HEWLETT-PACKARD

Basic Statistics & Data Manipulation Pac

Series 80

Series 80
Basic Statistics and Data Manipulation Pac

October 1982

Reorder Number
00085-90620

Printed in U.S.A.

Introduction

The Basic Statistics and Data Manipulation Pac (BSDM) was developed for three major purposes. First, it provides a common data base which may be accessed by advanced HP-85 statistical pacs (General Statistics excluded). So, if you want to use more than one statistical routine on your data, you won't have to key in the numbers again and again.

Second, in almost any type of statistical analysis there are procedures which are used to condition or "massage" the original data. There are also many common I/O operations involving the data. BSDM takes care of these procedures all at once for all pacs. I/O operations such as entering and listing the data are done in BSDM instead of in each statistical routine. Similarly, algebraic transformations, recoding, sorting and data editing are offered by BSDM. So, instead of learning the formats of data manipulation routines in each statistics package, you learn it once.

Third, BSDM provides insight into your data by computing a wide assortment of summary statistics. These allow you to understand the characteristics of the data and to summarize that information.

Hewlett-Packard would like to acknowledge the work of Thomas J. Boardman, Ph. D., Statistical Laboratory, Colorado State University, in the development of this pac.

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Contents

Program Operation Hints

These programs have been designed to execute with a minimum amount of difficulty, but problems may occur which you can easily solve during program operation. There are four different types of errors or warnings that can occur while executing a program; input errors, math errors, tape errors and image format string errors.

The input errors include errors 43, 44, and 45. These errors will cause a message to be output followed by a new question mark as a prompt for the input. You should verify your mistake and then enter the correct input. The program will not proceed until the input is acceptable. There is a complete discussion of INPUT in your Owner's Manual if you need more detail.

The second type of error which might occur is a math error (1 thru 13). With DEFAULT ON, the first eight errors listed in Appendix E of your Owner's Manual cause a warning message to be output, but program execution will not be halted. The cause of these errors can usually be attributed to specific characteristics of your data and the type of calculations being performed. In most cases, there is no cause for alarm, but you should direct your attention to a possible problem. An example of such a case is found in the Standard Pac when the curve fitting program computes a curve fit to your data which has a value of 1 for the coefficient of determination, r^2. The computation of the F ratio results in a divide by zero, Warning 8.

The third type of error, tape errors (60 thru 75) may be due to several different problems. Some of the most likely causes are the tape being write-protected, the wrong cartridge (or no cartridge) being inserted, a bad tape cartridge, or wrong data file name specification during program execution. Appendix E of your Owner's Manual should be consulted for a complete listing.

The fourth type of error is due to generalizing the output to anticipated data ranges. In many cases, the output has assumed ranges which may or may not be appropriate with your data. Adjusting the image format string for your data will solve this type of problem. You may also want to change the image string if you require more digits to the right of the decimal point.

These are the more common problems which may occur during program operation. Your Owner's Manual should be consulted if you need more assistance.

Two versions of the program have been designed to run specifically on either a tape or disc. The operation of the disc version is explained in Appendix D of this manual.

Notes

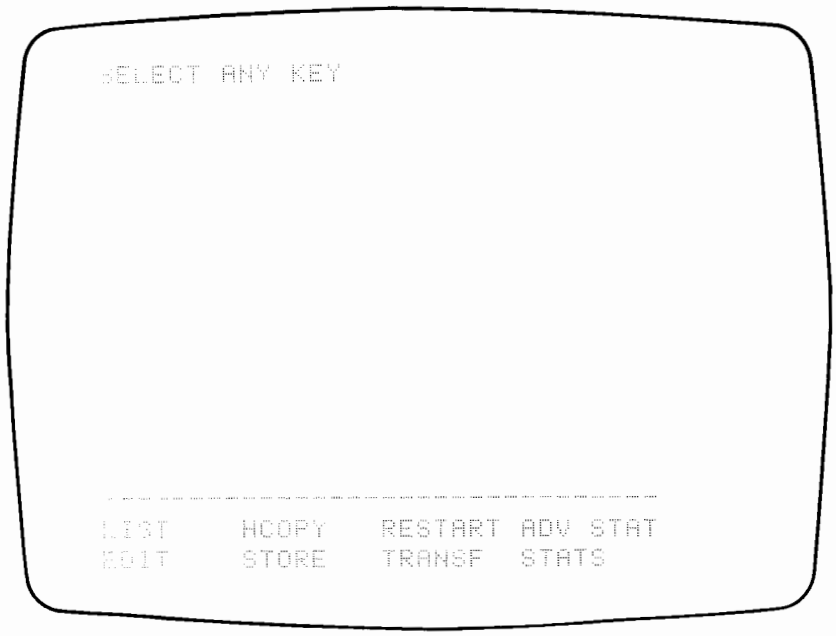
Program Usage

General

The Basic Statistics and Data Manipulation Pac (BSDM) contains an autostart program, Autost, so that you can start by inserting the cartridge and turning the machine on. BSDM is broken into modules, each of which performs specific functions. These modules are: input of data, output of data, editing, transformations, and basic statistics.

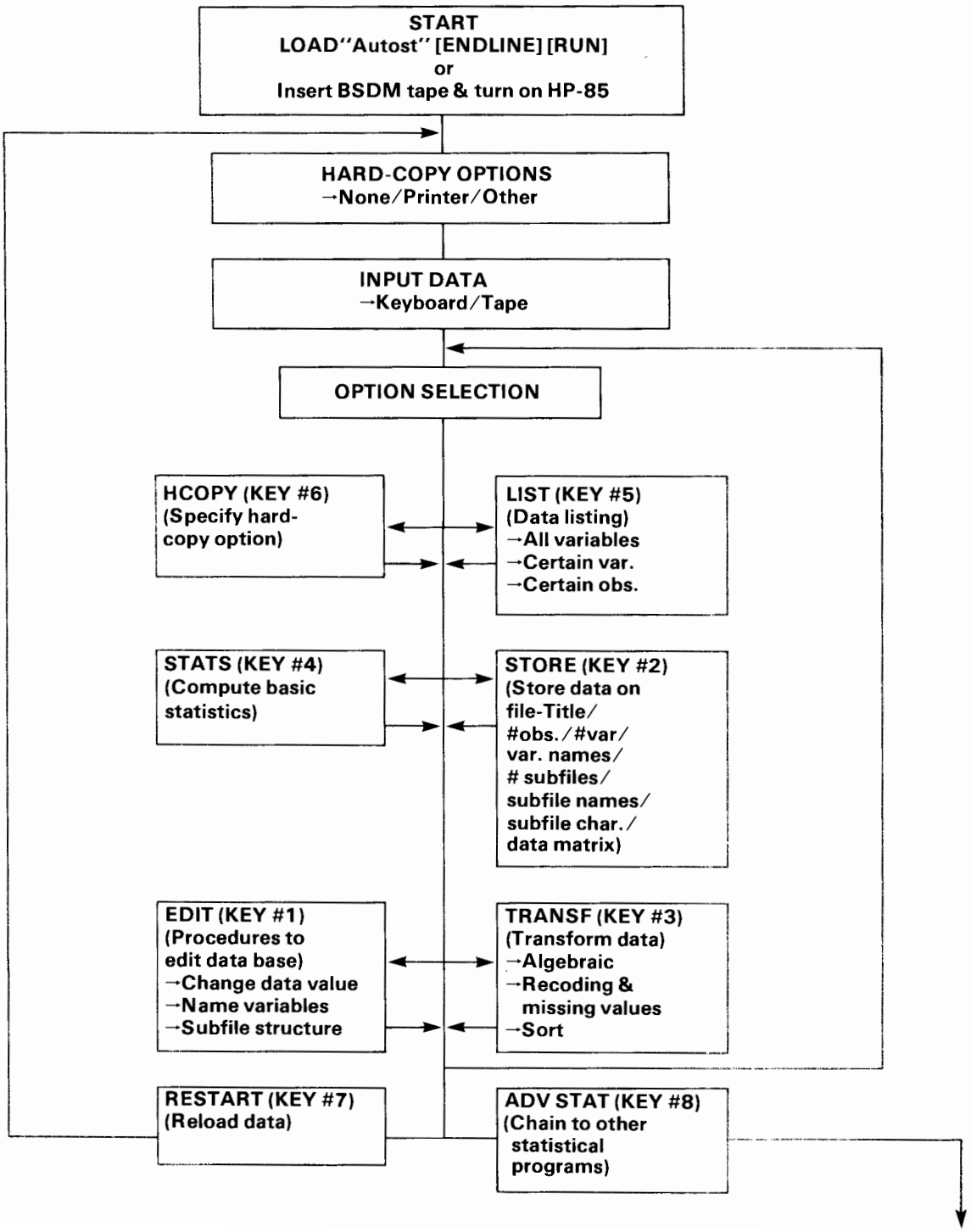
This set of programs allows you to enter a data matrix into memory and to then perform various operations on the data. The data entry may be made via keyboard or tape cartridge. The operations on the data set include editing (i.e., data editing, naming and creating subfiles), transforming (i.e., algebraic transformations, recoding and sorting), storing and listing.

Features include a provision for missing data values, a provision for incorporating a subfile structure, the ability to store the data matrix and related information, error detection and the ability to correct many possible errors.



After data has been entered, the above keylabels show the commands that are available. The programs have been designed to run using the internal tape cartridge. If you have purchased the Mass Storage ROM, you may want to change the programs in this pac to take advantage of the disc-based mass storage system. Refer to the Mass Storage ROM Manual to obtain help in translating programs.

Program Flow



Input of Data

The data matrix incorporated in this program should be thought of as a p-by-n array whose columns correspond to observations and whose rows correspond to variables as shown below.

	Observations				
Variables	O_1	O_2	O_3	...	O_n
V_1				...	
V_2				...	
V_3				...	
.	.	.	.		.
.	.	.	.		.
.	.	.	.		.
V_p				...	

Subfiles may be created, in which case the structure becomes only slightly more complex as shown below.

Variables	**Subfile 1** $O_1\ O_2 \dots O_{n_1}$	**Subfile 2** $O_{n_1+1}\ O_{n_1+2} \dots O_{n_1+n_2}$	...	**Subfile S** $O_{n_1+\dots+n_{s-1}+1} \dots O_{n_1+\dots+n_s}$
V_1				
V_2				
.				
.				
.				
V_p				

The Basic Statistics and Data Manipulation routines are capable of handling a maximum of 12 variables and 500 data values. For example, for a two-variable data set, up to 250 observations can be entered. For a four-variable data set, up to 125 observations can be used. A maximum of ten subfiles can be specified. More data values can be used if you have the 16K-byte additional memory module. (Refer to Appendix to change # of variables.)

Data can be entered either from the keyboard or from a data file. The data file could either be the scratch file used by BSDM, (DATA), or a user created file on the tape cartridge.

Special Considerations

- The prompts concerning the data medium and program medium may cause confusion. The word "medium" is used since the set of programs making up the "BASIC STATISTICS AND DATA MANIPULATION" package may be on tape, floppy disc or hard disc. Thus, the "program medium" refers to either the cartridge or the disc on which the programs making up this package are stored. Conversely, the "data medium" refers either to the cartridge or to the disc on which the file containing the data matrix resides. In some cases, the program medium and the data medium may be the same. However, this cannot be determined by the program and hence, the prompts are still displayed to make sure the correct medium is in the correct device.

- If the data is on a mass storage device, it may have gotten there in one of four ways. The following discussion explains the prompts that apply to each situation.

  1. If the data was entered using this statistics package (and was the last data set used on this package), it will be on the cartridge in the scratch file called "DATA". Thus, an affirmative answer to the prompt "Is data stored on the program medium's scratch file (DATA)?" will retrieve the data and related information.

  2. The data may have been entered using the Basic Statistics and Data Manipulation (BSDM) routines from this or any other statistics package and then stored using the STORE routine of BSDM. In this case, you should answer Yes to the prompt "Was data stored by this program?". Then, after the file name is specified the data and related information will be retrieved.

     The data may have been acquired and placed in a file by a process unrelated to the BSDM routines. The following two possibilities exist.

  3. The data may be stored as: all observations of variable one followed by all observations of variable two, etc. This is in the same configuration as data stored by the BSDM routines, i.e., variables = rows and observations = columns. To retrieve the data, a Yes response to the prompt "Is the data in proper configuration...?" should be given.

  4. The data may be stored as: all variables of observation one followed by all variables of observation two, etc. This is the transpose of what is expected by the BSDM routines, i.e., observations = rows, variables = columns. To retrieve this type of data a Yes response should be given to the prompt "Data stored as contiguous array with observations = rows...?".

Missing Values

The number used to designate a missing value is -999999.9999. The justification for this number is that (besides seeming unlikely to occur as a legitimate data point) it is easily picked out in a listing of the data. It may be more desirable to designate a missing value by a more easily typed number, for example, by 0 if zero is not a legitimate data point. The zeros could then be converted to the missing value recognized by the programs. This may be accomplished in two ways. First, during the input procedure, answer "No" to the prompt, "Are missing values denoted by -999999.9999?" Then input the value used to denote missing values. Or, in the second case, the Transformation program contains a missing value option. A value denoting a missing value can be specified for any or all variables. Any observation having the denoted value is transformed to the value -999999.9999.

Incorrect Responses

If a response outside the range of plausible responses is input from the keyboard, a message so stating is displayed for about three seconds. Program execution is resumed by asking the question or a previous question again.

If a plausible response is given, but it is not correct, one of three possibilities exists. First, if an incorrect value has been entered for a data point, it may be corrected using the EDIT routine. Second, in many cases, responses to several questions are printed on the CRT and then a question such as "Is the above information correct?" is asked. This allows any of the printed information to be changed. Third, if a YES/NO question is answered incorrectly or if the above options are not offered, the program can be re-started by pressing KEY LABEL, then the soft key for the program you are in.

Output of Data

Once the data has been entered, it can be output to a file or to a printer/CRT.

Store

The STORE feature allows you to store the data matrix and related information in a file so that it may be retrieved at a later date for further analysis. The program also allows you to specify the file name.

The store feature will be useful in two different situations. First, if an investigator has a data set which he may want to analyze further at a later date, he may store it and retrieve it later via the Basic Statistics And Data Manipulation Pac. Secondly, if several people have access to the data input programs, it becomes mandatory that each be able to store his data set in a unique place. Note that if only one person uses the routine on one data set, it is unnecessary to use the store feature since the data and related information are kept in "DATA"—the scratch file on the program medium.

The existence of a file is checked in the program in an attempt to avoid the accidental loss of existing data. Thus, when a file is specified to receive the data, an attempt is made to ensure that you are not accidentally storing the new data in a file which you did not know existed.

List

The program allows you to obtain a listing of the data matrix. The listing will appear on the device that has been specified for hard-copy in the START routine or in the HCOPY routine. You can list all the data, a certain set of variables, or a certain set of observations.

If the number of variables to be listed exceeds 4, the data will be output in groups of four variables at a time. This was done to make the output more readable.

Hcopy

This routine allows you to change the device on which the hard-copy output will be printed, or conversely, to specify that no hard-copy is desired.

Editing

Once the data has been entered, this routine allows you to correct any errors made during input. The editing functions can be broken into three categories: changing data values, naming variables and changing subfile specifications.

Change Data Values

This routine is designed to allow you to perform a variety of editing procedures on the data matrix. The editing capabilities include: the changing of an incorrect data value, the deletion of a variable, the deletion of an observation, the addition of an observation, the insertion of an observation (if the data is ordered), and the addition of a variable. All of the above operations can be performed repeatedly; for example, three variables could be deleted in succession.

Special Considerations

Order of Corrections

As stated in the program note displayed on the screen, the data is renumbered after deletions or insertions are performed. For this reason, if more than one deletion (insertion) is to be performed, it is recommended that the highest-numbered observation (or variable) be deleted, then the next highest-numbered, etc. For example, if observations three and eight are to be deleted, then it is recommended to delete observation eight first, then observation three. Notice that if observation three were deleted first, the subsequent renumbering would move observation eight to position seven. The recommendation is meant to alleviate confusion which may occur due to the renumbering.

Subfiles

Insertions or deletions of observations will affect the content of subfiles which are in existence at the time of editing; for example, if subfile one consists of the first 10 observations while subfile two consists of the last 20 and if observation five is deleted, then observation ten (formerly numbered 11) will have jumped from subfile two to subfile one. Thus, it may be necessary to change the subfile structure after editing. Hence, it is recommended that subfiles be created only after all editing has been performed.

Correcting a Data Value

When correcting a data value, you must specify the variable number and observation number of the value to be corrected. Then, the old value is displayed so you can be sure you are altering the correct value.

Inserting an Observation

If an observation is to be inserted, the position of the insertion must be specified by entering the number of the observation which the insertion will precede. For example, if an observation is to be inserted between observation 8 and 9, you must enter 9 when the prompt "Insertion to percede observation #?" is displayed.

Exceeding Program Limitations

If the addition of an observation or of a variable will exceed program limitations, these options will not be executed.

Methods and Formulae

The data matrix is redimensioned into a row vector to facilitate the shuffling of elements necessitated by the editing operations. The vector contains all the observations of variable one, followed by the observations of variable two, etc. When an observation is inserted, for example, the elements of the data vector are shuffled one at a time to make room for the incoming observation. Similarly, when an observation is deleted, the remaining observations are "packed" together so that the resultant data vector has no "holes" between observations.

Name

This routine allows you to rename the data set, to rename variables and/or to rename subfiles. These names are then stored, along with the data, on the program medium's scratch file ("DATA"). You may change a single variable or subfile name, or you may change a set of successive names.

Subfile Structure

This routine allows you to specify subfiles or logical groupings of the observations. This may be accomplished by entering the number of observations in each subfile or by entering the observation number of the first observation in each subfile. Names for the subfiles are entered in both cases. A third option allows you to destroy the existing subfile structure.

Special Considerations

Use of Subfiles

Subfiles may be created in order to specify logical groupings of observations. A subfile structure allows you to consider each subfile as a separate data set or to lump all the subfiles together and analyze the overall data set. For example, suppose an investigator wished to measure several variables on 50 trout. He would like to analyze the data separately for each of the three species of the trout. He could form three separate data sets and do the individual analyses, then later join the three sets together for the overall analysis. However, since the same variables were measured on each species of fish, this situation is well-handled by specifying a subfile for each variety. The subfile structure options make it possible to do the analysis by subfile as well as for the overall data set.

Editing and Sorting

Certain operations in the data editing and sorting sections may cause observations to move from one subfile to another. To avoid undesired results such as this, it is recommended that subfiles be specified after any editing or sorting has been carried out.

Transformations

In many analyses, the statistician prefers to work with some transformation of the original data. The transformation routines provide great flexibility in this area. Four major categories are addressed: algebraic transformations, missing value assignment, recoding and sorting.

Algebraic Transformations

This routine allows you to transform one or two variables in the data matrix via pre-specified functions or through a function which you specify. The transformed data may then be treated as a new variable, or it may replace the elements of an existing variable. Hence, transformations on more than two variables may be made iteratively or via a user-defined transformation.

Special Considerations

Transformations Available

1. a*X b +c
2. a*X b + Y c
3. aX + bY + c
4. a*log bX + c
5. a*log bX + cY
6. standardize variable
7. user defined

The log function in transformations (4) and (5) refers to natural logarithm of a positive X to the base e (2.71828182846). Thus, note the argument of the log function (bX or bX + cY depending on the transformation) must be positive. If not, an error message will appear and the user should try a different transformation. Many users may wish to use the log transformations to the base 10. If so, simply divide the number you wish to input for "a" by the natural log of 10(LOG10).

Missing Values

None of the pre-specified transformations are applied to missing values. Thus, missing values are unaffected by these transformations. However, this is not necessarily the case with the user-defined transformation. If you define a transformation and there are missing values, you must make provisions to ensure that the transformation is not applied to the missing values (unless, of course, this is desired). This may be accomplished as explained in the following section, "User-Defined Transformation."

User-Defined Transformation

You may use up to 10 lines of code to define your own transformation—namely lines 1440 through 1449 in the program TRANS1. The transformation should be of the following form:

D(Z,I) = 'user-defined transformation'

Remember that with the user-defined transformation, the user is responsible for dealing with missing values. Consider the following example. Suppose your data set consists of four variables. There are missing values. You desire to form variable five as the sum of the exponentials of variables one and three. If there is a missing value in either of these variables, you wish to assign a missing value to the transformed variable. Recall that the data is of the form D(I,J) where I is the variable number and J is the observation number. In the transformation routine the variable I is used to denote the variable where the transformed data is to be stored. Thus, to accomplish the above-described transformation, follow this sequence:

1. Type: (LOAD) "TRANS1"
2. Press: (END LINE)
3. Type: 1440 IF D(1,I)<>-999999.9999 AND D(3,I)<>-999999.9999 THEN 1443
4. Press: (END LINE)
5. Type: 1441 D(5,I) =-999999.9999
6. Press: (END LINE)
7. Type: 1442 GOTO 1450
8. Press: (END LINE)
9. Type: 1443 D(5,I) = EXP(D(1,I)) + EXP(D(3,I))
10. Press: (END LINE)
11. Type: 1444 GOTO 1450
12. Press: (END LINE)
13. Type: (STORE) "TRANS1"
14. Press: (END LINE)

The binary program REDZER is not needed in memory when this is done.

Now the user-defined transformation has been stored in the TRANS1 file. If you now run the BSDM pac and ask for a user-defined transformation, this is the one you will get.

Notice that some of the lines 1440-1449 may have been defined by a previous user-defined transformation. **Make sure they do not interfere with your transformation.**

Recoding and Missing Values

This routine program allows you to assign codes to various categories or classes of data or to assign missing values. The categories are intervals along the real number line and 20 of these may be specified. The recoding is done on one variable at a time. The same coding scheme may be used iteratively on successive variables. A summary of the coding intervals, codes, and number of observations assigned to each code is printed as hard copy.

Special Considerations

Coding Schemes

Three coding schemes are available, contiguous—equal intervals, contiguous—unequal intervals, and missing value recoding. If the coding intervals are all of the same length and are contiguous, that is, together they form a connected interval, then the interval construction can be accomplished internally knowing only the interval length and lower limit for the first interval. Similarly, if the intervals are of unequal lengths but contiguous, for example, [10, 20), [20, 25), [25, 40), [40, 70), then the lower limit of each interval needs to be specified but the upper limit may be computed internally. Hence, these two coding schemes are meant to minimize the amount of information which needs to be entered from the keyboard. The missing value coding scheme asks you for a variable number and the numeric value used to designate missing values for that variable. Thus, each variable can have its own numeric value to denote missing values. These missing values will be recoded to the value -999999.9999.

Same Coding Scheme

The coding is carried out on one variable at a time. However, if you desire to code both variables one and two according to the same coding intervals, these intervals need to be constructed only once. A positive response to the option offered by "Use same coding scheme?" allows variable two to be coded according to the same scheme without constructing the intervals a second time. If, however, you desire to code variable two according to a different scheme, it is possible to construct a second set of coding intervals by giving a negative response to the above prompt.

Brackets

The brackets used to denote the coding intervals are meant to follow their usual mathematical interpretation, that is, the intervals are closed on the left and open on the right.

Observation not in an Interval

If an observation does not fall into any of the coding intervals, its value is not changed during the coding process.

Sort

This routine allows the data matrix, or subfiles thereof, to be sorted according to the values of one variable. For example, suppose an investigator has five observations of three variables, say height, weight and age and wanted to arrange the observations in ascending order according to age. This is accomplished by sorting the data matrix according to variable three.

Special Considerations

Subfile Structure Options

If subfiles are ignored, the entire data set will be sorted and, in the process, the composition of the subfiles is subject to change. The option of sorting certain subfiles may be used to sort a single subfile or a set of successive subfiles according to one variable. The option of sorting all subfiles may be used to sort every subfile. The options of sorting certain subfiles and sorting all subfiles treat each subfile as if it were a separate data set. Thus, the sort is done with respect to one subfile at a time.

What Happens

It is important to note that entire observations are moved when the sort is carried out. Thus, referring to the example given, a person's height and weight remain with the person's age as shown below.

Original Matrix

VARIABLE

Observation	Height	Weight	Age
1	72	170	21
2	70	165	25
3	69	150	20
4	70	165	25
5	73	160	19

Matrix Sorted By Age

VARIABLE

Observation	Height	Weight	Age
1	73	100	19
2	69	150	20
3	72	170	21
4	70	165	25
5	70	165	25

Basic Statistics

This routine computes a variety of summary statistics for data which was entered via the BASIC STATISTICS AND DATA MANIPULATION program set. The statistics may be computed by subfile or for the entire data set (ignoring subfiles). Basic statistics which are computed include: number of observations, number of missing values, means, standard deviation, coefficient of skewness, coefficient of kurtosis, coefficient of variation, standard error of the mean, and confidence interval of the mean. An option is available to obtain the correlation matrix. Order statistics computed include: the maximum, minimum, range, midrange, median, 25^{th} percentile, and 75^{th} percentile.

Special Considerations

Hard-copy Output

If a hard-copy of the statistics is not being made, the program halts occasionally so that you may study the results. In this case, it is necessary only to press [CONT] to continue program execution.

Order Statistics

To obtain the order statistics, the data matrix is sorted and the observations of each variable are arranged in ascending order. At the end of the program the original data matrix is re-loaded. Thus, if the program is aborted, that is, if another key is pressed before the re-loading can occur, the data matrix will be in the sorted state. Hence, if the portion of the program used to calculate additional order statistics is accessed, abortion of the program should be discouraged.

Methods and Formulae

Let N(I) be the number of observations of the I^{th} variable in the data set or subfile, whichever is applicable. Let D(I, J) be the J^{th} observations of the I^{th} variable. The following formulas are computed for the I^{th} variable.

- Sum: $S(I) = \sum_{J=1}^{N(I)} D(I, J)$

- Mean: $M(I) = \frac{S(I)}{N(I)}$
- Variance: $V(I) = \frac{\sum_{J=1}^{N(I)} (D(I, J))^2 - N(I)(M(I))^2}{N(I) - 1}$
- Standard deviation: $Sd(I) = (V(I))^{1/2}$
- Second moment: $M_o(I) = \frac{(N(I) - 1)\, V(I)}{N(I)}$
- Skewness: $Sk(I) = \frac{\sum_{J=1}^{N(I)} (D(I, J))^3 - 3M(I) \sum_{J=1}^{N(I)} (D(I, J))^2 + (2N(I)M(I)^3)}{(M_o(I))^{3/2}\, N(I)}$
- Kurtosis: $K(I) = \frac{\sum_{J=1}^{N(I)} (D(I, J))^4 - 4M(I) \sum_{J=1}^{N(I)} (D(I, J))^3 + 6(M(I))^2 \sum_{J=1}^{N(I)} (D(I, J))^2 - 3(M(I))^4\, N(I)}{(M_o(I))^2\, N(I)} - 3$

- Confidence interval on the mean: $M(I) \pm T(SE(I))$ where the T value is computed as follows:

Let C be the confidence coefficient for a confidence interval on the mean. The following operations are used to obtain the desired t-value.

$$P = \frac{1 - \frac{C}{100}}{2}$$

$$V = \left(\ln\left(\frac{1}{P^2}\right)\right)^{1/2}$$

$$X = 2.5155174 + .802853V + .010328V^2$$

$$Y = 1 + 1.432788V + .189269V^2 + .001308V^3$$

$$Z = V - \frac{X}{Y}$$

$$M = N(I) - 1$$

Then the desired t-value is:

$$T = Z + \frac{Z^3 + Z}{4M} + \frac{5Z^5 + 16Z^3 + 3Z}{96M^2} + \frac{3Z^7 + 19Z^5 + 17Z^3 - 15Z}{384M^3} + \frac{79Z^9 + 776Z^7 + 1482Z^5 - 1920Z^3 - 945Z}{92160M^4}$$

- Standard error: $Se(I) = \frac{(V(I))^{1/2}}{(N(I))^{1/2}}$
- Coefficient of variation: $Cv(I) = \left| \frac{(V(I))^{1/2}}{(M(I))} \right| (100)$
- Correlations: Suppose we have the following data matrix:

	Observation				
Variable	**1**	**2**	**3**	**4**	**5**
1	5	M	3	4	5
2	6	7	M	6	4
3	1	3	2	1	1

An M denotes a missing value. When computing the correlation between variables 1 and 2, we discard observations 2 and 3 since variable 1 is missing a data value for observation 2 and variable 2 is missing the data value for observation 3. However, when computing the correlation between variables 1 and 3, we need only discard observation 2. Similarly, the correlation between 2 and 3 is computed by discarding only observation 3. Hence, the correlations may be based on different numbers of observations. An observation is thrown out if and only if a data value from that observation is missing from one of the two variables for which the correlation is being computed. With this in mind, let N(I, J) be the number of observations used to compute the correlation between variables I and J. Then, the correlation is:

$$C(I,J) = \frac{\sum_{K=1}^{N(I,J)} D(I,K)D(J,K) - \frac{\sum_{K=1}^{N(I,J)} D(I,K) \sum_{K=1}^{N(I,J)} D(J,K)}{N(I,J)}}{\left[\sum_{K=1}^{N(I,J)} (D(I,K))^2 - \frac{\left(\sum_{K=1}^{N(I,J)} D(I,K)\right)^2}{N(I,J)}\right]^{1/2} \left[\sum_{K=1}^{N(I,J)} (D(J,K))^2 - \frac{\left(\sum_{K=1}^{N(I,J)} D(J,K)\right)^2}{N(I,J)}\right]^{1/2}}$$

Ranges and Percentiles

Let M(I) be the largest data value of the I^{th} variable, m(I) be the smallest data value of the I^{th} variable.

1. Range: R(I) = M(I) – m(I)
2. Midrange: $Mr(I) = \frac{M(I) + m(I)}{2}$

The percentiles are computed as follows: Let P be the percentile in question. If N(I)*P/100 is an integer, the P(I) = (D(I, N(I)*P/100) + D(I, Q))/2, where Q is the next integer value between N(I)*P/100 and the observation index of the median. If N(I)*P/100 is not an integer, the P(I) = D(I, N(I)*P/100 + Q) where

$$Q = \begin{cases} 1 \text{ if } P \leqslant 50 \\ -1 \text{ if } P < 50 \end{cases}$$

The median refers to the 50^{th} percentile.

Example

Here is a series of examples showing the output from each of the procedures.

```
**********************************
*                                *
*       DATA MANIPULATION        *
*                                *
**********************************

         PLANT OUTPUT DATA
Number of obs:  17
Number of variables:   5
Variable names:
    1.   TEMP.
    2.   PROD.
    3.   DAYS
    4.   # PEO.
    5.   H20USE



            TEMP.          PROD.
               DAYS           # PEO.
            H20USE
  1         14.9000     6396.0000
               21.0000      134.0000
         3373.0000
  2         18.4000     5736.0000
               22.0000      146.0000
         3110.0000
  3         21.6000     6116.0000
               22.0000      158.0000
         3180.0000
  4         25.2000     8287.0000
               20.0000      171.0000
         3293.0000
  5         26.3000    13313.0000
               25.0000      198.0000
         3390.0000
  6         27.2000    13108.0000
               23.0000      194.0000
         4287.0000
  7         22.2000    10768.0000
               20.0000      180.0000
         3852.0000
  8         17.7000    12173.0000
               23.0000      191.0000
         3366.0000
  9         12.5000    11390.0000
               20.0000      195.0000
         3532.0000
 10          6.9000    12707.0000
               20.0000      192.0000
         3614.0000
```

This example has 17 observations with the same 5 variables measured on each observation.

The initial names for the five variables were specified here. Note that the variable names must be six or fewer characters.

As the data is entered by variable for each observation, the values are printed. The sequence of variables for each observation is 1(TEMP.), 2(PROD.), 3(DAYS), 4(# PEO.), and 5(H2OUSE).

```
11         6.4000    15022.0000
              22.0000       200.0000
         3896.0000
12        13.3000    13114.0000
              19.0000       211.0000
         3437.0000
13        18.2000    12257.0000
              22.0000       203.0000
         3324.0000
14        22.8000    13118.0000
              22.0000       197.0000
         3214.0000
15        26.1000    13100.0000
              21.0000       196.0000
         4345.0000
16        26.3000    16716.0000
              21.0000       205.0000
         4936.0000
17         4.2000    14056.0000
              22.0000       205.0000
         3624.0000



**********************************
*                                *
*        CURRENT NAMES           *
*                                *
**********************************

          VARIABLE NAMES

 1. TEMP.
 2. PROD.
 3. DAYS
 4. PEOPLE
 5. H2OUSE



**********************************
*                                *
*             STORE              *
*                                *
**********************************

Data and related information is
stored in MARK2.
```

Notice that we have changed the names for variable 4 and 5 at this point by using the EDIT key.

The data set is stored on **our data tape** on a file which we called MARK2, using the STORE key.

```
**********************************
*                                *
*         DATA EDITING           *
*                                *
**********************************

Obs.# 11 Var.# 2 -- correct
value =  15024

Obs.# 10 has been deleted,
 16 observations remain.

Obs.# 17 Var.# 1  =  4.2
Obs.# 17 Var.# 2  =  12707
Obs.# 17 Var.# 3  =  20
Obs.# 17 Var.# 4  =  192
Obs.# 17 Var.# 5  =  3614
Total number of observations
now =  17


**********************************
*          DATA LISTING          *
*          ON DATA SET:          *
*       PLANT OUTPUT DATA        *
**********************************


            TEMP.          PROD.
               DAYS           PEOPLE
OBS#
 1         14.9000     6396.0000
              21.0000      134.0000

 2         18.4000     5736.0000
              22.0000      146.0000

 3         21.6000     6116.0000
              22.0000      158.0000

 4         25.2000     8287.0000
              20.0000      171.0000

 5         26.3000    13313.0000
              25.0000      198.0000

 6         27.2000    13108.0000
              23.0000      194.0000

 7         22.2000    10768.0000
              20.0000      180.0000
```

We made a change to observation #11. The second variable (PROD.) which was 15022 is now changed to 15024. We then deleted observation number 10 entirely. We added a new observation which will be number 17 since we previously deleted number 10. This was accomplished using the EDIT key and selecting the "change data values" option.

This is a listing of the 'corrected' observation. Note that the fifth (and next three variables if present) are printed below the first four variables.

```
 8         17.7000    12173.0000
              23.0000       191.0000

 9         12.5000    11390.0000
              20.0000       195.0000

 10          6.4000    15024.0000
              22.0000       200.0000

 11         13.3000    13114.0000
              19.0000       211.0000

 12         18.2000    12257.0000
              22.0000       203.0000

 13         22.8000    13118.0000
              22.0000       197.0000

 14         26.1000    13100.0000
              21.0000       196.0000

 15         26.3000    16716.0000
              21.0000       205.0000

 16          4.2000    14056.0000
              22.0000       205.0000

 17          4.2000    12707.0000
              20.0000       192.0000
```


```
            H2OUSE
OBS#
 1       3373.0000

 2       3110.0000

 3       3180.0000

 4       3293.0000

 5       3390.0000

 6       4287.0000

 7       3852.0000

 8       3366.0000

 9       3532.0000
```

```
 10      3896.0000

 11      3437.0000

 12      3324.0000

 13      3214.0000

 14      4345.0000

 15      4936.0000

 16      3624.0000

 17      3614.0000



**********************************
*                                *
*       DATA MANIPULATION        *
*                                *
**********************************

          PLANT OUTPUT DATA
Data file name: MARK2
Number of obs: 17
Number of variables:  5
Variable names:
    1.  TEMP.
    2.  PROD.
    3.  DAYS
    4.  PEOPLE
    5.  H2OUSE

Subfiles:  NONE

**********************************
*                                *
*     DATA TRANSFORMATIONS       *
*                                *
**********************************

The following transformation was
performed: a*(X^b)+c
where X is Variable # 5
       a =   .2642
       b =   1
       c =   0
Transformed data is stored in
Variable # 5 (H2OUSE).
```

The original data is reloaded into the 85 using the RESTART key. We will use the original data at the beginning of each data manipulation operation. (With the corrected names for variables 4 and 5.)

We have chosen transformation No. 1, namely $Y = a*(X \wedge b)+c$

or

X_5 (new) $= .2642 * X_5$ (original).

This transformation converts water use in gallons to water use in litres. This was accomplished using the TRANSF key and selecting the "Algebraic Transformations" option.

```
**********************************
*          DATA LISTING          *
*          ON DATA SET:          *
*       PLANT OUTPUT DATA        *
**********************************


            H2OUSE
OBS#
 1        891.1466

 2        821.6620

 3        840.1560

 4        870.0106

 5        895.6380

 6       1132.6254

 7       1017.6984

 8        889.2972

 9        933.1544

 10        954.8188

 11       1029.3232

 12        908.0554

 13        878.2008

 14        849.1388

 15       1147.9490

 16       1304.0912

 17        957.4608
```

This listing is done only for new variable 5.

```
**********************************
*                                *
*       DATA MANIPULATION        *
*                                *
**********************************

         PLANT OUTPUT DATA
Data file name: MARK2
Number of obs: 17
Number of variables:  5
Variable names:
    1.  TEMP.
    2.  PROD.
    3.  DAYS
    4.  PEOPLE
    5.  H2OUSE

Subfiles:  NONE
```

Reload original data set to 85, using the RESTART key.

```
**********************************
*                                *
*             RECODE             *
*                                *
**********************************

Var.# 5 is recoded into 5
categories and the recoded
values are stored in Var.# 6
where:

                 CATEGORY BOUNDS
CELL#          LOWER          UPPER
   1.       3000.000       3400.000
   2.       3400.000       3800.000
   3.       3800.000       4200.000
   4.       4200.000       4600.000
   5.       4600.000       5000.000

CELL#         CODE      FREQ.
   1.        30.00         8
   2.        34.00         4
   3.        38.00         2
   4.        42.00         2
   5.        46.00         1
```

Suppose we wish to recode the original variable 5 (H_2O USE) into five cells with values 30, 34, 38, 42, and 46. The RECODE key allows us to specify our lower and upper bounds for each cell and then determine the number of observations within each cell. The new coded values were stored in variable 6. This was accomplished using the TRANSF key and selecting the "Recoding and Missing Value" operation.

```
**********************************
*          DATA LISTING          *
*          ON DATA SET:          *
*        PLANT OUTPUT DATA       *
**********************************
```

OBS#	H2OUSE	RECODE
1	3373.0000	30.0000
2	3110.0000	30.0000
3	3180.0000	30.0000
4	3293.0000	30.0000
5	3390.0000	30.0000
6	4287.0000	42.0000
7	3852.0000	38.0000
8	3366.0000	30.0000
9	3532.0000	34.0000
10	3614.0000	34.0000
11	3896.0000	38.0000
12	3437.0000	34.0000
13	3324.0000	30.0000
14	3214.0000	30.0000
15	4345.0000	42.0000
16	4936.0000	46.0000
17	3624.0000	34.0000

Variable 5 and recoded H2OUSE in Variable 6 are printed out.

```
**********************************
*                                *
*      DATA MANIPULATION         *
*                                *
**********************************

         PLANT OUTPUT DATA
Data file name: MARK2
Number of obs: 17
Number of variables:  5
Variable names:
    1.  TEMP.
    2.  PROD.
    3.  DAYS
    4.  PEOPLE
    5.  H2OUSE

Subfiles:  NONE

**********************************
*                                *
*              SORT              *
*                                *
**********************************

Data set: PLANT OUTPUT DATA
 has been arranged in order
 according to Variable #  2




**********************************
*          DATA LISTING          *
*          ON DATA SET:          *
*       PLANT OUTPUT DATA        *
**********************************


            TEMP.          PROD.
               DAYS           PEOPLE
OBS#
 1         18.4000     5736.0000
              22.0000      146.0000

 2         21.6000     6116.0000
              22.0000      158.0000

 3         14.9000     6396.0000
              21.0000      134.0000

 4         25.2000     8287.0000
              20.0000      171.0000
```

Reload original data set, using the RE-START key.

We have chosen to rearrange the data set by sorting from smallest production to largest production in variable two. This was accomplished using TRANSF key and selecting the "Sort" operation.

The sorted data set is printed out for all five variables. Variables 1 through 4 are printed to the left. The last variable is printed out below. If 8 variables had been used the first four would be printed out followed by the next four.

5	22.2000	10768.0000	
	20.0000	180.0000	
6	12.5000	11390.0000	
	20.0000	195.0000	
7	17.7000	12173.0000	
	23.0000	191.0000	
8	18.2000	12257.0000	
	22.0000	203.0000	
9	6.9000	12707.0000	
	20.0000	192.0000	
10	26.1000	13100.0000	
	21.0000	196.0000	
11	27.2000	13108.0000	
	23.0000	194.0000	
12	13.3000	13114.0000	
	19.0000	211.0000	
13	22.8000	13118.0000	
	22.0000	197.0000	
14	26.3000	13313.0000	
	25.0000	198.0000	
15	4.2000	14056.0000	
	22.0000	205.0000	
16	6.4000	15022.0000	
	22.0000	200.0000	
17	26.3000	16716.0000	
	21.0000	205.0000	

OBS#	H2OUSE
1	3110.0000
2	3180.0000
3	3373.0000
4	3293.0000
5	3852.0000

```
 6      3532.0000

 7      3366.0000

 8      3324.0000

 9      3614.0000

 10      4345.0000

 11      4287.0000

 12      3437.0000

 13      3214.0000

 14      3390.0000

 15      3624.0000

 16      3896.0000

 17      4936.0000


**********************************
*                                *
*       SUBFILE STRUCTURE        *
*                                *
**********************************

  Subfile    1st obs.   no.of obs.
 1. FY'76           1           12
 2. FY'77          13            5


**********************************
*                                *
*       DATA MANIPULATION        *
*                                *
**********************************

         PLANT OUTPUT DATA
Data file name: MARK2
Number of obs: 17
Number of variables:  5
Variable names:
    1.  TEMP.
    2.  PROD.
    3.  DAYS
    4.  PEOPLE
    5.  H2OUSE

Subfiles:  NONE
```

Suppose that the first 12 observations were from the 1976 fiscal year and the next 5 observations showed to date the available monthly information for the next 5 months in fiscal 1977. We have created two subfiles to separate the two fiscal years. This was accomplished by pressing the EDIT key and selecting the "Subfile Structure" operation.

Reload data set into 85, using the RESTART key.

```
**********************************
*       SUMMARY STATISTICS       *
*          ON DATA SET:          *
*       PLANT OUTPUT DATA        *
**********************************


         BASIC STATISTICS
         ----------------

Var.              # of          # of
Names             Obs.       Missing
TEMP.               17             0
PROD.               17             0
DAYS                17             0
PEOPLE              17             0
H2OUSE              17             0

Var.
Names               Mean     Std. Dev.
TEMP.          18.2471        7.5122
PROD.       11610.4118     3174.1994
DAYS           21.4706        1.4628
PEOPLE        186.8235       21.9950
H2OUSE       3633.7059      491.2972

Var.                          Coef of
Names       Std.Error       Variation
TEMP.          1.8220         41.1692
PROD.        769.8564         27.3392
DAYS            .3548          6.8129
PEOPLE         5.3346         11.7731
H2OUSE       119.1571         13.5206

Var.          Coef of        Coef of
Names        Skewness       Kurtosis
TEMP.          -.5270         -.9397
PROD.          -.6850         -.4848
DAYS            .4919          .2301
PEOPLE        -1.2511          .4101
H2OUSE         1.3231          .9980


 95% CONFIDENCE INTERVAL ON MEAN
 -------------------------------

Var.
Names    Lower Limit   Upper Limit
TEMP.        14.3837       22.1104
PROD.      9977.9848    13242.8388
DAYS         20.7183       22.2229
PEOPLE      175.5119      198.1351
H2OUSE     3381.0416     3886.3702
```

Basic Statistics across all 17 observations.

User must specify the confidence level for the interval. We chose to obtain a 95% confidence interval.

```
        CORRELATION MATRIX
        ------------------

          PROD.    DAYS     PEOPLE

TEMP.     -.0924    .2686   -.1074
PROD.               .1057    .9185
DAYS                         .0319

          H2OUSE

TEMP.       .2504
PROD.       .6309
DAYS       -.0888
PEOPLE      .4134

         ORDER STATISTICS
         ----------------

Var.
Names          Maximum          Minimum
TEMP.          27.2000           4.2000
PROD.       16716.0000        5736.0000
DAYS           25.0000          19.0000
PEOPLE        211.0000         134.0000
H2OUSE       4936.0000        3110.0000

Var.
Names            Range        Midrange
TEMP.          23.0000          15.7000
PROD.       10980.0000       11226.0000
DAYS            6.0000          22.0000
PEOPLE         77.0000         172.5000
H2OUSE       1826.0000        4023.0000

Var.
Names           Median
TEMP.          18.4000
PROD.       12707.0000
DAYS           22.0000
PEOPLE        195.0000
H2OUSE       3437.0000

Var.
Names          25-th %          75-th %
TEMP.          13.3000          22.8000
PROD.       10768.0000       13114.0000
DAYS           20.0000          22.0000
PEOPLE        180.0000         198.0000
H2OUSE       3324.0000        3624.0000
```

The correlation matrix will only be printed if the user requests it. The variables PROD. and PEOPLE are highly correlated. (r=.9185).

Of course many other operations are possible by using the manipulation keys. And of course by specifying subfiles, you could have obtained the basic statistics for each subfile.

Notes

Appendix A

Limitations

The programs have been designed to operate in the basic machine. The maximum number of elements is 500. Hence, for two variables, a maximum of 250 observations may be input. This may be changed if more memory is available. However, the scratch file "DATA" on the "BASIC STATISTICS AND DATA MANIPULATION" tape cartridge may not be able to contain the increased amount of data. A new "DATA" file could be created to contain the increased data, or with a few program changes the data could be stored automatically on a data file on another mass storage medium and used as the "DATA" file on the cartridge presently is used.

If more than 500 elements are desired, a number of changes must be made. In file "START", line 180, change O2=500 to O2=N, where N is the total number of elements desired. Also, in all COM statements, the array D (?, ?) must be dimensioned as D(1, N). The following table gives the location of each COM statement:

File Name	Lines
"Autost"	20
"START"	30
"INPUT"	30
"LIST"	30
"EDIT"	30
"EDIT2"	40
"BASIC"	30
"BASIC2"	30
"ORDER"	30
"STORE"	40
"HCOPY"	30
"TRANS1"	30
"TRANS2"	30
"TRANS3"	30
"NEWST"	30
"ADVST"	30

The maximum number of variables is 12. To increase this, change N2 in file "START", line 170, from 12 to the number desired, N. In all COM statements, the vector V1$[*] must be dimensioned as V1$[M] where M = 6* N and N is the number of variables. If more than 12 variables are desired, some further changes must be made: in file "STORE", line 360, the CREATE statement should be changed to CREATE F$,2+(8+O2*8) DIV M, M (where M = 288+N*6); and the scratch file "DATA" must be made large enough to accommodate the larger data array.

The following line changes should also be made. In file "BASIC", change line 50 to DIM K1(N), K2(N), N(N), M(N), V(N). In file "BASIC2", line 60 should be changed to DIM N(N), M(N), V(N)., In file "ORDER", line 50 should be changed to DIM N(N), M(N), V(N), R(2*N), T(N).

The "BASIC STATISTICS AND DATA MANIPULATION" tape cartridge contains the data points used in the examples for the BASIC STATISTICS AND DATA MANIPULATION routines on the file "DATA". The user may wish to page through the manual and try each of the programs available in the pac, then compare the results with those in the examples. It should be noted, however, that each example was run using the original data and not data which had been transformed or edited.

Appendix B

Data File Configuration

The scratch file on the program medium, "DATA", and any files created to hold stored data and related information are configured as follows.

The data file is broken into logical records of 300 bytes each. The first logical record is a "header file", which contains information pertinent to the data set stored in the remaining logical records. The header file contains the following information (variables):

- data set title (T$(20 characters))
- number of observations (O1)
- number of variables (N1)
- variable names (V1$(6 characters per variable name))
- number of subfiles (S1)
- subfile names (S1$(6 characters per subfile name))
- subfile characterizations (S2(*))

The remaining logical records contain D(*,*) – the data matrix.

Appendix C

Program Documentation

The documentation for the programs in this pac is contained in the programs. DOCST1 and DOCST2.

The major variables are defined in addition to comments for major sections of code. To obtain the documention, load and run each program.

Appendix D

Using the Disc Version

The following information will increase your understanding of the disc version of this pac, and hopefully facilitate operation of the programs.

Printer Prompt

You have the ability to choose the output device by selecting the proper output code. After loading the program and pressing (RUN) , the printer prompt will ask you to specify the output device with the following codes:

Enter: 1 (END LINE) will direct system output to the CRT

Enter: 2 (END LINE) will direct system output to the internal printer

other numbers of specific printers will direct system output to an external printer.

A system output test is included with the above entry which will advance the desired printer one line if the system is operating properly.

Output via the CRT

When the CRT is chosen as the output device, the program will pause when displaying more than one full screen to allow full retention of output data. Simply press (CONT) to continue viewing until output is complete.

Operating Limits

The maximum operating limits of some of the programs have been slightly modified to accommodate the disc version of this pac. This need only be of concern as you approach these maximum operating limits.

References to Tape

All references to tape in this manual will be understood as references to the current mass storage medium, and therefore will apply to the disc version of this pac.

Appendix E

Using the Basic Statistics and Data Manipulation Pac With an Eighty-Column Display

Note: The actual error messages, prompt messages, and report formats may be different than those listed in the manual due to the larger display on your computer.

Operating Limits

The operating limits are dependent on available read-write memory (random-access memory or RAM). The programs in the eighty-column display Basic Statistics and Data Manipulation (BSDM) Pac automatically calculate these limits based on the amount of RAM available according to the table below. The number of variables and observations allowed will be displayed at the top of the screen when the BSDM Pac is run.

		32K	64K	96K	>=160K
Variables	N2	15	40	60	85
Observations	O2	600	2400	3200	5700
Subfiles		15	40	60	85

Remember that the number of observations listed above should be divided by the number of variables used to determine the maximum number of observations per variable. For example, if your computer has 96K bytes of RAM memory available and you have 32 variables, then you may have up to 100 observations per variable.

Data files created using the HP-85 BSDM Pac, including the scratch file "DATA", may be used by the eighty-column display BSDM Pac. However, files stored with the eighty-column display BSDM Pac cannot be used by the HP-85 Pac.

User-Defined Transformations

The same cautions and instructions for creating user-defined transformations should be observed as according to page 14 of this manual; but the following sequence of steps should be followed instead of those listed.

1. Type: `LOAD "TRANS1"`
   Note: Key #6 may be used here.
2. Press: (END LINE)
3. Type: `LOADBIN "REDZERg"`
   Note: Key #13 may be used here.
4. Press: (END LINE)

5. Type: `DELETE 1440, 1449`

   Note: Key #6 may be used here.

6. Press: END LINE
7. Enter the lines of code for your transform using line numbers 1440-1449, pressing END LINE after each line.
8. Type: `STORE "TRANS1"`

   Note: Key #7 may be used here.

9. Press: END LINE

Your user-defined transform is now stored in the `TRANS1` program file.