)p0),(1pφpB)p1)\B
[13] →0×ppL[;iC]←A
[14] L←(1,((C+1pφpA)p0),(-1+1pφpB)p1)\B
[15] →0×ppL[;1+iC]←A
[16] →0=p□+'ARGUMENTS OF AND ARE NOT CONFORMABLE.'
[17] 'AN ARGUMENT OF AND IS OF IMPROPER RANK.'
V

```

```

VANOVA[ ]V
V ANOVA;Y;K;BTSS;MSR;R;MAX;BLSS;MSBL;MSE;WTSS;EDF;TDF;TSS;YBAR;N;
T;BLOCK
[1] INPUT
[2] SUMSQ
V

```

```

VA3R[ ]V
V Z←A3R X;Y1;Y2;X1;X2;N;C
[1] Z←ONE3 X
[2] →1×(pppX←Z)×0≠+ / |Z=X
V

```

```

VA3RSR[ ]V
V Z←A3RSR X
[1] X←A3R X
[2] Z←A3R SPLIT X
[3] →2×(pppX←Z)×0≠+ / |Z=X
V

```

```

VBOXPLOT[ ]V
V Z←BOXPLOT X;L;U;B
[1] X←(÷1E-20+U÷L)×((U÷L,X)→LENGTH×L÷L÷L,X)+,X×LENGTH-1
[2] 0.1×[0.5+10×L;(0[LENGTH-6)ρ' '];0.1×[0.5+10×U
[3] →9×10=U÷L
[4] Z←LENGTH FILL X
[5] B←' | | | ' [1+Z]
[6] B←B, ' *|*|* →x 0000000000 ●●●●●●●●●● '[1+Z]
[7] B←B, ' _|_|_ 23456789 23456789 '[1+Z]
[8] →0×ρρZ←(3,LENGTH)ρB
[9] 'RANGE OF DISTRIBUTION IS ZERO'
V

```

```

VCHISQUARE[ ]V
V T CHISQUARE X;DF;C
[1] →3×1((ρX)=ρT)
[2] →0×ρ[ ]←'LENGTH ERROR: X AND T MUST BE THE SAME SIZE.'
[3] C←+/(X-T)×2)÷T←T×+/X
[4] 'CHI-SQUARE VALUE = ' ;C; ' DF = ' ;(ρX)→1
V

```

```

VCODERES[ ]V
V Z←CODERES X;Y;N;Q;LW;UW
[1] N←ρY+Y[AY←,X]
[2] Q←0.5×Y[L0.25×3+N×13]+Y[L0.25×1+N×13]
[3] LW←Q[1]→D←Q[3]→Q[1]
[4] UW←Q[3]→D
[5] Y←(X>LW)+(X>Q[1])+(X>Q[3])+X>UW
[6] Z←'→→→++' [1+Y]
[7] Z←((ρZ[;1]),2×ρZ[1;])ρ(,Z) AND,(ρZ)ρ' '
V

```

```

VCOMPARE[ ]V
V Z←COMPARE X;U;L;Y;M;C;J;P
[1] X←(+1E-20+U-□+L)×((□+U+[/Y)-DEPTH×L+[/Y+(MISS≠,X)/,X)+X×DEPTH-1
[2] M←(+1E-20+U-L)×(U-DEPTH×L)+MISS×DEPTH-1
[3] Z←(DEPTH,6×C+ρX[1;])ρ' '
[4] J←0
[5] J←J+1
[6] P←DEPTH FILL P[▲P+(P≠M)/P+X[;J]]
[7] Z[;-2+6×J]←' | -1-T| '[1+P]
[8] Z[;-1+6×J]←' *T=1= | × 0000000000 ●●●●●●●● '[1+P]
[9] Z[;6×J]←' | -1-T| 23456789 23456789 '[1+P]
[10] +5×J < C
[11] Z←Z
V

```

```

VCONDENSE[ ]V
V R←CONDENSE X;I;N;NR;U;Q;Z
[1] R←10
[2] UTCOND X
[3] +0×1(NUM≠2)
[4] R←(NR,2)ρZ
V

```

```

VCONTINGENCY[ ]V
V CONTINGENCY F;R;C;T;E;CHI2;DF
[1] R←+F
[2] T←+/C+ /F
[3] E←C°.×R+T
[4] CHI2←+/,((F-E)*2)÷E
[5] DF←×/(ρF)-1
[6] 'CHI-SQUARE VALUE= ';CHI2;' DF= ';DF
V

```

```

VCONDENSE[ ]V
V R←CONDENSE X;I;N;NR;U;Q;Z
[1] R←10
[2] UTCOND X
[3] →0×1(NUM≠2)
[4] R←(NR,2)ρZ
V

```

```

VCONTINGENCY[ ]V
V CONTINGENCY F;R;C;T;E;CHI2;DF
[1] R←+F
[2] T←+/C←+/F
[3] E←C○.×R÷T
[4] CHI2←+/,((F÷E)*2)÷E
[5] DF←+/(ρF)÷1
[6] 'CHI-SQUARE VALUE= ';CHI2;' DF= ';DF
V

```

```

VCORRELATION[ ]V
V R←CORRELATION W;Z;C;S;CH;MEANS;VAR
[1] C←(QZ)+.×Z+W÷(ρW)ρ(MEANS←+÷W)÷1+p(-2+1,1,ρW)ρW+pCH+' '
[2] R←'BF8.2' FMT(0,1ρS),[1](1ρS),C÷S○.×S÷(VARS←+÷Z*2)*
0.5
V

```



```

VFMT[ ]V
V OL←E FMT R;S;W;Δ;G;X;T;K;J;M;Q;P;D;N;O;L;B;V;CH;H
[1] N←Q+1+M+ρR+(1⌈-2+ρR)ρR
[2] OL←((1=1+M)+ 1 0 ×M+M+2+H+1<ρCH+CH,',' )ρΔ←'0123456789.'
[3] →E×1(N+O=N)VV+1≥ρS←,E
[4] L0:→~1VV(xP+4×Q=ρK+ρX←' ')^V/('A',O+ '□')∈S
[5] →(L0+(V+O=ρS+J+S)+1B=M[2]+1),L←(1×B+O+.=K),P×~'A'∈K+K,(J+S1',' )
    ↑S
[6] →E+×ρS←'TEXT DELIMITER'
[7] →L3=3××(ρG+K=K+(Kε-1+Δ)/K)⌊W+ρX←(ρK+(K1O)+K)+(=(φK)1O)+K
[8] L←:(D+~1+G+K∈Δ)/L3=2×(ρK)≠W+1+O+ 'XA'∈K+(~Kε' ,')/K
[9] →L3×1(B≠+/G)×M[2]+101⌊1=Δ1(B+⌊1=G1O)+K
[10] →L3=φO,=(L+ 'EET'∈K)/×W+101⌊1=Δ1(⌊1=G+B1',' )+B←(1=(φG)1O)+K
[11] A←(1+ρX←((1⌈ρA)⌊(M[1]=H),W)+A)φA
[12] L3:→(HD×1HΛ~'X'∈K),E=ρX←=W,D+OρP←((M=H,0)×1,W)ρX
[13] →L4=1~1+L,Q+1+ρR←(0 1 ×ρP+R[; 1M[2]+Q⌊M[2]⌈Q×VΛD))+R
[14] P+P÷10×L+⌊10●⌋P+O=P
[15] →L3×1O=J++/B←('B'∈K)×O=P←(⌊0.5+N×,P)÷N+10×D+101⌊1=Δ1G+B
[16] L4:→(ρ1+ρL)/F=ρρX←(1 0 ×ρG+JρT\ ' ' )ρJ+J,O+V/T+O>P+B/P
[17] →(×L←(O⌈L×J+ 'Z'∈K)⌈.×~T←(T+O+1+⌊10●1⌈⌋P)>O+L+W=D+O+~2+L)/L/F,F,
    I
[18] →E+×ρS←'FIELD WIDTH'
[19] →L4+1+1((J[2]+Lv.<O)+O+1+10⌈.≤⌋L+(B/,L)+T+10=⌊P>W=D+O+
    3
[20] T←~J+P[T/11+J]+L+ρ1ρX←'E','+0~'[Jρ2=×L],Δ[1+Q(Oρ10)τ⌊L]
[21] F:→(Jv2≥D×~'T'∈K)/I,N+ρX+Δ[11,1+Q(Dρ10)τ⌈N×1⌋⌊P],X
[22] D←,(-N)+(D⌈.×φX[;2+D]≠1+Δ)○.<D+1D=1
[23] X←NρX,X[D/1ρX←,X]←' '
[24] I:→(J+JvO=+/O+O⌈L=O)/I+ρD+ρP+G,Δ[1+Q(Lρ10)τ⌊⌋P]
[25] P+Dρ(,O+GφO)\(,O+O○.<(-G)φ1L+G+1+ρG)/,P
[26] →HD=1JvL←~'L'∈K,P[T/1D+1+X+ρP+P,X;]←'*'
[27] P+Xρ(,φO)\(,O+~X+~O)/,P
[28] →(~H)/E=N+1,D+OρP+B×(D,X←W×1=2×L)+P
[29] HD:CH←(ρK←(⌈-1+D+O,(M[2]⌊ρD)ρD←(' '=CH)/1ρCH)ρCH)φCH
[30] D←,(M[2],X)+ 0 -1 +(M[2],B)ρ(,φD○.≥1B+⌈/D+1+D=⌈-1φD)\K
[31] →(L0=VΛ×Q),ρOL+OL,((1=1+M)+M×1,W)ρD,,P
[32] E:K←'NO VALID E, I, OR F PHRASE'
[33] 'FMT PROBLEM ' ,K;(1,ρS)ρS
V

```

```

VFILL[ ]V
V Z←W FILL X;N;D;X1;X2;X3;X4;Y;I;J;S
[1] N←pX←X[1]X←X
[2] Z←0.5×X[10.25×3+N×13]+X[10.25×1+N×13]
[3] S←(Z[1]←0.5+1.5×D),(Z[1]←0.5+D+Z[3]←Z[1]),Z[1]
[4] S←[0.5+S,Z[2],Z[3],[Z[3]+0.5+D],Z[3]+0.5+
1.5×D
[5] X1←(X2≤S[1])/X2+(X≤S[2])/X+10.5×X
[6] X2←(X2>S[1])/X2
[7] X4←(X3≥S[7])/X3+(X≥S[6])/X
[8] X3←(X3<S[7])/X3
[9] Y←((X>S[2])×X<S[6])/X
[10] Z←Wp0
[11] Z[I]+20+9[1](-1,X1)°.=I+1(S+0[S[W])[1]
[12] Z[I]+10+9[1](-1,X2)°.=I+S[1]+11+S[2]←S[1]
[13] Z[I]+10+9[1](-1,X3)°.=I+1+S[6]+11+S[7]←S[6]
[14] Z[I]+20+9[1]X4°.=I+1+S[7]+11+W←S[7]
[15] +18×10=pY
[16] Z[J+1+Y]+Z[I+1+Y]+9
[17] Z[I+10[J←I+1]]+8
[18] Z[S[3]+10[S[5]←S[3]+1]]+6
[19] Z[S[3],S[5]]+0
[20] Z[S[4]]+1
[21] Z[S[3],S[5]]+Z[S[3],S[5]]+ 2 4
V

```

```

VINPUT[ ]V
V INPUT;J;M;Z
[1] 'ENTER NUMBER OF TREATMENTS.'
[2] J←0×K←0
[3] 'ENTER VECTOR OF NUMBER OF OBS. FOR TREATMENTS 1 TO ';K
[4] Y←((MAX←[1/N+0],K)p0
[5] +7×1(K=pN)
[6] +4×pp←'RE-ENTER COUNT VECTOR, ONE ELEMENT PER TREATMENT.'
[7] +0×1(K<J+J+1)
[8] 'ENTER ';N[J];' OBSERVATIONS FOR TREATMENT ';J
[9] +7×pppY[;J]+(MAX,1)pZ,(M←MAX←pZ+0)p0
V

```

```

VKS[ ]V
V F KS G;Z;N;X;NF;NG;MAX;SIGN;-
[1] D+NF+1+0*N+pZ+Z[ΔZ+F,G]
[2] →A1×1(N<→+1)
[3] NF+NF,(+/F≤X+Z[ ])÷pF
[4] NF+NF,(+/F<X)÷pF
[5] NG+(+/G≤X)÷pG
[6] SIGN+×NF+,NF0,-NG,(+/G<X)÷pG
[7] D+D,MAX×1+((NF=MAX+[/NF+|NF|)/SIGN)
[8] →2,NF+10
[9] A1:'MAX F(X)-G(X)=';[/D
[10] 'MIN F(X)-G(X)=';[/D

```

```

VLINE[ ]V
V Z+LINE W;X;Y;N;M;P;I;J;K;X1;Y1;X2;Y2;F;SL;YI
[1] →3×12=ppW
[2] W+Q(2,pW)p(1pW),W
[3] F+(N+pX+X[P+ΔX+W[;1]))pK+SL+0
[4] X1+0.5×X[I+10.5×M+1]+X[J+10.5×1+M+10.5×N+3]
[5] X2+0.5×X[N+1-I]+X[N+1-J]
[6] Y1+0.5×Z[I]+(Z+Z[ΔZ+M+Y+W[P;2]-F])[J]
[7] Y2+0.5×Z[I]+(Z+Z[ΔZ+(M)+Y])[J]
[8] F+(SL+SL+(Y2-Y1)÷X2-X1)×X
[9] K+K+1
[10] →6×1K<2
[11] YI+0.5×Y[10.5×N+1]+(Y+Y[ΔY+W[P;2]-F])[10.5×N+1]
[12] 'SLOPE: ';SL;' Y-INTERCEPT: ';YI
[13] Z+W[;2]-F+YI+SL×W[;1]
[14] →16×1KSELECT=1
[15] →0,,Z+Q(2,N)pF,Z
[16] Z+Q(2,N)pW[;1],Z

```

```

VLIT[ ]V
V R+LIT A
[1] R+,'0123456789'[1+((1+[10●A)p10)τA]

```

```

VLSLINE[ ]V
V Z←LSLINE X;N;B
[1] →A1×12=ppX
[2] X←Q(2,N)pX←(1N+pX),X
[3] A1:B←X[;2]⊕(Q(2,N)p(,X[;1]),(N+1+pX)p1)
[4] 'SLOPE: ' ;B[1];' Y-INTERCEPT: ' ;B[2]
[5] Z←Q(2,N)p(X[;1]),X[;2]←(B[2]+X[;1]×B[1])
V

```

```

VMEDPOLISH[ ]V
V Z←MEDPOLISH X;k;C;N;RI;CI;RA;CA;S;RG;RE;CE;TV
[1] R←pX[;1]
[2] N←R×C←pX[1;]
[3] X←X←TV←0.5×Z[10.5×N+1]+(Z←Z[ΔZ←,X])[10.5×N+1]
[4] RE←Rp0
[5] CE←Cp0
[6] Δ←S←+/|,X
[7] Z←(R,C)p(,X)[Δ,X+(1R)○.×Cp1×RG←(1/,X)←1/,X]
[8] X←X←(RI←+/0.5×Z[;10.5×C+1]+Z[;10.5×C+1])○.×Cp1
[9] RE←RE+RI
[10] Z←Q(C,R)p(,QX)[Δ,QX+(Rp1)○.×(1C)×RG]
[11] X←X←(Rp1)○.×CI←+/0.5×Z[10.5×R+1;]+Z[10.5×R+1;]
[12] CE←CE+CI
[13] →6×1EPSILON<1←+/|,X÷S
[14] Z←X
[15] →17×1RESIDS≠2
[16] Z←Q(2,N)p((,RE○.×CE)÷TV),,Z
[17] →21×1NORMEFFECTS=0
[18] RE←RE←RA←(+/RE)÷R
[19] CE←CE←CA←(+/CE)÷C
[20] TV←TV←RA+CA
[21] 'TYPICAL VALUE: ' ;TV
[22] 'ROW EFFECTS: ' ;RE
[23] 'COLUMN EFFECTS: ' ;CE
V

```



```

V MSTATS[ ] V
V MSTATS W;Z;C;S;CH;MEANS;VARS;N;J
[1] C+(QZ)+.xZ+W-(pW)pMEANS+(+W)÷N+1+p(-2+1,1,pW)pW+pCH+' '
[2] 'BF8.2' FMT(0,1pS),[1](1pS),C÷S+.xS+(VARS++Z*2)*
    0.5
[3] 'MEANS' [ ],BF8.2' FMT MEANS
[4] 'STD.DEV' [ ],BF8.2' FMT S+S÷(N-1)*0.5
[5] Z+1J+0
[6] →END×1(pS)<J+J+1
[7] C+X[AX+W[;J]]
[8] →6,Z+Z,C+0.5×C[10.25×3+N×13]+C[10.25×1+N×13]
[9] END;Z+((pS),3)pZ
[10] 'L. QRTLS' [ ],BF8.2' FMT Z[;1]
[11] 'MEDIANS' [ ],BF8.2' FMT Z[;2]
[12] 'U. QRTLS' [ ],BF8.2' FMT Z[;3]
V

```

```

V NUMSUM[ ] V
V NUMSUM X;N;Q;T1;CH
[1] CR
[2] '
[3] 70p' '
[4] N+pX+X[AX+(X≠MISS)/X+,X]
[5] 'SAMPLE SIZE = ';N
[6] 70p' '
[7] Q+0.5×X[10.125×7+N×17]+X[10.125×1+N×17]
[8] T1+Q[4],0,Q[4],0,0,((Q[2]+Q[6])÷2),Q[2],0
[9] T1+T1,Q[6],(Q[6]-Q[2]),((Q[1]+Q[7])÷2)
[10] T1+T1,Q[1],0,Q[7],(Q[7]-Q[1]),((X[1]+X[N])÷2)
[11] T1+T1,X[1],0,X[N],X[N]-X[1]
[12] T1+ 4 5 pT1
[13] CH+'MIDPTS,.,LOQ/8/MIN,.,MEDIAN,.,UPQ/8/MAX,.,SPREADS,'
[14] 'BF10.2, [ ] | [ ]' FMT T1
[15] 70p' '
V

```

```

V ONEH[ ] V
V H+ONEH X;A;B;C
[1] A+0,X,0
[2] B+X,0,0
[3] C+0,0,X
[4] H+ -2+(C+B+2×A)÷4
[5] H+X[1],2+H,X[pX]
V

```



```

VONE3[ ]V
V Z←ONE3 X;Y1;Y2;X1;X2;N;C
[1] Y1←(3×X[2])→2×X[3]
[2] Y2←(3×X[N→1])→2×X[-2+N←pX]
[3] X1←Y1, -1+X
[4] X2←1+X, Y2
[5] Z←(Y1,X,Y2)[(1N)+1+(C=X1<X)→(X≤X2)=C+X1<X2]
V

```

```

VPARTIAL[ ]V
V P←PARTIAL W;C;CIN;M;Y;X;BETA
[1] M←1+pW
[2] 'ENTER COLUMNS OF MATRIX TO BE PARTIALLED OUT.'
[3] A1→A2×10÷+/~(C←[ ]∈M
[4] Y←W[;CIN←(~M∈C)/M]
[5] →0,pP←CORRELATION R←Y→X+.×BETA←Y↔X+1,(2↑(pW[;C]),1)pW[;C]
[6] A2→A1,p'UNACCEPTABLE VALUES, ENTER INTEGERS FROM 1 TO ';M
V

```

```

VSHOWRES[ ]V
V Z←SHOWRES X;Y;N;Q;D;LW;LF;UW;UF
[1] N←pY←Y[AY←,X]
[2] Q←0.5×Y[[0.25×3+N×13]+Y[[0.25×1+N×13]
[3] LW←Q[1]→D←Q[3]→Q[1]
[4] UW←Q[3]+D
[5] LF←LW→D÷2
[6] UF←UW→D÷2
[7] Y←(X>LF)+(X>LW)+(X>Q[1])+(X>Q[2])+(X>Q[3])+(X>UW)+X>UF
[8] Z←'000000×X'[1+Y]
[9] Z←((pZ[;1]),2×pZ[1;])p(,Z) AND,(pZ)p' '
V

```

```

VREGRESS[ ]V
V Z←Y REGRESS X;N;K;C;XPXINV;XPY;BETA;RSS;TSS;S2;ESS;WID;DEP
[1] X←(2+(pX),1)pX
[2] X←(0,1-ΔINTERCEPT)+1,X
[3] XPXINV←(QX)+.×X
[4] BETA←XPXINV+.×XPY-(QX)+.×Y
[5] RSS←((QBETA)+.×XPY)-C+((+Y)*2)÷N+p,Y
[6] ESS←(TSS-((QY)+.×Y)-C)-RSS
[7] S2←,ESS÷(N-1)-K+(p,BETA)-ΔINTERCEPT
[8] CR
[9] ' ANOVA'
[10] CH←'SOURCE,DF,SUM SQUARES,MEAN SQUARE,F-RATIO'
[11] ' '
[12] 'REGRESSION',I4,BE16.4' FMT(K),(,RSS),(,RSS÷K),(,RSS÷K)÷S2
[13] CH←' '
[14] 'RESIDUAL',I4,BE16.4' FMT((N-1)-K),(,ESS),S2,0
[15] 'TOTAL',I4,BE16.4' FMT(N-1),(,TSS),0,0
[16] ' '
[17] 'R SQUARE: ',,RSS÷TSS
[18] 'STD ERROR: ',,S2*0.5
[19] CH←'COEFFICIENTS,T STATISTICS'
[20] 'F15.4' FMTQ(2,p,BETA)p(,BETA),(,BETA)÷(1 1 QV+S2×XPXINV)*
0.5
[21] 'DO YOU WANT A PRINTOUT OF THE VARIANCE-COVARIANCE MATRIX?'
[22] →A1×1'Y'≠1↑
[23] 'VARIANCE-COVARIANCE MATRIX: ',CH←' '
[24] 'E12.4' FMT V
[25] A1:'DURBIN-WATSON: ';(+(1+C)-(-1+C))*2)÷+/(,C+Y-X+.×BETA)*
2
[26] Z←Q(2,N)p(,X+.×BETA),,C
[27] B1:'DO YOU WANT TO FORECAST A VALUE FOR Y?'
[28] →C1×1'Y'≠1↑
[29] 'ENTER X VECTOR (';K;' VALUES)'
[30] 'FORECAST OF Y VALUE: ';(C+(1-ΔINTERCEPT)+1,)+.×BETA
[31] 'VARIANCE OF FORECAST ERROR: 'S2×1+C+.×XPXINV+.×QC
[32] →B1
[33] C1:'DO YOU WANT TO SCAT RESIDUALS VS. PREDICTED Y?'
[34] →0×1'N'=1↑
[35] DEP←0.5×WID+L/70,(I/((0.75×N),30))
[36] SCAT Z
V

```

```

VSCAT[ ]V
V W+SCAT Z;N;X;Y;C;R;U;S;L;I;J;K;UT;CL;G;D;B;A;C;V
[1] +3*(2=+/2=N)v(x/N)>+/N+ρZ
[2] Z+Q(2,ρZ)ρ(ρZ),Z+,Z
[3] Y+Z[;1+1C+1+(ρZ)[2]]
[4] R+ρZ+X+,Z[;1]
[5] L+U+S+2ρ0
[6] J+1+0*ρ(D+NDIVX,NDIVY),B+WID,DEP
[7] UT+10*[10●CL+1E-20+((U[J]+[ /Z)-S[J]+[ /Z):D[J]
[8] S[J]+UT*[S[J]:UT+UT[1+Δ|CL=UT+(1 2 5)*UT]
[9] U[J]+UT*[U[J]:UT
[10] L[J]+1+G*[L(B[J]=1)+G+(U[J]*S[J]):UT
[11] Z+,Y
[12] +7*13>J+J+1
[13] A+(φL)ρ0
[14] X+1+L0.5+(L[1]=1)*(X=S[1]):U[1]=S[1]
[15] Y+1+L0.5+(L[2]=1)*(Y=S[2]):U[2]=S[2]
[16] I+1
[17] +20*1<C
[18] A[Y[I;1];X[I]]+10[A[Y[I;1];X[I]]+1
[19] +18+6*R<I+I+1
[20] J+1
[21] D+0=V+A[Y[I;J];X[I]]
[22] A[Y[I;J];X[I]]+(10*V>K+1)+((K+1)*K=V)+(K+35*2*J)*D
[23] +21*1R≥I+I+1
[24] +21*1C≥J+J+I+1
[25] O+(φρA)L1[1+L0.5+(L=1)*S:S=U
[26] A[;C[1]]+A[;C[1]]+36*0=A[;C[1]]
[27] A[C[2];]+A[C[2];]+35*0=A[C[2];]
[28] W+' 023456789ⓁLKKJJIHHGGFFEEEDDCBBAA-'[1+●A>
]
[29] 'RANGE OF X: ';S[1],U[1]
[30] 'RANGE OF Y: ';S[2],U[2]

```

V

```

      VSPLIT[ ]V
V Z←SPLIT X;P;Q;R;S;T;N;I;J;C
[1] P←3+X
[2] Q←2+1+X
[3] R←1+2+X
[4] S←3+X
[5] I←(((P<Q)×R>S)+(P>Q)×R<S)×Q=R
[6] Q←(X+X[1],X,X[N])[1+I+(I≠0)/I+I×1-3+N+pX]
[7] R←(3×Q)÷2×X[I]
[8] P←X[I+2]
[9] T←(3×S+X[I+4])÷2×X[I+5]
[10] X[I+2]←(R×1=J)+(P×1=J)+Q×0=J+(C=P<Q)∧(Q≤R)=C+P<R
[11] X[I+3]←(T×1=J)+(P×1=J)+S×0=J+(C=P<S)∧(S≤T)=C+P<T
[12] Z←1+1+X
V

```

```

      VSTATISTICS[ ]V
V STATISTICS R;XBAR;VAR;N;Q;S;Z
[1] XBAR←(+/R)÷N+p,R
[2] VAR←(+/(R-XBAR)*2))÷N-1
[3] S←R[AR]
[4] Q←0.5×S[0.25×3+N×13]+S[0.25×1+N×13]
[5] 'MEAN: ';XBAR
[6] 'VARIANCE: ';VAR
[7] 'STD. DEV.: ';S+VAR*0.5
[8] 'COEFF. OF VARIATION: ';S÷|XBAR
[9] 'LOWER QUARTILE: ';Q[1]
[10] 'UPPER QUARTILE: ';Q[3]
[11] 'MEDIAN: ';Q[2]
[12] 'TRIMEAN: ';0.25×Q[1]+Q[3]+2×Q[2]
[13] 'MIDMEAN: ';(÷pZ)×+/Z+((R≥Q[1])∧(R≤Q[3]))/R
[14] 'RANGE: ';(↑/R)∧L/R
[15] 'MIDRANGE: ';((↑/R)+L/R)÷2
[16] 'MEAN ABSOLUTE DEVIATION: ';(+/|R-Q[2])÷N
[17] 'INTERQUARTILE RANGE: ';Q[3]-Q[1]
[18] 'COEFF. OF SKEWNESS: ';((+/(R-XBAR)*3)÷N-1)÷S*3
[19] 'COEFF. OF KURTOSIS: ';(((+/(R-XBAR)*4)÷N-1)÷S*4)÷3
V

```



```

VSTEMLEAF[ ]V
V Z+STEMLEAF X;C;R;S;SI;I;J;F;A;W;L;WW;AA;XW
[1] C+10*1-[10*R+1E-20+(X[pX])*(X+X[A,X])[1]]÷SCALE
[2] SI+(1+3*+/(R*C)> 25 50)++/(pX)> 25 100
[3] X+[0.5+X*C*10*+/(SI= 2 3 6
[4] F++/(SI=19)/ 0.5 2 1 1 0.5 2 2 1 0.5
[5] A+20p'0123456789ABCDEFGHIJ',Z+' '
[6] I+F*[X[1]÷10*F
[7] XW+8+WW+WIDTH=4 .
[8] S+' ' [1+J+X[1]≥0]
[9] L+1+(|W+(X≤(10*I)+J*-1+10*F)/X)÷10*[|I
[10] AA+A[1+(10 10)τ|LI]
[11] +((pA[L])≤WW)/L1
[12] Z+Z,XWpS,AA,'|',(WW+A[L]),'+',(LIT(pA[L])÷WW),2p' '
[13] →L2
[14] L1:Z+Z,XWpS,AA,'|',A[L],XWp' '
[15] L2:I+I+F*1-(I=0)*X[1]<0
[16] →8*10<pX+(pW)+X
[17] Z+((L(pZ)÷XW),XW)pZ
V

```

```

VSUMSQ[ ]V
V SUMSQ;C;NUMBER;B;BLDF
[1] T++*Y
[2] TSS+(+/(+Y*2))-C+((+/T)*2)÷NUMBER++/N
[3] BLSS+(+/(+((B++/Y)*2)÷K))-C
[4] MSBL+BLSS÷BLDF+MAX-1
[5] →7*1(OPTION=2)
[6] BLSS+BLDF+0
[7] WTSS+TSS-BLSS+BTSS+(+/(T*2)÷N))-C
[8] MSR+BTSS÷K-1
[9] TDF+NUMBER-1
[10] F+MSR+MSE+WTSS÷EDF+TDF-BLDF+K-1
[11] BLOCK+(B÷K)÷YBAR+(+/T)÷NUMBER
[12] ' ANCOVA TABLE'
[13] ' SOURCE DF SS MS F'
[14] CH+' '
[15] ' TREATMENT, I5, F13.2, F11.2, F8.2' FMT(K-1), BTSS, MSR, F
[16] →(OPTION=1)/L5
[17] ' BLOCKS, I8, F13.2, F11.2, F8.2' FMT(MAX-1), BLSS, MSBL, (MSBL÷MSE
)
[18] L5: ' ERROR, I9, F13.2, F11.2' FMT EDF, WTSS, MSE
[19] ' TOTAL, I9, F13.2' FMT TDF, TSS
[20] ' RES-SQUARE = , F5.3' FMT(BTSS+BLSS)÷TSS
[21] ' OVERALL MEAN = , F10.2' FMT YBAR
[22] ' TREATMENT EFFECTS , F6.2' FMT(T÷N)÷YBAR
[23] →(OPTION=1)/0
[24] ' BLOCK EFFECTS , F6.2' FMT BLOCK
V

```



```

      VTWICE[ ]V
V  Z←TWICE X;Y
[1] Z←Y+A3RSR X-Y+A3RSR X
V

```

```

      VUTCOND[ ]V
V  UTCOND X
[1] X←(2p(pX),1)pX
[2] Z←''
[3] I←0×NR+(pX)[2]
[4] N←pU+U[1]U←(U≠MISS)/U+X[;I+1]
[5] Q←0.5×U[10.125×7+N×17]+U[10.125×1+N×17]
[6] →7+2×[NUM÷2]
[7] Z←Z,Q[4]
[8] →4+11×NR=I+I+1
[9] Z←Z,Q[4],Q[6]~Q[2]
[10] →4×NR>I+I+1
[11] Z←Z,U[1],Q[2],Q[4],Q[6],U[N],N
[12] →4+13×NR=I+I+1
[13] Z←Z,U[1],Q[1 2 4 6 7],U[N],N
[14] →4+16×NR=I+I+1
[15] CH←'MEDIAN'
[16] →0×p[ ]←'F10.2' FMT Z←(NR,1)pZ
[17] Z←(NR,6)pZ
[18] CH←'MIN,LOQ,MEDIAN,UPQ,MAX,SIZE'
[19] →0×p[ ]←'F10.2' FMT Z
[20] Z←(NR,8)pZ
[21] CH←'MIN,LO8,LOQ,MEDIAN,UPQ,UP8,MAX,SIZE'
[22] [ ]←'F9.2' FMT Z
V

```

REFERENCES

1. APL(CMS)--An Introduction, Naval Postgraduate School Technical Note No. 0141-33, August 1977.
2. IBM Manual GH20-0906-1, APL/360-OS and APL/360-DOS User's Manual.
3. Gilman, L. and Rose, A.J., APL-An Interactive Approach, John Wiley and Sons, Inc., New York, 1974.
4. Katzan, H., APL Programming and Computer Techniques, Van Nostrand Reinhold Publishers, New York, 1970.
5. McNeil, D.R., Interactive Data Analysis: A Practical Primer, Wiley Interscience, New York, 1977.
6. McNeil, D.R., Interactive Exploratory Data Analysis Using APL, Princeton University Technical Report No. 54, Series 2, 1974.
7. Mosteller, F. and Tukey, J.W., Data Analysis and Regression: A Second Course in Statistics, Addison-Wesley Publishing Co., Reading, Massachusetts, 1977.
8. Tukey, John W., Exploratory Data Analysis, Addison-Wesley Publishing Co., Reading, Massachusetts, 1977.

DISTRIBUTION LIST

	NO. OF COPIES
Defense Documentation Center (DDC) Cameron Station Alexandria, VA 22314	2
Dean of Research, Code 012A Naval Postgraduate School Monterey, CA 93940	1
Library, Code 0212 Naval Postgraduate School Monterey, CA 93940	1
Library, Code 55 Naval Postgraduate School Monterey, CA 93940	1
Dean J. R. Borsting Naval Postgraduate School Monterey, CA 93940	1
Dean D. A. Schradly, Code 013 Naval Postgraduate School Monterey, CA 93940	1
Professor D. R. Barr, Code 55Bn	1
Professor D. P. Gaver, Code 55Gv	1
Professor J. K. Hartman, Code 55Hh	1
Professor G. T. Howard, Code 55Hk	1
Professor P. A. Jacobs, Code 55Jc	1
Professor H. J. Larson, Code 55La	1
Professor P. A. W. Lewis, Code 55Lw	1
Professor A. W. McMasters, Code 55Mg	1
Professor P. R. Milch, Code 55Mh	1
Professor G. K. Poock, Code 55Pk	1
Professor R. R. Read, Code 55Re	1
Professor M. G. Sovereign, Code 55Zo	1
Professor Q. W. Washburn, Code 55Ws Naval Postgraduate School Monterey, CA 93940	1
Professor P. M. Carrick, Code 54Ca	1
Professor C. R. Jones, Code 54Js	1
Professor R. A. Weitzman, Code 54Wz Naval Postgraduate School Monterey, CA 93940	1

DISTRIBUTION LIST CONT.

	NO. OF COPIES
Professor Toke Jayachandran, Code 53Jy Naval Postgraduate School Monterey, CA 93940	1
Professor D. G. Williams, Code 0141 Naval Postgraduate School Monterey, CA 93940	1
Computer Center Library, Code 0141 Naval Postgraduate School Monterey, CA 93940	2
R. J. Stampfel Code 55 Naval Postgraduate School Monterey, CA 93940	1
Professor F. R. Richards, Code 55Rh Naval Postgraduate School Monterey, CA 93940	100