MULTIDISCIPLINARY ACCIDENT INVESTIGATION REPORT AUTOMATION

Manual for SPAD, Simplified Procedure for Analysis of Data Volume 3 of 5

HIGHWAY SAFETY RESEARCH INSTITUTE
THE UNIVERSITY OF MICHIGAN
ANN ARBOR, MICHIGAN 48104

OCTOBER 1972
CONTRACT No. DOT-HS-031-1-037
FINAL REPORT

Prepared for:
Department of Transportation
National Highway Traffic Safety Adminstration
Washington, D.C. 20590

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Multidisciplinary Accident Investigation Report Automation, Volume 3 of 5, Manual for SPAD Simplified Procedures for Analysis of Data

5. Report Date
October 1972

6. Performing Organization Code

7. Author/s)

8. Performing Organization Report No.
UM-HSRI-SA-72-1-3

9. Performing Organization Name and Address
Highway Safety Research Institute
University of Michigan
Ann Arbor, Michigan 48105

10. Work Unit No.

11. Contract or Grant No.
DOT-HS-031-1-037

12. Sponsoring Agency Name and Address
National Highway Traffic Safety Administration
Department of Transportation
Washington, D.C. 20590

13. Type of Report and Period Covered
Final Report
February 1971 through
October 1972

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

This is the third volume of a five volume final report describing results of the Multidisciplinary Accident Investigation Report Automation contract.

Access to data stored in the accident data banks at the Highway Safety Research Institute (HSRI) is considerably simplified through use of the SPAD (Simplified Procedures for Analysis of Data) system. This system allows users who have minimal knowledge in the operation of time-shared computer terminals to query the data banks for answers to many accident related questions.

This manual provides the necessary background for operating the Michigan Terminal System from a remote terminal; for understanding the basic structure and contents of the accident files and analysis tools; and for using the SPAD system to obtain the desired analysis results.

This manual and many of the HSRI accident data banks were developed under support from the Motor Vehicle Manufacturers Association.

17. Key Words
Multidisciplinary Accident Investigation Reports, Computer Automation of Accident Report, Statistical Analysis Programs, Instruction Manual, Timeshared Programs

18. Distribution Statement
Unlimited

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
None

21. No. of Pages
125

22. Price

A

MANUAL FOR

SPAD

(Simplified Procedures for Analysis of Data)

October 1972

Prepared by

Systems Analysis Group
Highway Safety Research Institute
University of Michigan
Ann Arbor, Michigan

FOREWORD

This document is the third volume of a five volume report describing final results of the Multidisciplinary Accident Investigation Report Automation contract conducted by the Highway Safety Research Institute (HSRI) for the National Highway Traffic Safety Administration. It documents the procedures used to access the HSRI accident data bank through the University of Michigan's time-shared computer network, and outlines the contents of the data bank.

Member companies of the Motor Vehicle Manufacturers Association (MVMA), staff personnel at the National Highway Traffic Safety Administration (NHTSA) headquarters in Washington, and six of the Multidisciplinary Accident Investigation teams are provided with full access to the accident data bank through the SPAD system via their own private computer terminals. The Canadian Department of Transport will soon be added to this list of users. The Highway Safety Research Institute maintains this system of access through continuing sponsorship by MVMA and NHTSA.

TABLE OF CONTENTS

List of Important Reference Items

SECTION I

INTRODUCTION

The Highway Safety Research Institute at The University of Michigan maintains an extensive collection of accident records that document important features of traffic collisions in many areas of the country. Due to the sheer volume of our accident records to date, it is necessary to utilize computerized storage and analysis of the data. The Statistical Research System was developed to provide the necessary analysis capability and is a descendent of the OSIRIS System originated at The University of Michigan. Some familiarity with the language of modern digital computer systems and the data analysis packages that may be used in conjunction with these systems is, of course, necessary for successful operation. If the necessary familiarity is overly complex, then the people who are most likely to benefit from the data will never achieve the proficiency required. In order to make the data as accessible as possible to those who can benefit from it, HSRI has developed the SPAD (Simplified Procedures for Analysis of Data) System to provide a simplified method of using the accident files in conjunction with the Statistical Research System. It is hoped that this feature will foster an increased usage of the accident data files to answer questions that arise in the course of highway safety related activities and to explore new problems that had not previously been uncovered.

HSRI obtains accident data from a variety of sources: the two most prominent are the police agencies and the various Multidisciplinary Accident Investigation Teams. In this data, each accident is characterized by a number of factors that describe the driver, vehicle, and environmental conditions recorded by the investigator. The volume of recorded accidents and the large number of factors for each accident make manual data handling procedures impossible. The accident data is consequently transcribed onto magnetic tape in a form that is usable on a digital computer.

If the stored information is to be of any value, it is necessary to have some technique for accessing the magnetic tapes and deriving from them the information that is desired by the data analyst. This function is performed for the user by the Statistical Research System (SRS)--a group of computer programs prepared for and maintained by HSRI for data manipulation and analysis. By proper use of SRS, the experienced user can create new data files, modify or update existing ones, and perform a number of analysis operations on existing files that range in complexity from listing a subset of the data for detailed scrutiny to complex statistical operations such as multivariate analysis of variance. The SRS in conjunction with the accident files comprises a powerful system for the collection and analysis of data.

The final component in the analysis hierarchy shown in Figure 1 is the computer and its executive supervisor that controls the operation of all functions resident on the machine. All HSRI accident files as well as the SRS are maintained on The University of Michigan's IBM 360/67 Computer under supervision of the Michigan Terminal System (MTS). MTS was developed for operation in a time-sharing mode, so that access to the computer could be gained from a variety of remote terminals for conversational usage of the computer in a real-time environment. By issuing the proper commands to MTS it is possible to sign on the system as a registered user and to perform all the operations necessary to access the desired data file, carry out the analysis required, and arrange for printout or storage of the results obtained.

While computerized storage and analysis is the only practical method of handling large data bases, it does introduce several difficulties for the user,

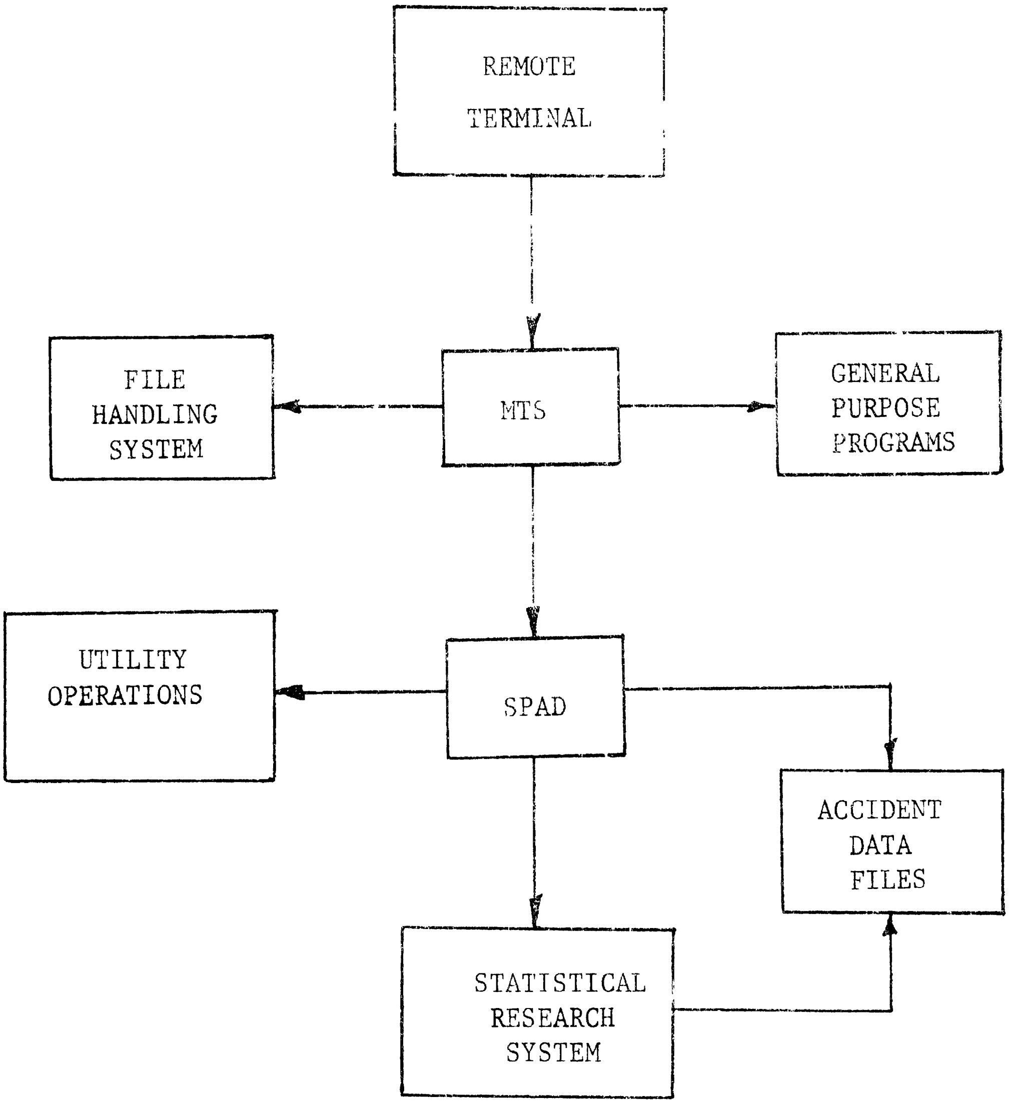


Figure 1 The Data Analysis Hierarchy

who, in most situations, is not experienced in computer operations. Thus, the minute attention to detail required to operate computers tends to repel many potential users who are unaccustomed to this detail in human relationships. Computers are designed for detailed tasks, however, and there is no reason that they should not be given the task of doing the operations that are difficult for the novice. The goal of the HSRI SPAD System is therefore clear: use of the computer itself to perform most of the detailed operations necessary to carry out an analysis task using the HSRI accident files. In implementing this goal, however, it is difficult to allow for all the possible manipulations that can be performed with MTS and SRS. Consequently, SPAD is designed to handle the routine operations normally encountered; the user is still encouraged to use the full capabilities of SRS to carry out more sophisticated analysis operations. In summary, SPAD is simply a technique to reduce the complexity of operations necessary to use SRS on MTS.

Before discussing the operational use of SPAD, a presentation of the background information on MTS, SRS, and the HSRI files necessary to successful use of the technique will be presented.

SECTION 2

THE MICHIGAN TERMINAL SYSTEM (MTS)

The University of Michigan computer system can service a large number of remote users concurrently, offering each a wide variety of services. The task of keeping track of all the programs in the machine and of devoting some attention to each of them on a regular basis is handled by the MTS operating system. In order to request service from the computer, a potential user must first identify himself as a registered account holder and then communicate his requests via the MTS Command Language.

Two operational modes within MTS are possible: conversational or batch. In the conversational mode, a remote terminal is used to communicate directly with MTS through a typewriter-like keyboard in conjunction with a telephone. Each command is processed as it is received and the results are reported back to the user as they are generated. Since each command may be based on previously obtained results, the conversational mode of operation is highly interactive.

Batch mode, on the other hand, is non-interactive. All requests must be completely preplanned and submitted to a remote batch terminal at the same time. Some time later--minutes or hours, depending on program length and computer processing load--the results may be picked up at the batch terminal.

The command language and its usage is essentially the same for both batch and conversational modes. The primary difference is that MTS replies after most commands in the conversational mode confirming the action that it has just taken, notifying the user of certain errors in the command, or requesting additional information that was not supplied. This feature is very useful when errors are made in the instructions given to the system. For example, if certain words are misspelled and MTS is unable to interpret the command, an error message will be issued. In some cases the error can be identified and the user will then be prompted for new information. In addition, if MTS is asked to empty or destroy files (that may contain important information) a confirmation request is issued before the operation is carried out, thereby allowing the user to change his mind.

For use with SPAD, especially for the novice user, conversational mode offers the most useful set of alternatives. Consequently, the discussion and presentation of examples will be limited to conversational usage. Due to the interactive nature of the conversational mode, our examples will consist of data entered by the user and data outputted by the computer. In order to distinguish the two, user supplied input lines will be underlined while machine supplied output will not. In practice, this underlining should not be used.

A. Terminals

Communication with MTS is generally carried out by means of a terminal connected to the computer through telephone lines. The remote terminal may be as simple as a touch-tone telephone or as complex as a remote computer; at the intermediate level of complexity and expense are the typewriter-like terminals (such as Teletypes, the IBM 2741, and the IBM 1050) and certain cathode-ray display terminals. The Teletype terminal device may be either a Model 33, 35, or 37 and may be ASR (automatic send and receive) or KSR (keyboard send and receive). The ASR models are equipped for use with paper tape.

In addition to the many types of terminal devices, there are several different types of control units used to connect the devices to MTS. The two most commonly used transmission control units are the Memorex 1270 Transmission Controller and the PDP-8 Data Concentrator. Each transmission control unit handles terminal devices differently; therefore, there are restrictions in their use. The Data Concentrator handles a wider variety of terminal devices than the Memorex 1270 and supports paper tape equipment of the ASR model Teletypes. The Model 37 Teletype can be used only with the Data Concentrator since its speed is not compatible with the Memorex 1270. In addition, the Data Concentrator supports a wide variety of more sophisticated terminal devices not described here.

Common terminal devices supported by MTS are described below:

1. The IBM 2741 Communications Terminal is very much like an ordinary electric typewriter. Its character set includes all commonly used text and programming symbols and it can be connected to the computer over great distances via telephone lines. Its output speed is about 14 characters per second and its carriage width is 130 characters.

2. The Model 33 Teletype is a commercial telecommunications unit which can be connected to the computer via telephone lines as the IBM 2741. Its output speed is 10 characters per second and its carriage width is 72 characters. It does not provide lower-case alphabetics. It is, however, the most common and the most inexpensive terminal available.

3. The Model 35 Teletype is functionally almost identical to the Model 33 but is more ruggedly built.

4. The Model 37 Teletype is a newer, faster model than the Models 33 and 35. Its output speed is 15 characters per second and its carriage width is 75 characters. It does provide lower-case alphabetics and many special symbols.

5. The GE TermiNet 300 is a high speed terminal that can operate at 10, 15, or 30 characters per second with a carriage width of up to 118 positions.

Physical characteristics of the various teletypes, the IBM 2741, and the GE TermiNet 300 are described in Appendix A.

Each of the terminal devices has a set of special control characters that instruct MTS to perform special functions (generally in reference to data input or output operations). These terminal control characters are shown in Table I.

The table indicates the control characters to:

° DELETE THE LAST CHARACTER TYPED: On the teletype, press CONTROL-H. Nothing will be **printed** on the paper as a result of this operation, and the printing element will not be backspaced. MTS automatically will have removed the last character typed. To delete the last n characters, enter CONTROL-H repeatedly, n times. On the IBM 2741, hit the backspace key. The print element will backspace, and the user may continue by typing over the deleted characters.

° DELETE A LINE: All characters already typed in the current line: Press CONTROL-N followed by CONTROL-S on the teletype. MTS will return "LINE DELETED" and will type a pound sign (#) on the next line, indicating that a new line can be entered. On the IBM 2741, the shift key should be pressed down and the dash character key pressed simultaneously, (underscore character) followed by the return key, in order to delete a line. MTS will respond with: LINE DELETED.

° END-OF-LINE: TERMINATE A LINE OF TYPE CHARACTERS: On the teletype, enter CONTROL-S or RETURN. On the IBM 2741, the return key should be pressed. MTS will then accept all characters typed in the current line as the total contents of that line. MTS will then process this line and return a pound sign (#) indicating that another line can be typed by the user. During the running of an analysis program, the carriage will return to the left margin without printing a pound sign (#), indicating that another line can be entered if requested by the program.

° INTERRUPT ANY CURRENT PROCESSING ACTION: --and regain contact with MTS (for example, stop an analysis program or stop the copying of a file): Press

TABLE I
TERMINAL CONTROL CHARACTERS

Function	TELETYPE (763-0300)	IBM 2741 (763-0510)	TERMINET 300 (763-0300)	TERMINET 300 (763-1500)
Character Deletion	CTRL-H	BACKSPACE	BS (Backspace)	BS (Backspace)
Line Deletion (Follow by line terminator)	CTRL-N	UNDERSCORE(_)	CTL-N	DEL (Delete)
Line Termination*	RETURN	RETURN	RETURN	RETURN
Attention Interrupt	BREAK	ATTENTION	INTERRUPT	INTERRUPT
End of File	CTRL-C	¢	CTL-C	CTL-C

(Note: CTRL-CHAR or CTL-CHAR indicates the character formed by first depressing the Control Key, then the character indicated.)

* Other characters are possible, but the indicated control keys are preferred.

the BREAK key on the teletype or the ATTENTION key on the IBM 2741. This operation is called an "ATTENTION-INTERRUPT"; MTS will respond with "ATTN!" or "ATTENTION INTERRUPT AT.........", return the print element, type a pound sign (#), and await the next instruction from the user.

All previous processing will have stopped, so the user may proceed with a new command. When the teletype BREAK key is depressed, the BRK-RLS (break release) key on the console will light up and the keyboard will be disconnected. To reactivate the keyboard, press the BRK-RLS key; the light should go out.

° GENERATE AN END-OF-FILE (EOF): The MTS command "$ENDFILE" can be used. In addition, CONTROL-C can be used on the teletype or "¢" can be used on the IBM 2741.

B. SIGNING ON AND OFF

The first steps in using MTS are to turn the terminal power on, make certain the device is in the "communicate" and not the "local" mode, and then dial the telephone number that establishes the computer link. A list of telephone numbers that are appropriate for the terminal used is shown in Table II.

TABLE II

MTS TELEPHONE NUMBERS

Teletype	(313) - 763-0300
IBM 2741	(313) - 763-0510
TermiNet 300 (Teletype Mode)	(313) - 763-0300
TermiNet 300 (High Speed Mode)	(313) - 763-1500 (Data Concentrator)
MTS STATUS	(313) - 763-0420

If the modem used with the terminal has a duplex switch, it should be in the half-duplex position for the 30300 line and in the full duplex position for the

31500 line. The exact method of dialing will depend on the terminal.

1. With a regular telephone and separate Modem simply dial the number and place the headset in the modem cradle.
2. For the Bell Dataphone, press the TALK/CLEAR button, dial the number, press DATA when a steady tone is heard and put the headset in the cradle.
3. For teletypes with a built in coupler and phone, press ORIG and dial the number.

When the connection is established MTS will output some information and descriptive header material that again depends on the terminal device used. Examples 1 and 2 show typical replies for a teletype and for an IBM 2741, respectively.

Example 1:

```
 MTS (LA35-0060)
  USE CONTROL-H FOR BACKSPACE.  SEE CCMEMO #M196.
 WHO ARE YOU?

 UMHYSRI A AA
#NEXT EXPECTED SHUTDOWN: 12:30 PM
#
```

Example 2:

```
   type an a, then return @idu ps pg @zus nu@tns
 a

 MTS (LA04-0037)
#NEXT EXPECTED SHUTDOWN: 12:30 PM
#
```

In these examples, the CONTROL-H and EXPECTED SHUTDOWN messages are informative and do not command user compliance. The only operator intervention required here is the entry of an "a" in Example 2. This action identifies the type of 2741 terminal that is being used. The "WHO ARE YOU" question in Example 1 is taken care of by an answerback device in the teletype. If there is no pound sign (#), the user should enter a CTRL-S after the "WHO ARE YOU".

The final pound sign "#" in the initial MTS connect message indicates that you are properly connected to the system and that MTS is waiting to receive its first instruction. The first command that is issued must be a valid signon command with the appropriate user number obtained from the Computing Center or from HSRI. The machine will respond with a request for the user password as shown in Example 3 (IBM 2741) or Example 4 (Teletype).

```
Example 3                 #$SIGNON MYID
                          #ENTER USER PASSWORD
                          ?■■■■■■■■■■■■■
```

```
                          #$SIGNON MYID
Example 4                 #ENTER USER PASSWORD
                          ?__________
```

The password is a "combination lock" that prevents unauthorized use of your ID number and should consequently always be concealed. For an IBM 2741 the password may be entered in the obscured area provided by the computer. On a Teletype or other terminal, the duplex switch should be placed in the full duplex position and the password typed in. In either case, if the password is correct, MTS will respond with the confirmation message shown in Example 5.

```
                          #**LAST SIGNON WAS:  (Time)   (Date)
Example 5                 # USER "MYID" SIGNED ON AT  (Time)  ON  (Date)
                          #
```

Rule 1: The first character of a command line to MTS is a "$". Although there are cases in which the dollar sign need not appear (e.g., see Section 2-F), it is always acceptable to include it.

Rule 2: The first MTS command must be a $SIGNON followed by a space and then the user ID.

When you get an ID number from the Computing Center, a password will already be assigned to it. This should be changed to something safer, more meaningful, and more easily remembered. The $SET command is used to change your password. To change your current password to a new one called NEWPASS use the command shown in Example 6.

Example 6

```
#$SET PW=NEWPASS
#
```

The $SET command can be the part of any job. The password should be no longer than twelve characters and must not contain any blanks.

The last command to MTS must be a $SIGNOFF command. This indicates that the user is finished using the system. At signoff, a number of pertinent statistics that describe the job are printed out at the terminal. A typical signoff sequence is shown in Example 7.

Example 7:

```
#$SIGNOFF
#OFF AT 16:55 26      12-16-71
#ELAPSED TIME         27.15 MIN.         $1.35
#CPU TIME USED        20.371 SEC.        $1.77
#CPU STOR VMI         10.083 PAGE-MIN.    $.53
#WAIT STOR VMI         6.558 PAGE-HR.
#DRUM READS         1382
#APPROX. COST OF THIS RUN IS     $3.64
#DISK STORAGE         24.567 PAGE-HR.     $.01
#APPROX. REMAINING BALANCE:     $33.03
```

<u>Rule 3</u>: The last command must be $SIGNOFF to tell MTS that you have finished using its facilities. MTS will return with statistics on the completed run showing how much the run cost, the time it required, etc.

If the full set of signoff statistics are not desired, it is possible to use the commands $SIGNOFF SHORT or $SIGNOFF $ to obtain an abbreviated printout. In particular, $SIGNOFF $ is quite useful and prints out only the cost of the run and the approximate remaining balance. Example 8 illustrates this procedure.

Example 8;

```
#$SIGNOFF $
#OFF AT 09:04.40     01-11-72
#   $.50
# $20.98
```

So far, several words of the MTS command language have been encountered (SIGNON, SIGNOFF, SET). There are many more, some of which will be introduced in the sections that follow. In general, the command statements have a fairly free format. A safe rule is:

> Rule 4: Don't leave a space between the dollar sign and the command word. In general you should leave a space between "words" when there is no explicit separator, such as in $SIGNON MYID. However, if there is a separator such as an equal sign (=), you should not leave any spaces around it. For example, in the phrase "PW=PASSWORD" there should be no blanks.

C. STORING INFORMATION IN MTS

Information is commonly stored in some device that is referred to as a file. However, this is an overworked term and may lead to some confusion. We have seen in the Introduction how the accident information is stored on magnetic tapes that are referred to as "Accident Files". Similarly, the Computing Center maintains programs and information on "Public Files" for utilization by all computer users. Finally, the user himself may create "Private Files" on his own user account to store the results of the analysis operations that he has performed on the SPAD system. These "files" are created and maintained by the user, and the Computing Center charges a storage fee that is proportional to the size of the storage space used (see Example 7). Storage size is measured in PAGES of information. One PAGE may contain up to 4096 characters and closely corresponds to 60 lines of data, each line containing 68 characters. An MTS PAGE is thus approximately equal to a full physical page of teletype output.

MTS records information on disk storage devices. A data file may be thought of as an area of this disk where information can be stored. Every file has a name (up to twelve characters long) which one uses to communicate to MTS about that particular file. Files may be PUBLIC (usable by everyone) or PRIVATE. We will discuss public files later; for now, attention will be directed to private files--those files created and manipulated by users for their particular purposes.

Private files belong to a specific user and may be accessed only by him unless he gives explicit permission to others. There are two types of private files, PERMANENT or TEMPORARY. Permanent files must be explicitly created by the user and exist until destroyed. Temporary files may be created by the user or by MTS when their names are first encountered by the system. More importantly for the user, they are automatically destroyed when the user signs off. Temporary files are distinguished by file names that begin with a minus sign (-). Thus, MYFILE is the name of a permanent file while -TEMP denotes a temporary file.

Files are composed of lines of information. If the information is put into the file by an SR system program, a line is generally the information contained in one SR system output record. If the information is put in the file by the user via a typewriter terminal, a line is the information typed before the line return code is entered (Control-S or Return). The line number for a given line may have any value in the range -99999.999 to +99999.999. For most purposes, line numbers start with 1 and increase consecutively by integer values. Information for different line numbers may be entered or taken out of the file in any order.

Before using a file, it is necessary to create it. While MTS will automatically create temporary files when necessary, it is always possible to create a file with the $CREATE command. In the examples that follow, it will be assumed that the user has correctly signed on the system and that MTS has responded with a pound sign (#) and is ready to receive commands.

To create a permanent file called MYFILE it is only necessary to issue the command $CREATE, followed by the desired name. The procedure together with the MTS response is shown in Example 9.

Example 9:

```
#$CREATE MYFILE
# FILE "MYFILE" HAS BEEN CREATED.
#
```

If the name of the file is not acceptable to the system, or if no file space is available on your account, an appropriate error message will be issued.

> Rule 5: A file of modest size can be created by the $CREATE command followed by a space and then the name that you wish to attach to it. A permanent file will be retained for your use until destroyed.

Before storing information in a file, it is necessary to make sure it is the active file. The active file denotes the storage space where MTS puts information lines that are submitted to it. Any file that exists on the user account can be made the active file by using the $GET command. The procedure is shown in Example 10.

Example 10

```
#$GET MYFILE
#READY
#
```

> Rule 6: The $GET command causes the file referenced to become the active file for the system. This step is necessary prior to adding information to, or correcting information in, an existing file.

If the file is to be used for storage of data generated by a program, then the information will be put into the file in a format that is under program control. If the information is to be entered by the user, then two courses of action are possible. If the lines are to be entered in a consecutive fashion, then we can use the $NUMBER command to automatically number the lines and enter the information in the active file. Example 11 shows a simple job that creates a file called FILEFACTS and then fills it with six lines of information.

Example 11:

```
#$CREATE FILEFACTS
# FILE "FILEFACTS" HAS BEEN CREATED.
#$GET FILEFACTS
#READY.
#$NUMBER
#      1_THIS SAMPLE FILE CONTAINS SOME
#      2_USEFUL BUT IRRELEVANT
#      3_INFORMATION ON FILES IN MTS.
#      4_ 1) A FILE IS A SET OF NUMBERED
#      5_LINES STORED UNDER SOME UNIQUE
#      6_NAME.
#      7_$UNNUMBER
#
```

Rule 7: The command $NUMBER instructs MTS to number the lines of data that follow and to put these lines in the active file. MTS types the line number and an underline character (_) and assigns the contents of the line entered to the given line number in the active file. The numbering starts with 1 and increases in unit increments (i.e., 1,2, 3: etc.)

Rule 8: Lines beginning with a dollar sign in the first position are considered to be commands and are executed by MTS. If it is necessary to enter a dollar sign into the first character of a line in the active file, a double dollar sign must be used. Thus, in Example 11, the line

```
#      7_$$UNNUMBER
```

would cause line 7 of the file to contain the phrase "$UNNUMBER".

Rule 9: The command $UNNUMBER turns off the automatic line numbering feature.

Data may also be added to the active file by entering the line number, a comma, and the data line directly to MTS. Thus, if the file FILEFACTS was an existing file, the first line of Example 11 could be entered by the following set of commands:

Example 12:

```
#$GET FILEFACTS
#READY
#1,THIS SAMPLE FILE CONTAINS SOME
#
```

The comma is a delimiter to separate the line number information from the contents of the line (which may begin with a number).

Rule 10: All lines of information which (1) start with a line number (either explicitly or via $NUMBER) and (2) do not have a single "$" following the number are put into the currently active file. Any line not starting with a line number or containing a single "$" after a line number is assumed to be a command to be executed immediately.

It is often necessary to modify, or revise, the contents of a file. A drastic modification is the elimination of all the information that currently exists in a given file. This is accomplished as shown in Example 13.

Example 13:

```
#$EMPTY MYFILE
#FILE "MYFILE" IS TO BE EMPTIED.PLEASE CONFIRM.
?OK
#DONE
#
```

Rule 11: The $EMPTY command will delete all the contents of the file specified, but will not destroy the file or modify its size (i.e., a large file remains large!).

Rule 12: Any response to the confirmation request, other than "OK", "O.K.", or "!" will abort the command.

Less drastic modifications may be performed by the procedure for entering new data shown in Example 12. For instance, if we want to change the information in file "FILEFACTS" generated in Example 11 we could use the procedure:

Example 14:

```
#$GET FILEFACTS
#READY.
#6,NAME ASSIGNED BY THE USER.
#2,
#3.5,YOU SHOULD LEARN THEM.
#
```

This sequence of instructions places a line numbered 3.5 between lines 3 and 4 and replaces the contents of line 6 with new information. Line 2 is deleted by terminating the line immediately after the comma. Note that it is not necessary to enter the information in line number order.

Remember from Rule 10, that MTS places all input lines that are interpreted as data lines into the currently active file. To protect files from having data stored inadvertently, it is good practice to release files from active status after the desired information has been entered. This is done with the $RELEASE command. Example 15 shows the use of the $GET and $RELEASE commands to assign the active file.

Example 15:

```
#1,THIS IS AN INPUT DATA LINE !
#THERE IS NO ACTIVE FILE TO PUT THAT IN.
#$GET FILE
#READY
#1,THIS IS AN INPUT DATA LINE !
#$RELEASE
#1,THIS IS AN INPUT DATA LINE !
#THERE IS NO ACTIVE FILE TO PUT THAT IN.
#
```

Once the data in our files has been modified and is ready for use, it is often necessary to list the file on the terminal so that we can examine the entire contents and see what line number corresponds to each data line. Similarly, we might desire to copy the information to another file. These operations are performed by the $LIST or $COPY commands. Before discussing these commands, however, it is useful to introduce the concept of reference to portions of a line file.

In many operations, we are not interested in the entire contents of a file, but only that information contained in a given line number range. Thus, we might want to know what is contained in the first five lines of the file, or perhaps in the last five lines. The portion of the file that is of interest can be specified by appending the line number range of interest. That is,

```
MYFILE(1,5)
MYFILE(20.6,399.2)
```

With this technique the last line is denoted by LAST so the third and fourth lines from the end of MYFILE can be referenced by

```
MYFILE(LAST-4,LAST-3)
```

If the last line of the file is one of the line number delimiters, it can be omitted in the specification. Thus, the last five lines of MYFILE can be referenced by

```
MYFILE(LAST-4)
```

As a final note, the contents of the file from line number one to the end of the file is referenced by the filename with no appended delimiters. For example,

```
MYFILE
```

refers to the contents of MYFILE from line number one to the end of the file. Since line numbers may be less than one, this may not reference the entire file.

With this convention out of the way, the list and copy commands are straightforward. To list the contents of file (line numbers and line content) use the MTS command

```
            $LIST MYFILE
     OR,    $LIST MYFILE(J)          J,K=Line Numbers
     OR,    $LIST MYFILE(J,K)
```

For example, we can list the contents of "FILEFACTS" to obtain the following printout.

Example 16:

```
#$LIST FILEFACTS
>      1      THIS SAMPLE FILE CONTAINS SOME
>      3      INFORMATION ON FILES IN MTS.
>      3.5    YOU SHOULD LEARN THEM.
>      4       1) A FILE IS A SET OF NUMBERED
>      5      LINES STORED UNDER SOME UNIQUE
>      6      NAME ASSIGNED BY THE USER.
#END OF FILE
#
```

Rule 13: The $LIST command, followed by a file name (with or without line number modifiers as explained above) causes the line number and full contents of each line to be printed at the terminal.

Similarly, we can use the copy commands

```
             $COPY MYFILE
OR,          $COPY MYFILE(J)          J,K=Line Numbers
OR,          $COPY MYFILE(J,K)
```

to copy the contents of the file without the line number information to the terminal. Again using the FILEFACTS example, we could print out the contents from line 3.5 to the end of the file as shown in Example 17.

Example 17:

```
#$COPY FILEFACTS(3.5)
>YOU SHOULD LEARN THEM.
>1) A FILE IS A SET OF NUMBERED
>LINES STORED UNDER SOME UNIQUE
>NAME ASSIGNED BY THE USER.
#
```

The $COPY command in general will transfer the contents of one file to another. Thus, we may use

Example 18:

```
#$COPY FILEFACTS TO NEWFILE
#
```

to copy the contents of FILEFACTS to another file called NEWFILE.

Rule 14: The $COPY command transfers the contents of one file into another. In this process, the lines are numbered sequentially from 1 as they enter the new file regardless of what their line number may have been in the oldfile. The oldfile is retained unaltered.

Rule 15: When copying to a terminal or line printer, the first character of each line is examined to see if it is blank or if it is a carriage control character. (0,-, +,&,9,1,2,4,6,8,;,<). If one of these characters is encountered, the appropriate carriage control action is taken (double space, new page, etc.), and the remainder of the line is printed as it is. If the first character is not recognized as a carriage control character, then the full line is printed.

The effect of Rule 15 can be seen in a comparison of Examples 16 and 17. In listing FILEFACTS, we see from Example 16 that line 4 begins with a blank and that the "1" lines up with "0" in line 3.5. Copying the file, however, as shown in Example 17, inddcates that the "1" now lines up with the "Y". This occurs because the blank at the beginning of the line 4 was interpreted as a no-action carriage control character. The true line contents may be obtained by using the $LIST command.

The final operation in file handling that we will discuss is the destruction of a file and its contents. Each file that is maintained on a user account takes up a part of the user's allocated file space and also accrues a tenancy charge that depends on the size of the file. For these reasons, it is desirable to destroy all files that are not needed. To do this, proceed as in Example 19.

Example 19 :

```
#$DESTROY MYFILE
#FILE "MYFILE" IS TO BE DESTROYED.PLEASE CONFIRM.
?OK
#DONE.
#
```

Rule 16: The $DESTROY command deletes the entire contents of the file specified, and removes the name of the file from the user catalog.

Rule 17: Any response to the confirmation request, other than "OK","O.K.", or "!" will abort the command.

D. RUNNING UTILITY PROGRAMS

Up to now we have discussed operations that have been performed under the direct control of MTS. It is possible to cause MTS to relinquish control to an operating program specified by the user. These programs are usually written in some programming language such as FORTRAN, BASIC, etc. The user may store one of these programs in a file called, for example, PROGRAM. To execute this program, it is only necessary to use the command $RUN as shown in Example 20.

```
#$RUN PROGRAM
#EXECUTION BEGINS.

[Machine under program control]

#EXECUTION TERMINATED.
#
```

Example 20:

Rule 18: The $RUN command causes the program stored in the referenced file to be loaded in the machine and executed. Operation of the machine is then under control of the program until execution is terminated and control is returned to MTS.

The real utility of the $RUN command for the non-programmer comes about from the fact that the University Computing Center maintains a large number of operating programs that may be accessed by any user. These programs are stored in the Public Files that were discussed at the beginning of this section. A public file has a name that begins with an asterisk (*); thus, *CATALOG is the name of a public file that contains a program to list the files currently maintained in the user's account. A number of these files that are of interest to the SEAD user are described below.

1) *CATALOG

This program produces a complete listing of all the user's permanent files and the total amount of storage space that is being utilized.

```
#$RUN *CATALOG
#EXECUTION BEGINS

[Listing of User Files]


USER MYID HAS XXXXXX FILE(S) WITH A TOTAL SIZE OF XXXX     PAGES
#EXECUTION TERMINATED
#
```

Example 21

2) *STATUS

Reports the amount of funds and storage assigned, used, and remaining at the time of last signoff. This report, together with the signoff information from the user's current run, will give a complete total of the funds expended.

```
#$RUN *STATUS
#EXECUTION BEGINS

STATUS OF MYID AT LAST SIGNOFF               USED         MAXIMUM        REMAINING

CUMULATIVE CHARGE          ($)             254.25         300.00           45.75
PERMANENT DISK SPACE       (PAGES)             32             50              18
CUMULATIVE TERMINAL TIME   (HR)             13.53


#EXECUTION TERMINATED
#
```

Example 22

3) *USERS

Reports the current load of users on the computing system in terms of terminal users, batch jobs, and non-MTS tasks.

```
#$RUN *USERS
#EXECUTION BEGINS
 THERE ARE 39 TERMINAL USERS, 5 BATCH TASKS, 39 AVAILABLE LINES, AND 12
 NON-MTS JOBS USING 1208 VIRTUAL PAGES AND 274 REAL PAGES.  HARDWARE IS
 CPU'S P1 P2, STORAGE A B C D E F,CCU'S 0 1
#EXECUTION TERMINATED
```

Example 23

A total of 44 jobs are being processed by MTS at the moment, 39 from remote terminals and 5 entered on cards at the Computing Center for batch processing. The additional 12 non-MTS jobs are major system communication and storage management routines. Thus the multi-programming system is processing a total of 56 active programs. Of the total virtual memory of approximately 2200 pages capacity (900 each on the two drums and 384 pages of core memory), 1208 were being used when *USERS was being executed; 274 of the 384 pages of the core memory (the "real" pages) were being used by active programs. All major hardware items are in operation, including two central processors (P1 and P2), the six magnetic core memory boxes (storage A,B,C,D,E,and F), and the two channel control units (CCU's 0 and 1).

4) *FILESNIFF

Gives a number of features of files that characterize the files on the user account. The user will be prompted for file names until an end of file code is entered (Control-C, $ENDFILE, or ¢). If the characteristics of only one file are desired, the modifier PAR=FILENAME may be appended after the file name.

```
#$RUN *FILESNIFF PAR=FILEFACTS
#EXECUTION BEGINS
   FILE=FILEFACTS
     LOC=DISK
     TYPE=LINE
     VOLUME=MTS005, NO. EXTENTS=01
     NO. LINES=00006     MAX. LINE LENGTH=00031
     NO. PAGES=(0002,0002), SIZE PARS=(000146,000146).
     PERMIT CODE = NONE
#EXECUTION TERMINATED
#
```

Example 24

This printout shows the number of lines in the files and the maximum line length. Under NO. OF PAGES, the first figure in the parenthesis indicates the file size that the user charge is based on. All files may be *FILESNIFF'ed by the commands

1) $RUN *CATALOG SPRINT=-A

2) $RUN *FILESNIFF GUSER=-A

E. RUNNING USER-PRODUCED PROGRAMS

In addition to the public files discussed above, there are a large number of programs that are produced and maintained by system users themselves for utilization by the entire computing community. These files are usually maintained in a read-only status and are useful in performing a wide variety of processing tasks. Because the programs are user-maintained, the responsibility for program operation and errors must be with the person who runs these programs although the originator in most cases is happy to be notified of any errors or difficulties. Please contact HSRI for details on running these programs.

An extremely useful statistical analysis program is CONSTAT, maintained by The University of Michigan Statistical Research Laboratory. This user-oriented, interactive program may be used to compute histograms, correlations, etc., and a wide variety of data handling and transformation tasks. A description of program features may be obtained by the following procedure:

1) Issue the MTS run command.

   $RUN STAT:CONSTAT

2) Certain program identification details will be printed out at the terminal.

3) The program will eventually issue the message

   WHICH COMMAND?

   After this request the user should enter INFO.

4. When the descriptive information has been typed out, the WHICH COMMAND? statement will again be issued. At this point enter FINI to terminate program execution and return user control to MTS.

Additionally, HSRI maintains an internal system of programs for the secondary processing of data generated by the SR system or by other data sources. Of prime interest are the plotting programs to provide graphical representations of the data. Most important are a 3D plotter program to provide graphical output of bivariate frequency information (i.e. a histogram with two independent variables) and a mapping program to produce grey level maps of the United States by state or the State of Michigan by county. The 3D plotter produces

CALCOMP graphical output, while the mapping programs output directly to the terminal device or line printer. An extremely useful histogram program is available under the name BARGRAPH. This program will produce histograms in a wide variety of formats from tabular input data.

F. ABBREVIATED MTS COMMANDS

When operating from a terminal in conversational mode on MTS, it is often possible to omit the dollar sign before the command and to use a shortened form of the command. For example

```
#C FILEFACTS
```

is equivalent to

```
#$COPY FILEFACTS
```

The savings in user interaction time is obvious.

In some cases the command may be shortened, but the dollar sign may not be omitted. Thus, entering the command U (for UNNUMBER) while the user is in the automatic numbering mode would cause the character U to be entered on the appropriate file line. The command "$U" is recognized as an MTS command to turn off numbering.

The acceptable short form of the MTS commands discussed in this section are indicated in the MTS command summary (presented in Section H) by underlining.

G. DEVICE CONTROL COMMANDS

In addition to the MTS commands that are prefixed by a dollar sign, there are a number of useful Device Control Commands that are prefixed by a percent sign (%). These commands do not instruct MTS, but simply tell the terminal control device how to handle the information that is passed between the terminal and MTS.

The most frequent use of these commands is in regard to line length. A "line" of information in MTS may be up to 255 characters long while output data lines from the SRS are 132 characters long. Line printers and some terminals (IBM 2741, Model 38 Teletype, etc.) can handle 132 characters, but most teletype-like terminals will only handle 70-80 characters.

If a file with a line length longer than a given device will handle is output to a terminal, the device controller will simply output the available number of characters and throw the rest of the line away. For example, a bivariate table from the SRS that is copied to a Model 33 or Model 35 Teletype will simply be truncated and only the left portion of the table will be obtained.

MTS can be forced to print the entire 255 character line by the following command sequence

Example 25:

```
#%LEN=255
%OK.
#
```

Any desired length can be substituted for 255 of course. This procedure will result in the MTS "line" being output as a number of terminal lines. The logical form of the output is consequently modified, but the information is saved.

The right margin can be similarly modified by the %RMAR command. For example, the 132 character IBM 2741 could be made to appear like a 72 character device with the command sequence

Example 26:

```
#%RMAR=72
%OK.
#
```

The RMAR command can be used in combination with LEN to control the form of the output.

Finally, there is a device control command to prevent the computer from disconnecting the telephone line when a $SIGNOFF command is given. The sequence

Example 27:

```
#%DON'T
%OK.
#
```

will accomplish this. After a signoff, the computer will resubmit the signon information to the terminal. This command is very useful when a single user wishes to consequtively sign on to different user ID's or when there is a queue of people waiting to utilize a single terminal.

H. MTS COMMAND SUMMARY

	Phone Lines (Area Code 313)
763-0300	Teletypes
763-0510	IBM 2741
763-1500	Data Concentrator
763-0420	MTS Operational Status
763-0428	HSRI Systems Analysis

	MTS General
$SIGNON MYID	To identify a user to the system
$SIGNOFF	To notify the system of a user's departure
$SIGNOFF SHORT	
$SIGNOFF $	
$SET PW=PASSWORD	To change the user password
$RUN *CATALOG	To load and initiate execution of a program
*USERS	
*STATUS	
*FILESNIFF	

MTS FILE HANDLING

$CREATE FILE	To create a permanent or temporary file
$EMPTY FILE	To empty the contents of a file without changing its size or destroying it
$DESTROY FILE	To destroy a permanent or temporary file
$GET FILE	To establish a file as the active file
$COPY FILE $COPY FILE1 TO FILE2	To copy from a file to a terminal or another file without line numbers
$LIST FILE	To list a file on the terminal with line numbers
$NUMBER	To start automatic numbering of input lines being read from the terminal and being written on the active file.
$UNNUMBER	To turn off automatic numbering
$RELEASE	To release the active file or device

DEVICE CONTROL

%LEN=NUM	To change the length of the output line at NUM characters
%RMAR=NUM	To change the right margin to NUM characters
%DON'T	To prevent MTS from hanging up at signoff.

SECTION 3

THE STATISTICAL RESEARCH SYSTEM (SRS)

The Statistical Research System (SRS) is a package of data handling and processing programs to permit the analysis of large quantities of information in a timely fashion. SRS is resident on the University of Michigan IBM 360/67 computer and may be operated in either batch or conversational mode under MTS.

The system is a group of application programs and a subroutine library. Each program is accessed by a unique reference number. The SRS was designed to process magnetic tape or direct access files and consequently permits the analysis of large data sets. Control operations necessary to handle the magnetic tape (tape rewinding, positioning, etc.) are performed by the programs and are transparent to the user.

Three features of the SRS that are pertinent to the SPAD user are discussed in this section:

1) SRS program descriptions

2) File Structure

3) Filtering and Recoding Variables

A) SRS PROGRAM DESCRIPTIONS

A complete listing of the SRS analysis programs is given in Table III. For the purposes of this manual, we will only be concerned with those programs that are followed by an asterisk. Users are encouraged to utilize the full capabilities of the system when they feel that they have the experience to do so; more information can be obtained by calling HSRI.

TABLE III

CATALOG OF ANALYSIS PROGRAMS IN THE STATISTICAL RESEARCH SYSTEM

1) DATA SET LISTING*
2) HISTOGRAM*
3) MEANS AND MARGINALS*
4) ONE-WAY ANALYSIS OF VARIANCE*
5) BIVARIATE FREQUENCIES WITH TWO DIGIT CODES AND FOUR FILTERS*
6) PRINT COLUMNS
7) BIVARIATE FREQUENCIES
8) BIVARIATE FREQUENCIES WITH TWO DIGIT CODES
9) MISSING DATA CORRELATIONS
10) PARTIAL CORRELATIONS
11) LINEAR REGRESSION
12) MUTLIPLE ANALYSIS OF VARIANCE
13) AUTOMATIC INTERACTION DETECTOR
14) MULTIPLE CLASSIFICATION ANALYSIS
15) FACTOR ANALYSIS

Five programs have been chosen for SPAD to satisfy most user needs. The SPAD programs are:

1) Data Set Listing

   Selects and prints a subset of the data file.

2) Histograms

   Prints a bar graph showing the univariate frequency distribution of one variable.

3) Means and Marginals

   Means: Provides case counts, sum of weights, ranges, means, standard deviations, skewness, and kurtosis.

   Marginals: Provides one-way frequency distribution (univariates) of individual variables and percentages on the distributions.

4) One-Way Analysis of Variance

   Produces one-way analysis of variance tables.

5) Bivariate Frequencies

   Produces bivariate (two-way) frequency tables for pairs of variables.

A brief description of these programs and the results they produce is presented below. Operational instructions for utilizing the programs in SPAD are discussed in Section 5.

Data Set Listing

This program is used to selectively retrieve and list a subset of the original file. Provisions are made for selecting the cases to be retrieved, and for selecting the variables to be listed. This program is not an analysis program in the strict sense, and is useful only when the detailed features of a small number of cases must be carefully perused. For instance, if only 10 to 20 cases having certain desired characteristics can be found in a given file (i.e., accidents involving a school bus, a fire, etc.), then any statistical analysis of the information is probably not meaningful. Such situations can be handled by listing a number of variables of interest for these cases. The resultant listings can be examined manually in detail to see if any significant features are discernible. In other cases, it may be desirable to determine the accident identification numbers for a subset of cases in order to obtain information from the reports themselves: this may be easily done with a Data Set List.

Histograms

One of the most meaningful quantities describing the nature of a given variable in an accident file is the distribution function for this variable. The distribution function describes the percentage of cases in the file that are characterized by each possible value of the variable in question. For example, if our variable describes the side of a coin that lands upwards after being thrown in the air (possible values, 1 = heads, 2 = tails), then the distribution function for a large number of tosses would show a value near 50% for both possible variable values. Other forms of the distribution function simply show the case count for each value of the variable and are not normalized to indicate percentages.

In the Histogram program, the possible values of the variable being analyzed are grouped into unique 'class intervals' that define a range of values on the horizontal axis of the histogram. The vertical axis represents the total number of cases that occur for each interval of the variable and is divided into units that reflect the approximate number of cases within the given classes.

For each histogram, the program also calculates the mean (M) and standard deviation (S) for the variable. These quantities are computed from the formulas:

$$M = \frac{1}{N} \sum_i X_i$$

$$S^2 = \frac{1}{N(N - 1)} [\sum_i X_i^2 - N^2M^2]$$

where X_i represents the score for the observation i, and N is the total number of observations.

Means and Marginals

This program is used to provide a number of summary statistics characterizing the distribution function for a user-supplied list of variables. This summary includes case counts, sum of weights for weighted data, number of missing data cases for one or two missing data codes, range of variable values, means, standard deviations, skewness, and kurtosis. In addition, during a second pass over the data, the program accumulates marginal distributions for each variable and optionally computes percentages on the marginals.

After the first pass over the data, the following quantities are printed out using one line for each variable, 15 variables per page:

1) The variable number

2) The variable name

3) The case count for the variable (excluding missing data)

4) The mean

5) The standard deviation

6) The minimum code value (can be negative)

7) The maximum code value (can be negative)

8) The case count for the first missing data code

9) The case count for the second missing data code (all code values = the missing code value specified)

10) Skewness (3rd moment)

11) Kurtosis (4th moment)

If the data is weighted, the statistics are of the weighted form and the following is also printed on the line:

12) Sum of weights

Optionally, additional passes over the data will accumulate marginal distributions for each variable. The missing data codes are included in the distribution range. These distributions may be weighted. Distribution percentages can optionally be computed and are printed out below the marginal for each code value. For each variable, starting with the lowest code, 15 code values are printed per line with the marginals (and percentages) underneath the respective code value. Each consecutive code value in the range is printed out until the maximum code value is reached.

One-Way Analysis of Variance

Analysis of Variance is a statistical technique to test the equality of more than two population means, thereby determining if the variations that occur are significant (in a statistical sense) or due to the action of chance alone. For example, a tire company may want to test the lifetimes of three or more different types of tires and determine from the average lifetime for each type if there are significant differences in tire durability.

For each value of the control variable specified, the program evaluates:

1) The control variable code value (CODES)

2) The case count for this code value (N)

3) The sum of weights (WEIGHT-SUM)

4) The percentage distribution function for 3) (%)

5) The mean value of the weighted dependent variable (MEAN)

6) The standard deviation of the weighted dependent variable (S.D. (ESTIM.))

7) The sum of weighted case values (SUM OF X) (Equal to N*MEAN)

8) The percentage distribution function for 7) (%)

9) The sum of squared, weighted case values (SUM OF X-SQUARED)

At the bottom of the data output page, a number of quantities are produced relative to the statistical significance of the variations encountered. These quantities are discussed in a more complete treatment of ANOVA presented in Appendix B.

The analysis of variation technique is very useful in data analysis operations. For instance, suppose we would like to evaluate the average number of fatalities per accident as a function of weather (i.e., clear or cloudy, raining, snowing, etc.). Running ANOVA on an accident file with Weather as the control variable and Number of Fatalities as the independent variable would compute the desired averages.

Bivariate Frequencies

In the Histogram program, the concept of a distribution function was introduced. For a single control variable this quantity shows the frequency with which the possible values are represented in the accident population. This concept can be easily extended to two or more dimensions.

In a general case, the distribution function for N variables indicates the relative occurrence of cases that are characterized by each possible combination of values for the N control variables.

If N = 2, the distribution is referred to as bivariate. As an example, we might consider the distribution of accident cases as a function of driver sex and of light condition (day, night, or dusk). A bivariate frequency table would then have the form:

	Male	Female
day	N_{11}	N_{12}
dusk	N_{21}	N_{22}
night	N_{31}	N_{32}

where the N_{ij} correspond to the case counts for the particular combination of parameters. That is, N_{22} represents the number of accidents involving a female driver at dusk, etc. The bivariate frequency concept is thus a logical extension of the histogram or one-way distribution concept and represents a more sophisticated breakdown of the factors involved.

The basic program output is thus bivariate frequency tables that represent the case counts obtained for chosen values of the row and column variables. In addition, a number of statistical parameters relevant to the frequency table may be computed. One-way percentage distributions that are based on either row or column totals can also be obtained as well as two-way percentage distributions based on grand totals. Missing data codes may be included or excluded in these percentage calculations.

A more complete discussion of the mathematical background concepts involved is presented in Appendix C.

B) SRS FILE STRUCTURE

HSRI accident files are stored on magnetic tape. The control operations necessary to handle the tape and read the information are performed by the SRS so that the user has no need to know the detailed structure of the files. There is a file structure concept that must be understood for successful use of the accident analysis system, however.

Each entry in the file is called a "CASE." In SRS data files, a CASE may refer to a variety of events, physical objects, or actions. In the most common situation, referred to as an ACCIDENT FILE , each accident is a case entry into the file. More often than not, of course, more than one vehicle is involved in an accident. It is possible to create a VEHICLE FILE by making each vehicle involved in an accident correspond to one case; descriptive information concerning the accident scene is repeated for each vehicle in a given accident. Again, each vehicle may contain several occupants and each occupant may be considered as a case (the OCCUPANT FILE). Finally, at the highest level of division usually found, each occupant (in each vehicle in each accident) may have several injuries. Using each separate injury as a case leads to the INJURY FILE concept.

Within each case, the descriptive information is coded into possible alphabetic or numeric values of certain VARIABLES that describe the case. This structure is illustrated in Figure 2 which shows a four-case file containing five descriptive variables. Here, the accident is characterized by the age and sex of the driver, the severity of the accident, the color of the striking vehicle, and the county where the accident took place. There is usually one variable for each question on an accident report. The number of variables (or pieces of information included) is on the order of one hundred for most accident files, but can go as high as 600-700 in the MDAI files where a large amount of detailed information is available.

	V1 DRIVER SEX	V2 DRIVER AGE	V3 SEVERITY	V4 VEHICLE COLOR *	V5 COUNTY
CASE 1	2	18	3	'bbbRED'	3
CASE 2	1	16	2	'bBROWN'	3
CASE 3	2	52	2	'ORANGE'	1
CASE 4	1	99	1	'bbBLUE'	1

Note: b = blank

CODEBOOK

V1 DRIVER SEX

1. Male
2. Female
3. Unknown

V2 DRIVER AGE

1
.
.
98

} Age in years

99 Unknown

V3 ACCIDENT SEVERITY

1. Property Damage
2. Injury Producing
3. Fatal
9. Unknown

V4 COUNTY

1. Oakland
2. Wayne
3. Macomb
9. Unknown

FIGURE 2. SAMPLE ACCIDENT FILE

The number of different code values that can be used to represent the possible states of a variable is determined by a quantity known as the FIELD WIDTH of the variable. Field width corresponds to decimal digits in a number: that is field width of one can store values from zero to nine; a field width of two can store values from zero to 99; etc. For instance, driver sex (Variable One in Figure 2) has a field width of one and is usually characterized by the three values

(1) Male

(2) Female

(3) Missing Data

The variable describing the age of the driver in years, on the other hand, has a field width of two characters and code values then can range from 00 to 99. The two variables we have discussed here have numeric code values (i.e., 1-3 or 0-99). Numeric codes are most useful for digital computers where arithmetic operations can be handled easily and quickly. However, some variables in the HSRI files have alphabetic code values. An alphabetic code value is denoted by primes before and after the value (i.e., 'RED ', 'FORD ', etc.) Thus we might have an alphabetic variable with a field width of six that could take on possible values from 'AAAAAA' to 'ZZZZZZ' (including blanks). The use of such a variable for car color could then have the code values

'bbbRED' = RED

'bBROWN' = BROWN   b = blank

'ORANGE' = ORANGE

etc.

as in Variable 4 of Figure 2. Most HSRI programs will only accept numerical codes for analysis purposes, although alphabetic codes can be used for filtering/subsetting purposes or recoded to numeric values (see Sections 2-D or 2-E).

Three types of data may be assigned to variable code values: numerical, ordinal, or categorical.

Numerical data involves the coding of arithmetic information. Thus in the age variable V2 (Figure 2), there is a direct correspondence between driver age and code value (code value 52 for Variable 2 represents a person whose age is 52 years). By performing arithmetic operations on the code values (summary, averaging) we obtain quantities that are meaningful in a physical sense.

Ordinal data deals with information that can be ordered--as numbers can--but cannot be assigned a specific value that is meaningful. For instance, a typical accident severity variable might have the code values:

1	=	Property Damage
2	=	Injury
3	=	Fatality

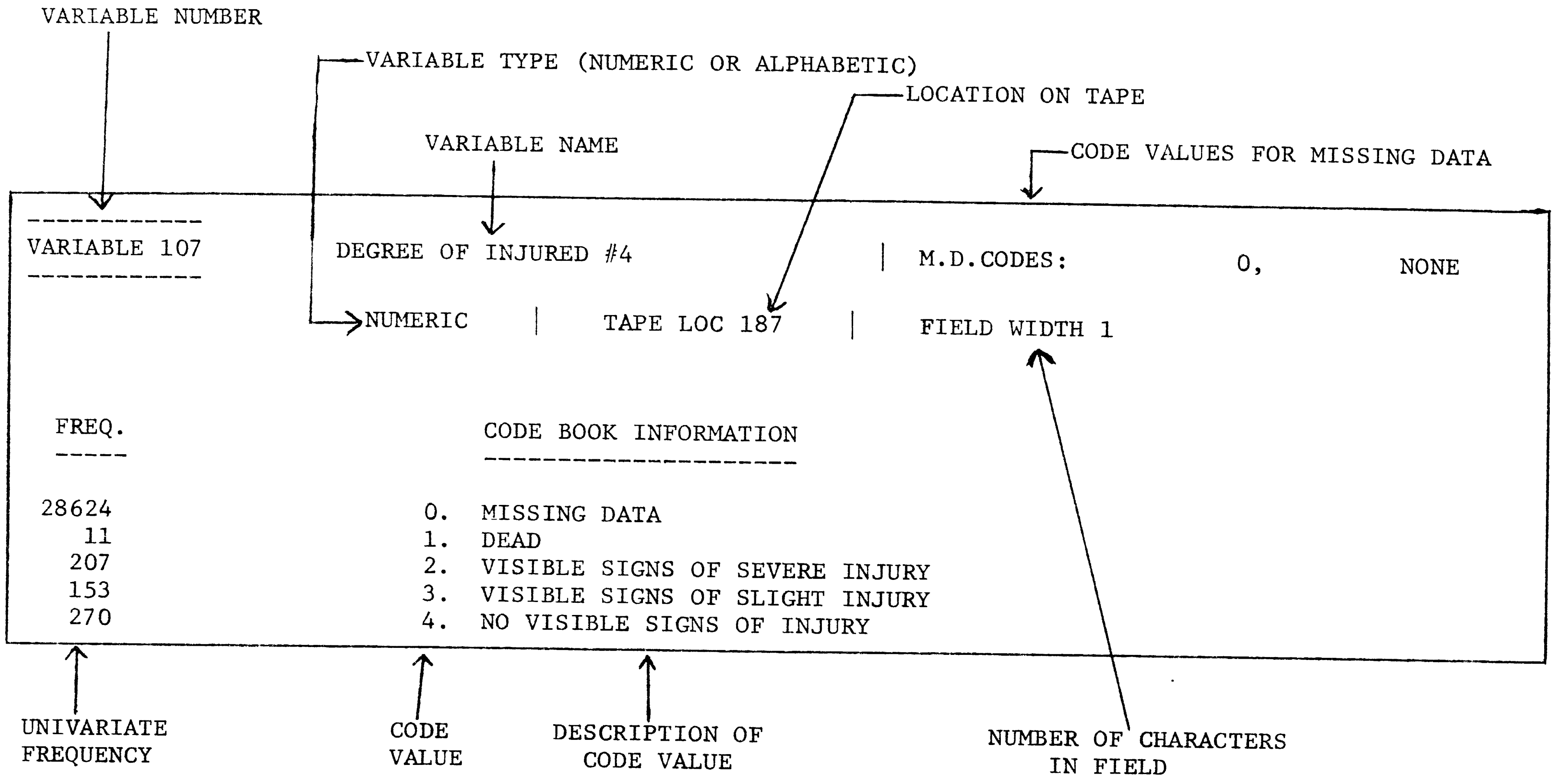
as shown in Variable 3 in Figure 2. Clearly, code 3 is more severe than code 2 which in turn is worse than code 1 (in some sense). The code values are therefore ordered in some severity scale. However, we are not saying that injuries are twice as severe as property damage events or two-thirds as severe as fatalities. Similarly, if we found that the <u>average</u> accident severity was 2.6 for a certain class of accidents, we would know that a typical accident is somewhat worse than one involving just injuries, but that someone is not always killed. It is not meaningful to say, however, the typical accident is 2.6 times as severe as a property damage event.

Finally, categorical data is information that cannot be ordered. For instance, a "COUNTY" variable (V5 of Figure 2) might contain a correspondence between codes values and an alphabetic listing of counties; that is:

Code Value 1	=	County A
Code Value 2	=	County B
Code Value 3	=	etc.

Here County B is not necessarily twice as big, as important, or as rich as County A. There is simply no direct numerical relationship among data values. The average county for a certain class of accident is similarly meaningless. This type of information is descriptive and cannot be compared numerically.

In the file structure, variables are characterized by a variable number (i.e., V29) and a field width. The correspondence between variable number and variable name, as well as the relationship between variable code values and descriptive factors of the accident, is expressed in the CODEBOOK for a given file. Figure 3 shows an entry from the codebook of a typical HSRI file.



```
-------------
VARIABLE 107        DEGREE OF INJURED #4                 | M.D.CODES:          0,        NONE
-------------
                    NUMERIC   |   TAPE LOC 187   |   FIELD WIDTH 1


   FREQ.                           CODE BOOK INFORMATION
   -----                           ---------------------

  28624                   0.  MISSING DATA
     11                   1.  DEAD
    207                   2.  VISIBLE SIGNS OF SEVERE INJURY
    153                   3.  VISIBLE SIGNS OF SLIGHT INJURY
    270                   4.  NO VISIBLE SIGNS OF INJURY
```

VARIABLE NUMBER
VARIABLE TYPE (NUMERIC OR ALPHABETIC)
LOCATION ON TAPE
VARIABLE NAME
CODE VALUES FOR MISSING DATA
UNIVARIATE FREQUENCY
CODE VALUE
DESCRIPTION OF CODE VALUE
NUMBER OF CHARACTERS IN FIELD

FIGURE 3. TYPICAL HSRI ACCIDENT FILE CODEBOOK ENTRY

C) VARIABLE LISTS

Use of the SRS programs frequently requires the specification of a variable list; that is, a definition of the variables to be used in program operation. The Data Set List Program, for example, requires a variable list to specify the variables that will be listed in the program output.

In SRS format a variable list is simply a string of variable numbers, each preceded by a "V" and separated by commas. A hyphen may be used to indicate an inclusive range of consecutive variable value. The list is ended with an asterisk (*). A typical variable list is:

```
V1, V3, V5 - V9, V12, V14-V28*
```

This would include only variable numbers V1, V3, V5, V6, V7, V8, V9, V12, and V14 through V28 in the analysis.

D) FILTER STATEMENTS

In many cases the accident files contain much more information than is necessary for the investigation to be performed. The use of data filters permits the inclusion of data that is pertinent or the exclusion of data that is not pertinent. Thus, filters are used on a data file to subset or retrieve just those cases of interest for analysis.

A filter statement is a phrase that may be up to three 80-character lines long or less, that begins with the words "INCLUDE" or "EXCLUDE," and that ends with an asterisk (*). In the statement, variables are denoted by the letter "V" and the variable number (i.e., V23). Code values are expressed as unique values (i.e., V45 = 'CHEV'); as unique values separated by commas (i.e., V45 = 'CHEV', 'FORD'); as a range of consecutive values whose limits are separated by a hyphen (i.e., V23 = 10-15); or as a combination of these conventions (i.e., V23 = 01, 02, 04, 12-24). Note that if the field width of the variable is greater than the number of digits in the code value, the extra digits must be padded with zeros. Up to 15 variables may be included in the filter statement.

The variable statements are combined with "AND" and/or "OR" to provide the desired filter action. The "AND" statement implies both variables listed must have the selected values; "OR" implies either may have the selected value. In addition, the "AND" relationship has precedence over the "OR". Up to 14 and/or connectives may be used in a filter statement. A summary of the filter command statements format for the SRS is given in Table IV.

In order to clarify the filtering concept, a number of examples are presented below.

EXAMPLES:

EXCLUDE V16=5 AND V14=1, 7-9*

This excludes those cases for which both variable 16 is coded 5 and variable 14 is coded 1,7,8, or 9.

INCLUDE V66=05,22 AND V52=004*

This includes only those cases for which variable 66 (a two-digit variable) has a code of 5 or 22 and variable 52 (a three-digit variable) has a code of 4.

INCLUDE V16=5 OR V14=9*

This includes only those cases for which either variable 16 is coded 5 or variable 14 is coded 9 or both.

INCLUDE V16=5 AND V14=9 OR V16=6*

This includes only those cases for which either both V16 is coded 5 and V14 is coded 9, or V16 is coded 6.

TABLE IV

SRS FILTER STATEMENT SUMMARY

INCLUDE	Includes only the cases that are specified
EXCLUDE	Includes all cases but the ones specified
V	Prefix for variable number (i.e., V23)
C1,C2, •••,	C3,C4-C5,C6-C7, ••• C8-C9 Format for acceptable code values
AND	Includes or Excludes only those cases for which both conditions are true
OR	Includes or Excludes only those cases for which either condition is true
*	Terminates a filter statement

NOTES

1) AND has precedence over OR

2) The code values specified in the filter statement must have the same field width as the variable they describe (check the file codebook).

STANDARD FORM

```
INCLUDE                            AND                            AND
         VN1=code value list 1          VN2=code value list 2          •••
EXCLUDE                            OR                             OR


AND
     VNn=code value list n *
OR
```

where:

N1,N2,•••,Nn are n variable number used in the filter (n ≤ 14)

••• represents the intervening records.

```
INCLUDE V16=5 AND V14=9 AND V66=05 OR V16=5 AND V14=9 AND V52=004*
```

If the user desires those cases where V16=5 <u>and</u> V14=9 <u>and either</u> V66=05 <u>or</u> V52=004, enter the above filter. Remember that AND takes precedence over OR so "INCLUDE V16=5 AND V14=9 AND V66=05 OR V52=004*" would not select the cases desired, e.g., all cases where V52=004 would be included.

```
INCLUDE V54='RED ','BLUE' AND V129='FORD'*
```

This includes only those cases where V54 is coded 'RED 'or 'BLUE' and V129 is coded 'FORD'.

```
EXCLUDE V78='CHEU' - 'CHEZ'*
```

This excludes all cases where V78 is coded 'CHEU', 'CHEV', 'CHEW', 'CHEX', 'CHEY', or 'CHEZ'.

E) RECODE STATEMENTS

Many analysis tasks are facilitated by the recoding of variable values: that is, the substitution of new code values for the permanent ones that exist in the accident file without permanently creating a new variable. For instance, a variable that specifies driver age by years might be recoded by the correspondence

00-09 → 01

10-19 → 02

etc.

In other cases, recoding provides a means of circumventing certain program restrictions. For example, the bivariate program only accepts a column variable with a field width of one. Portions of a variable with a field width of two can be used in a bivariate by recoding the desired portion into the range of 0 to 9. A final very important use of the recode option is the conversion of alphabetic code values to numeric values in order to permit the use of analysis programs on otherwise unusable variables.

Note that the recoding is temporary. The analyst does not have to permanently alter the data file by computing and storing a new variable. This option permits a dynamic recoding, i.e., good only during program operations. The modifications are lost when execution is terminated.

Because Data Set List is not an analysis program, the RECODE option is not valid. The RECODE statement may be used with all other SPAD analysis programs.

Each recode statement must begin with the word "RECODE" and end with an asterisk (*). The recode statement has the generic form

RECODE VN (ORIGINAL VALUE LIST 1) = NEW VALUE 1, (ORIGINAL
VALUE LIST 2) = NEW VALUE 2, •••, (ORIGINAL VALUE LIST n-1) =
NEW VALUE n-1, (ELSE) = NEW VALUE n*

The term "VN" is the standard designation for variable number N with the prefix letter "V". Each original value list (enclosed in parenthesis) indicates the original accident file code values that are to be given the new value indicated. This list has the same form as the acceptable code values in a filter statement. Thus, commas separate unique values, and a hyphen indicates ranges of values. The "ELSE" operand in the initial value list assigns any unspecified original values to the specified new value and must occur at the end of a statement if it is used. The "ELSE" statement prevents new recoded values from overlapping permanent original values. "ELSE" is helpful in recoding wild or stray original values to an "Unknown" code value, and must be used with alphabetic variables.

The temporary recode option is valid only for the recoding of numerics to numerics, and alphabetics to numerics.

Each recode statement specifies the code translation of a single variable, but up to five statements may be used in one program operation; however, there is an overall restriction that no more than ten 80 character lines be used for all recodes. The statements may be entered in any order, but the actual operation will be performed sequentially in the order indicated by their entry. Finally, the field widths of the initial and final code value values must correspond to the field width of the variable as specified in the file codebook (see, for example, Figure 3).

A summary of the RECODE statement parameters is presented in Table V and a number of practical examples are given below.

RECODE EXAMPLES:

```
RECODE V87 (00-09) = 01, (10-19) = 02, (20-29) = 03, (30-39) = 04,
(40-49) = 05, (ELSE) = 06*
```

This example brackets the values into groups of ten and lumps all values of 50 or greater into a single value.

```
RECODE V13 ('WHITE') = 00001, ('YELLO') = 00002, ('GOLD ')=00003,
(ELSE) = 00009*
```

This recodes alphabetic color information into numeric values. Note that field width of new values equals original width.

```
RECODE V33 (0,1,8) = 9*
```

This converts all 0, 1, or 8 code values to 9 and leaves the remainder unchanged. No "ELSE" is used.

```
RECODE V17 (00-04) = 01, (05-09) = 02, (10-14) = 03, (15-19) = 04,
(20-24) = 05*
```

Recodes code values below 25 into 5 unit increments.

TABLE V

SRS RECODE STATEMENT SUMMARY

RECODE	First term in a recode statement.
V	Prefix for variable number (i.e., V23)
C1,C2,C4-C6,•••	Form of initial code value list.
(Initial code value list)=final code value	Code value recode specification.
ELSE	Represents all codes not specified explicitly by a recode specification.
,	Separates individual recode specifications.
*	Terminates a recode statement.

NOTES

1) ELSE, if used, must be the last recode specification in a statement.

2) The code values specified as initial or as final values in the recode statement must have the <u>same field width</u> as the variable they describe.

STANDARD FORM

RECODE VN (Original value list 1)=New value 1, (Original value 2)=
New value 2, •••, (Original value n)=New value n, (ELSE)=New value*

where:

N is the variable number

n is the number of the last set of values to be recoded

•••represents the intervening records

F) LABEL STATEMENT

Provisions are made on SRS to label the output information. A label of up to 80 characters may be used for this purpose. Any combination of letters and numbers may be used, but the label must not begin with "INCLUDE," "EXCLUDE," or "RECODE." The label may be left blank by entering an end of line code (CNTR-S or RETURN) when a label is requested.

The label is very useful where several persons share one computer user number. It is recommended that labels contain a name or initials, date, and subject title for a program run.

G) MULTIPLE RESPONSE VARIABLES

The majority of accident data variables encountered in SPAD files are denoted as single response. That is, each case entry in the file is allowed only one coded value for a specific variable. For example, a variable documenting lap belt usage might have two possible code values to record whether the belt was worn or not; however, for a given case only one of these values could be entered in the data file.

The SRS provides for the use of variables that are denoted as multiple response. Such variables allow more than one valid coded entry to be recorded for a single case. Consider a variable used to record the object struck by the case vehicle in an accident. If this variable were single response, then only one object could be coded: this might be the first object struck,the object that caused major damage to the vehicle, or any other object that fulfills some criteria related to the accident. The use of a multiple response variable (or MRV), however, would permit the investigator to record a variety of objects struck. For instance, a simple list of struck objects could be entered in any order, or a chronological ordering of struck objects could be used for successive responses of the variable. The use of MRVs in accident files therefore permits additional information to be incorporated into the file without the complicating necessity of using multiple variables. This technique has many advantages in data analysis procedures.

Because MRVs are different from single response variables in many respects, there are special considerations that must be observed in their use. A number of these considerations will be discussed below.

When using a MRV in an SRS global filter (i.e., Section 3-D), a case will pass the filter (that is, be included or excluded) if <u>any</u> response satisfies the filter condition. This means that some uses of filter statements for single response variables are inappropriate for MRVs so that careful consideration must be given to the exact filter action. Some examples of filter action are shown below for a two response variable (denoted by variable number VN).

GLOBAL FILTER	CASE VALUES FOR VN		RESULT
INCLUDE VN=5,7*	1	8	Excluded
INCLUDE VN=5,7*	1	5	Included
INCLUDE VN=5,7*	7	3	Included
EXCLUDE VN=1-3,5-9*	4	2	Excluded
EXCLUDE VN=1-3,5-9*	8	4	Excluded
EXCLUDE VN=1-3,5-9*	4	4	Included

Note that the discussion above applies exclusively to <u>global filters</u>; that is filters that apply to the entire data set used for analysis. These global filter statements are entered in response to the FILTER(RECODE OR LABEL) request. Certain programs have a provision for local filters (Histogram, Analysis of Variance, and Bivariate Frequencies). It is important to note that MRVs may not be used in local filters.

In a strict sense, MRVs cannot be recoded. If Recode is used, only the first response is recoded; the remaining responses are unaltered by the operation. Consequently, RECODE may be utilized for certain particular operations if this restriction is understood.

When using MRVs in SPAD programs, only Data Set List, Means and Marginals, and Bivariate Frequencies handle all responses correctly. The other programs use the first response only.

A summary of the system features pertinent to multiple response variables is shown in Table VI.

Table VI

SPAD Program Multiple Response Variable Summary

Program	Recode Capability	Global Filter Capability	Analysis Operation
Data Set Listing	No recode possible	MR Variables handled	Non-Analysis program, but all responses will be listed out
Histogram	First response only	MR variables handled	First response only
Means and Marginals	First response only	MR variables handled	All Responses
Analysis of Variance	First response only	MR variables handled	First response only
Bivariate Frequencies	First response only	MR variables handled	All Responses

SECTION 4

HSRI ACCIDENT FILES

A total of twenty-four accident data files are currently (October 1972) available to users of the SPAD system. These files are based on eight mass accident data sources, and offer a variety of accident analysis opportunities.

The following descriptions define each accident data source and the available files in terms of representative years, data origin, case count, variable count, investigation level, and unique file biases and attributes.

WASHTENAW COUNTY, MICHIGAN ACCIDENT AND DRIVER RECORD FILES:

Data obtained from the Ann Arbor City Police Department, Washtenaw County Sheriff's Office, and the Michigan State Police has been built into a Level I accident file. The file is especially useful for file interrogations of a broad area representing urban, rural, academic, and industrial populations.

The Michigan Secretary of State authorities have furnished driver record data for Washtenaw County, which have been built into a driver file. The file is valid for each driver at least through 1969.

File Statistics:

File	Years	Number of Cases	Number of Variables
Washtenaw County	1968-1971	28,969	159
Washtenaw Driver File	through 1969	17,989	48

OAKLAND COUNTY, MICHIGAN ACCIDENT FILES:

Accident information furnished by the Michigan State Police and Traffic Improvement Association of Oakland County has been built into Level I accident files for the 1968, through 1972 calendar years. The 1968 through 1971 files contain additional variables which further document occupant statistics. Each data entry represents a single accident situation, and the file allows analysis of a combined urban and rural locality.

File Statistics:

File	Years	Number of Cases	Number of Variables
Oakland Co. 1968	1968 calendar year	25,387	120
1969	1969 calendar year	29,265	179
1970	1970 càlendar year	29,650	156
1971	1971 calendar year	29,362	187

CPIR REVISION 2 AND 3 VEHICLE, OCCUPANT, AND INJURY FILES:

Data based on the General Motors Collision Performance and Injury Report (Long Form) Revision 2 and 3 have been built into two sets of Level III files. Accident information obtained from Revision 3 report forms have been built into vehicle, injury, and occupant files, while the Revision 2 data have been built into a vehicle and an occupant file.

Revision 3 data originate from reports submitted by Multidisciplinary Accident Investigation teams sponsored by the National Highway Traffic Safety Administration and the Automobile Manufacturer's Association. A significant sampling bias accompanies the case study investigation opportunities that the file offers. Since the data are obtained from case studies, the file is definitely biased, and does not represent a sample of cases from each submitting area.

Accident investigation funded by the Automobile Manufacturer's Association and conducted by Donald Huelke at University of Michigan and Arnold Siegel of the Trauma Research Group at UCLA is the source of the Revision 2 data. Like the Revision 3 data, the Revision 2 data are also biased in that they reflect case studies , rather than a sample of cases in one locale.

File Statistics

File	Years	Number of cases	Number of Variables
CPIR 3 Vehicle	1969 to present	2543	576
Occupant		4169	636
Injury		14,485	647
CPIR 2 Vehicle	1967-1969 (overlap with CPIR 3)	716	320
Occupant		1162	507

CORNELL LEVEL I ACCIDENT AND LEVEL II ACCIDENT, VEHICLE, AND OCCUPANT FILES:

Accident information from police reports, driver records, and vehicle registration records has been build into a Level I accident file representing the eight western counties of New York State.

Level II data from police reports have been built into accident, vehicle, and occupant files. Both Level I and Level II data were collected by Cornell Aeronautical Laboratories, funded by the Automobile Manufacturer's Association. All four files are useful for analyses of a large multi-county area.

File Statistics:

File	Years	Number of cases	Number of Variables
Level I Accident	1970 calendar year	39,992	159
Level II Accident	October 1970 - March 1972	11,503	32
Vehicle		22,289	65
Occupant		31,462	80

BEXAR COUNTY, TEXAS ACCIDENT AND VEHICLE FILES:

Data for the San Antonio and entire Bexar County, Texas area have been built into Level I accident and vehicle files for the 1969 and 1970 calendar years. The data were obtained from the Texas Department of Public Safety. Since Bexar County parallels Oakland County, Michigan in size and composition, the file offers comparative analysis opportunities.

File Statistics:

File	Years	Number of Cases	Number of Variables
Bexar Co. 1969 Accident	1969 calendar	26,673	56
Vehicle	year	45,859	139
1970 Accident	1970 calendar	27,458	56
Vehicle	year	47,284	139

DADE COUNTY, FLORIDA ACCIDENT FILES:

The Metropolitan Dade County Public Safety Department has furnished accident data for the metropolitan Miami and entire Dade County, Florida area. The data have been built into Level I accident files for the 1969 and 1970 calendar years. While each accident case is not documented to the extent of the Washtenaw County or Oakland County, Michigan files, the Dade county file is useful for analyses of a metropolitan area. The user should take the file size into consideration of the analysis costs involved.

File Statistics:

File	Years	Number of Cases	Number of Variables
Dade County 1969	Jan. 1969-June 1969	31,056	83
1970	1970 calendar year	61,767	83

DENVER COUNTY, COLORADO ACCIDENT FILES:

Level I accident files for the 1969 and 1970 calendar years have been built from data obtained from the State of Colorado Department of Revenue. The data represent the Denver city and entire Denver County, Colorado area. The file offers analysis capabilities for a metropolitan area, and contains extensive variable documentation for each accident case entry.

File Statistics:

File	Years	Number of cases	Number of variables
Denver County 1969	1969 calendar year	25,581	234
1970	1970 calendar year	29,432	217

SEATTLE, WASHINGTON ACCIDENT FILE:

The state of Washington has furnished 1969 calendar year data that have been built into a Level I accident file. The file represents the greater Seattle area, and is especially useful for vehicle model size and type analysis, because of the extent of make/model category variables.

File Statistics:

File	Years	Number of cases	Number of Variables
Seattle 1969	1969 calendar year	26,000	194

SECTION 5

OPERATING THE SPAD SYSTEM

The information presented in earlier sections of this manual provides the background for operating SPAD. This section deals with the actual operational considerations involved.

A) SPAD COMMAND STRUCTURE

The basic SPAD system instruction is the MTS command

```
$SOURCE  HSRI:SPAD(N)
```

where N is a number that depends upon the operation to be performed as well as the data file to be utilized. The number N can be broken down into the arithmetic sum of three other numbers (D, A, and U):

$$N = D + A + U$$

where D depends only on the accident Data file to be accessed, A depends only on the Analysis operation to be utilized, and U depends on the Utility operation to be performed. These numbers are subject to some revision as the accident files are updated and modified. A listing of the currently applicable values are summarized in Section 5-P.

As we will see in the following discussion, three SPAD commands are used during the course of a typical simple analysis: the first instructs MTS to mount the proper magnetic tape on an available tape drive; the second initiates execution of the appropriate SRS program to perform the desired analysis; and the third instructs MTS to dismount the magnetic tape, thus freeing the tape drive. All commands are the same standard form--only the number N changes in accordance with the desired operation.

B) OPERATING SEQUENCE

A sequence of necessary operations to use the SPAD system is shown below:

1) CONNECT TERMINAL AND SIGN ON MTS.

2) MAKE A SETUP FILE (IF DESIRED).

3) CREATE OR EMPTY THE DATA STORAGE FILE "TBBL."

4) MOUNT THE ACCIDENT FILE TAPE.

5) RUN THE DESIRED ANALYSIS PROGRAM.

6) CHECK THE RESULTS FOR SUCCESSFUL OPERATION.

7) DISMOUNT THE ACCIDENT FILE TAPE.

8) PRINT THE RESULTS.

9) SIGN OFF MTS.

With the exception of Steps (1) and (9) which have been discussed in Section 2, the remaining items will be discussed in order below. The use of setup files (Step 2) is closely associated with particular analysis programs and will consequently be discussed after Step 5. In an actual operation, the order of events shown above is applicable.

C) DATA FILE "TBBL"

All SRS programs operating under SPAD control are designed to store their output results in a private MTS user file called "TBBL." Before making any run, therefore, it is necessary to create "TBBL" on the user account or to empty "TBBL" if it already exists and contains data from previous analysis operations. The MTS commands $CREATE TBBL or $EMPTY TBBL will accomplish this (see, for example, Section 2-C).

Remember that TBBL will exist permanently on the user account and incur storage charges. If this file is large (i.e., refer to * FILESNIFF) and will not be used for some time, it may be more economical to destroy TBBL and recreate it when necessary. This may be accomplished by the MTS command $DESTROY TBBL and $CREATE TBBL.

D) MOUNTING THE ACCIDENT FILE TAPE

It is always necessary to mount a magnetic tape containing the desired accident file on a tape drive at the Computer Center before making an analysis run. If this is not done, difficult-to-interpret error messages will be generated by MTS. Tapes can be mounted with the SPAD command

```
$SOURCE  HSRI:SPAD(N)
```

The operation code N is equal to the data file number D for tape mounts because the analysis and utility indices are both zero for the operation. That is

TAPE MOUNTS	N = D

Example (27) shows the mounting of a CAL Level 2 Vehicle file tape (D = 500).

```
#$SOURCE HSRI:SPAD(500)
#$RUN *MOUNT PAR=C0008 9TP *T* SIZE=3600 RING=OUT LP=OFF VOLUME=HSRCRN 'CORN1'
#EXECUTION BEGINS
C0008 9TP *T* SIZE=3600 RING=OUT LP=OFF VOLUME=HSRCRN 'CORN1'

*T*: MOUNTED ON T911
#EXECUTION TERMINATED
#
```

Example 28

Remember that it takes time for an operator to obtain the tape from a rack and physically mount it on an available tape drive!

The message *T*: MOUNTED ON T911 indicates that the tape has been successfully mounted. Two other situations commonly occur: 1) All the tape drives may be in use, or 2) Someone else may be using the tape requested. These conditions will result in the mount comments

```
*T*: ALL 9TP BUSY
```

or

```
*T*: TAPE NOT AVAILABLE
```

These messages may be worded differently, but their meaning is clear.

If the user does not receive a clear indication that the tape (designated by *T*) has been mounted on a drive, then the analysis task must be suspended (ALL 9TP BUSY) until a tape drive is available or redefined (TAPE NOT AVAILABLE) to access an available file.

In many cases, a number of different accident files are contained on the same magnetic tape. As an example, all five of the CPIR (Collision Performance and Injury Report) files are on a common tape. If more than one analysis is to be performed on different files that are physically on the same magnetic tape, then it is only necessary to use the mount command once: the analysis programs on the various files may be run consecutively with the appropriate SPAD command (see Section 5-E). Before running an analysis on data that is on another tape, however, it is necessary to dismount the current tape and to mount the new tape using the procedure discussed above. Mounting a new tape will automatically dismount the tape that is already mounted. MTS will notify the user of the actions taken via appropriate messages. A list of equivalent SPAD mount commands (i.e., files on the same tape) is presented in Section 5-P.

E) RUNNING THE ANALYSIS PROGRAM

Execution of the desired analysis program is performed by the standard SPAD command with D equal to the value used to mount the tape, A equal to the value for the desired analysis program and U equal to zero. Thus

```
ANALYSIS    N = D + A
```

An analysis of variance on the CAL Level 2 vehicle file tape mounted in Example 28 would be performed as in Example 29.

```
#$SOURCE HSRI:SPAD(505)
#$RUN HSRI:PG640 5=*SOURCE*+*MSOURCE* 8=TBBL
#EXECUTION BEGINS
  DDEF STATEMENT=

 UNIT 1 READING TAPE *T*            ,HSRCRN   ,FILE IS CRNVDIC   0  0  0

 UNIT 2 READING TAPE *T*            ,HSRCRN   ,FILE IS CRNVDAT   0  0  0
 FILTER (RECODE OR LABEL)=
```

Example 29

After the SPAD command is entered, the remaining output is automatic. The format will change, of course, depending on the analysis program utilized, but the general action will be as shown in Example 29.

After printing the filter request, control is returned to the user who must then enter a filter statement, recode statement, or data output label as defined in Section 3. The statements must be entered in the order: 1) Filter, 2) Recode, 3) Label. If a filter statement is entered, the program will return with a prompt RECODE (OR LABEL) =. After the fifth recode statement is entered, a prompt LABEL = will be issued. Any statement may be entered to any prompt, however, as long as they are in the proper order. Thus a recode or label may be given for a filter and a label may be given for a recode. Remember that RECODE cannot be used with Data Set List so that only filter and label statements are appropriate.

Because each analysis program has different input specifications, information entered by the user after the label request depends upon the program that is executing. At this point, then, the specific operational details of the five SPAD analysis programs will be considered in detail. Each program writeup is followed by a sample data output format (i.e., Figs. 4-8).

F) THE DATA SET LIST PROGRAM

ANALYSIS VALUE A=1

This program lists a subset of a data file and, optionally, its corresponding dictionary. The user can list a subset of the variables and a subset of the data using a filter statement. The data can be listed at intervals (e.g., every third case). Underlined responses are recommended for SPAD users.

EXECUTION:

0. $SOURCE HSRI:SPAD(D) D=Data File Value (Tape Mount)

1. $SOURCE HSRI:SPAD(D+1)

2. FILTER(OR LABEL)=

   Filter,or Label (See Section 3: Note no recode is possible with a Data Set List)

3. LABEL= (Prompted for only if the above was a filter)

   Label (See Section 3)

4. DESCRIPTION OF PARAMETERS?

   "YES" description printed out

   "NO" description not printed out

5. IFALL=(whether to print all variables)

   0 all variables are printed

   1 user will be prompted for a variable list.
   Only a subset of the variables will be printed.

6. LSTNTH=(whether to print all the cases found)

   1 or blank all cases found are printed

   5 every fifth case found is printed, etc.

7. LSTNTH=above answer IS THIS CORRECT?

   Answer: if answer is not "YES" then "LSTNTH" request will be repeated.

8. ISPACE=

   0 or blank single spacing

   1 double spacing

9. NODICT=

   0 list dictionary for variables printed

   1 dictionary not listed

10. IFTERM=

   0 print results off-line

   1 print results on-line

11. THE VARIABLE LIST IS:

   (Prompted for only if "IFALL" = 1)

   Supply variable list statement (See Section 3).

12. $ENDFILE

Output is a printed listing with the following characteristics:

(1) The label will be printed on the top of each page.

(2) Each record will have a sequence number to aid in identification.

(3) When all variables are listed, 120 characters per line will be printed.

When only a subset of the variables is printed, three spaces are placed between variables by the program. The number of variables per line depends on the width of the variables.

(4) Column and variable identification:

When all variables are listed, a column guide heading is printed over each record indicating every tenth column.

When only a subset of the variables is printed, the variable number is printed on every page to identify the variable.

RESTRICTION:

When listing a subset of variables, a maximum of 400 variables may be listed. (When listing all variables, there is no restriction.)

```
                               AA-147  SOUTH OF PACKARD ON CARPENTER
      1          3          9     10    11    13      14      15      16     17   18   19   20   28   32   33   35    37          VAR.#
  1   6902385    99999      05    09    69    2220    0216    1501    901    2    1    3    1    2    2    2    03    02
  2   6800872    99999      02    12    68    1715    0216    1501    905    2    1    1    1    2    2    1    02    01
  3   6904021    99999      08    04    69    1910    0216    1501    300    2    1    2    1    2    2    3    03    00
  4   6804471    99999      09    21    68    1330    0216    1501    903    2    1    1    1    2    1    1    01    00
  5   7004620    99999      08    29    70    1245    0216    1501    902    2    2    1    2    1    1    0    06    00
  6   6900425    99999      01    17    69    1951    0216    1501    902    2    2    3    2    1    2    0    06    00
  7   6804663    99999      10    03    68    1630    0216    1501    600    2    1    1    1    2    1    2    03    00
  8   6903848    99999      07    25    69    1455    0216    1501    903    2    1    1    1    2    1    1    01    00
  9   7003239    99999      06    12    70    1745    0216    1501    200    2    1    1    2    2    2    3    03    00
 10   6905855    99999      11    04    69    1030    0216    1501    300    2    2    1    2    2    2    3    03    00
 11   7006291    99999      11    14    70    1315    0216    1501    500    2    3    1    2    2    1    1    03    03
 12   6903089    99999      06    14    69    2110    0216    1501    905    2    1    3    1    2    2    2    01    03
 13   7001622    99999      03    14    70    1455    0216    1501    500    2    1    1    1    2    3    2    03    03
 14   7005585    99999      10    12    70    2325    0216    1501    906    2    2    1    2    1    1    0    07    01
 15   7006550    99999      11    25    70    1228    0216    1501    500    2    1    1    3    2    1    1    01    05
 16   6903717    99999      07    19    69    1630    0216    1501    905    2    2    1    2    2    1    2    03    00
 17   7004343    99999      08    13    70    1420    0216    1501    902    2    1    1    1    2    1    1    01    00
 18   6902079    99999      04    22    69    0845    0216    1501    150    2    2    1    3    1    2    0    06    00
 19   7007078    99999      12    16    70    9999    0216    1501    903    2    3    1    3    2    1    2    04    00
 20   6806417    99999      12    23    68    1408    0216    1501    150    2    3    1    2    2    1    1    01    00
 21   6903198    99999      06    20    69    1715    0216    1501    903    2    2    1    2    2    2    3    04    00
 22   7004810    99999      09    08    70    1900    0216    1501    901    2    1    2    1    2    2    2    03    00
 23   6903392    99999      07    01    69    1835    0216    1501    300    2    1    1    1    3    1    1    01    00
 24   7006476    00138      11    24    70    1815    0216    1501    500    2    1    3    1    2    1    3    03    00
 25   7005517    99999      10    09    70    1740    0216    1501    901    2    2    2    2    2    1    1    01    00
```

FIGURE 4 Sample Data Set List Output

G) THE HISTOGRAM PROGRAM

ANALYSIS VALUE A=2

The histogram program uses one variable to produce a graph on a 50 by 100 grid representing the relative frequencies of the variable. The vertical bar represents the number of cases falling within an interval, and the horizontal bar represents the intervals by either 3" + "s or 3" -"s beginning with the first interval and continuing to the thirty-third interval. The mean, variance, standard deviation, and median are optionally listed below the histogram. The intervals and their respective frequencies appear in a summary of all the histograms produced at the end of the job. A maximum of 15 histograms will be produced in one pass over the data, and up to 100 can be requested in a single run.

In the event that the variable used to produce the histogram cannot "fit" into 33 intervals when assuming a class width of one, then LWIDTH, the class width, must be specified by the user. The class width provided by the user is used to compute the lower bound and upper bound of each interval.

The user has the capability to apply local filters (see cols. 41-70 of Histogram Table Line description) to each histogram specifying a subset of the data to be analyzed.

RESTRICTIONS:

1. Maximum number of intervals produced is 33.
2. A maximum of 100 histograms can be produced per run.
3. A maximum of 15 histograms will be produced per pass over the data.
4. A variable may be used in more than one histogram.
5. The variable list must include any weight variables that are used.
6. Missing data are deleted from the histogram.
7. The computation of the median assumes that the midpoint of the frequency lies beyond the first interval. In the event that 50% or more of the frequency is in the first interval, the comment MEDIAN=0.0 will be presented and processing will continue.

EXECUTION:

0. $SOURCE HSRI:SPAD(D) D=Data File Value (Tape Mount)

1. $SOURCE HSRI:SPAD(D+2)

2. FILTER(RECODE OR LABEL)=

   Filter, Recode, or Label (See Section 3)

3. RECODE(OR LABEL)= (Prompted for only if the above was a filter)

   Recode or Label (See Section 3)

4. LABEL= (Prompted for only if the above was a recode)

   Label (See Section 3)

5. DESCRIPTION OF PARAMETERS:

   YES if format is to be described
   NO if format is not to be described

6. THE VARIABLE LIST IS:

   Variable list statement with all variable numbers to be used, including the weight variable if used.

7. HISTOGRAM CARD ON UNIT 4?

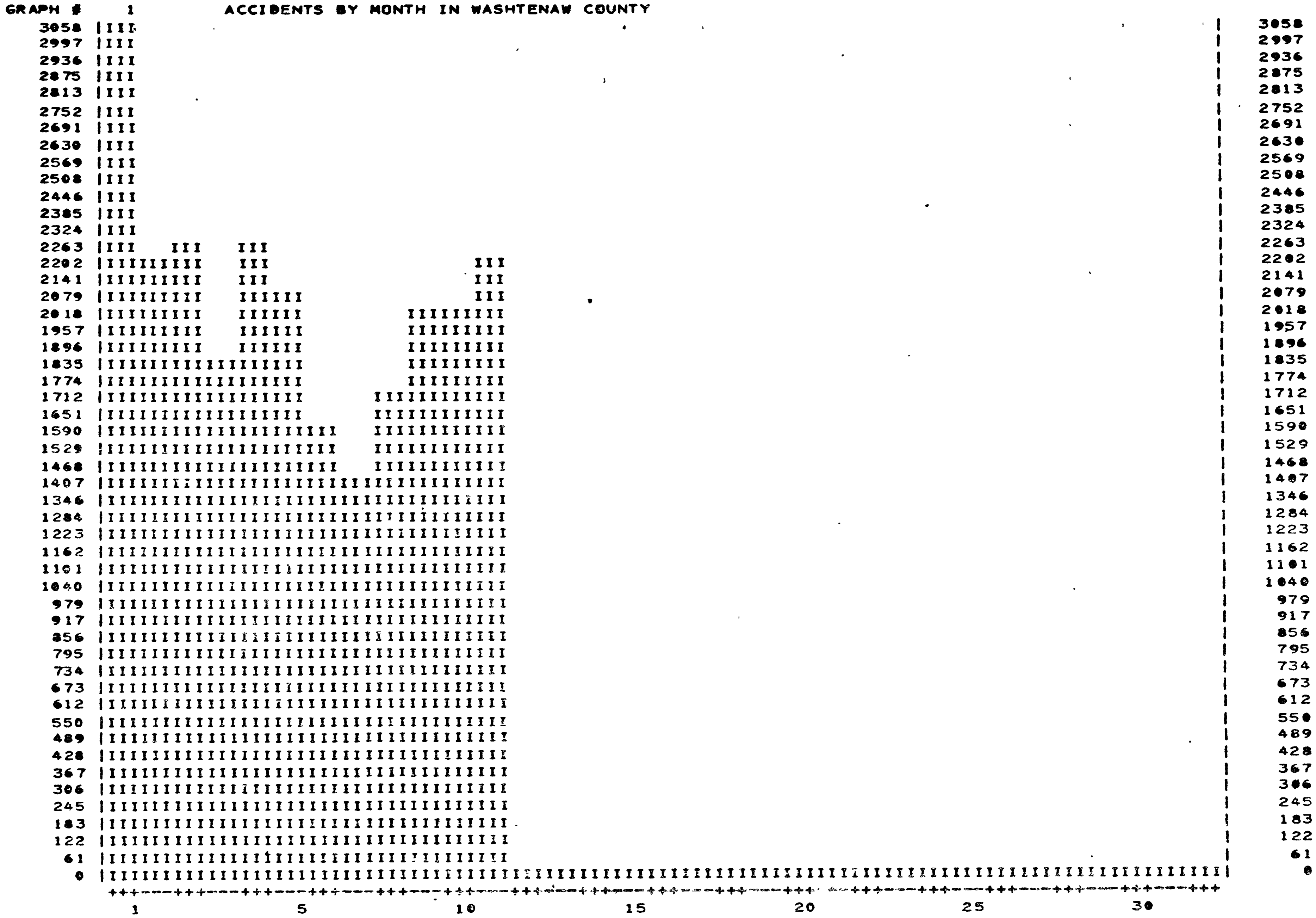
   NO

8.

....5	...10	...15	...20	...25	...30	...35	...40	...45	...50	...55	...60	...65	...70
ID#	VAR #	MIN	MAX	WID	WT	MED		FV1	FV1 MIN	FV1 MAX	FV2	FV2 MIN	FV2 MAX
Histogram Number - unique for each histogram (e.g., 1, 2...)	Variable Number to be used for histogram Note that a V does not precede number.	Minimum value of variable in integer mode (123 not 1.23)	Maximum value of variable in integer mode	Width of each interval when data is grouped (MAX-MIN+1)/33	Variable number to be used as a Weight variable	1 to compute median 0 do not compute median		First Filter variable used to create a subset of the data for this histogram (blank if none)	Minimum value of first filter variable if FV1 used (inclusive between minimum and maximum)	Maximum value of first filter variable, if FV1 used	Second filter variable for this histogram (blank if not used)	Minimum value of second filter variable, if FV2 used (inclusive between minimum and maximum)	Maximum value of second filter variable if FV2 used

....5 ...10 ...15...20...25...30...35...40...45...50...55...60...65...70
One line per histogram statement; up to 100 lines

9. $ENDFILE



```
VAR. NO.   9     MEAN =  6.183      VARIANCE =          13.154     STD. DEV. =      3.627      VALID CASES\ 25072
NUMBER CASES >IMAX OR <IMIN =       1
TOTAL CASES READ=   25079     TOTAL NUMBER OF CASES IN HISTOGRAM=   25072     MEDIAN =   0.0
```

Figure 5 Sample Histogram Output

```
*************************************************************************************************************************
```

H) MEANS AND MARGINALS PROGRAM

ANALYSIS INDEX A=4

The purpose of this program is to provide case counts, sum of weights, number of missing data cases for one or two missing data codes, ranges, means, standard deviation, skewness, and kurtosis for a user-supplied list of variable numbers for a standard data file. In addition, during a second pass of the data the program accumulates marginal distributions for the same variables, and optionally computes percentages on the marginals.

RESTRICTIONS:

(1) Codes may be up to 5 digits, and may start and end at any value (including negative values).

(2) The absolute difference between the maximum and minimum code value cannot exceed 24,400.

(3) The number of variables cannot exceed 200.

The missing data codes are included in the distribution range. These distributions may be weighted. Distribution percentages can optionally be computed and are printed out below the marginal for each code value. For each variable, starting with the lowest code, 15 consecutive code values are printed per line with the marginals (and percentages) underneath the respective code value. Each consecutive code value in the range is printed out until the maximum code value is reached.

EXECUTION:

0. $SOURCE HSRI:SPAD(D) D=Data File Value (Tape Mount)

1. $SOURCE HSRI:SPAD(D+4)

2. FILTER(RECODE OR LABEL)=

   Filter, Recode, or Label (See Section 3)

3. RECODE(OR LABEL)= (Prompted for only if the above was a filter)

   Recode or Label (See Section 3)

4. LABEL= (Prompted for only if the above was a recode)

   Label (See Section 3)

5. DESCRIPTION OF PARAMETERS?

   "YES" parameters are printed out
   "NO" parameters are not printed out

6. NOMEAN=

   0 mean statistics printed off-line (printed into your file)

   1 no statistics, program skips to "IFMARG", line 8.

   2 mean statistics printed on-line (printed out on terminal)

7. IFWT=

   0 no weighting of means

   N weight variable number (ex.: 1,5,96)

8. IFSKWT=

   1 computes skewness and kurtosis

   0 no computation

9. IFMARG=

   1 computes marginals

   0 no computation, program skips to Variable List, line 11.

10. IFWTMG=

    0 No weighting of marginals

    N Weight variable number

11. IFPERT=

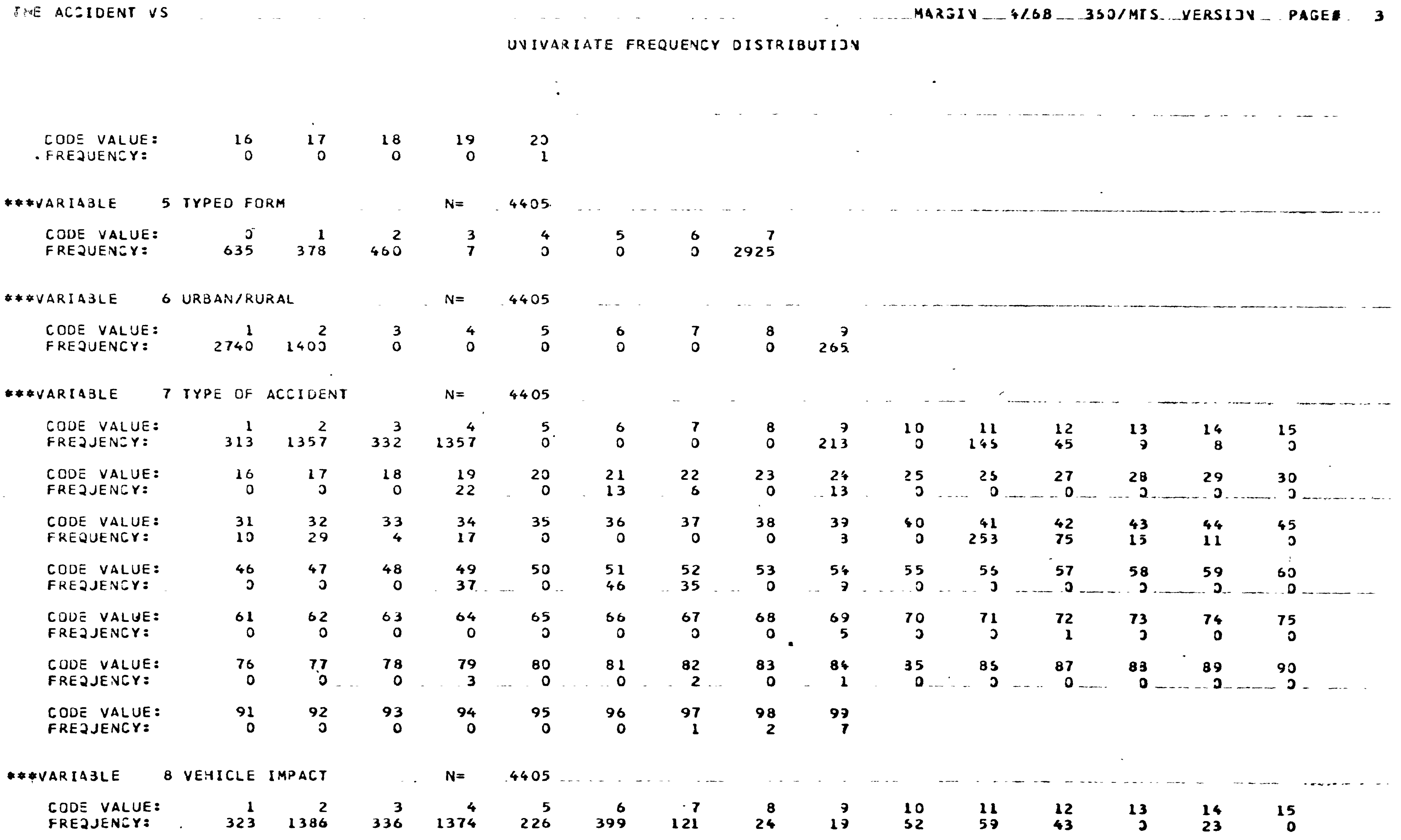
    1 Computes distribution percentages

    0 No percentages

12. THE VARIABLE LIST IS:

    Include all variable numbers to be processed, including any weight variables. (Cannot exceed 200 variables)

13. $ENDFILE



```
THE ACCIDENT VS                                                                    MARGIN   4/68   360/MTS  VERSION   PAGE#   3

                                          UNIVARIATE FREQUENCY DISTRIBUTION


     CODE VALUE:        16      17      18      19      20
     FREQUENCY:          0       0       0       0       1

***VARIABLE    5 TYPED FORM               N=     4405

     CODE VALUE:         0       1       2       3       4       5       6       7
     FREQUENCY:        635     378     460       7       0       0       0    2925

***VARIABLE    6 URBAN/RURAL              N=     4405

     CODE VALUE:         1       2       3       4       5       6       7       8       9
     FREQUENCY:       2740    1400       0       0       0       0       0       0     265

***VARIABLE    7 TYPE OF ACCIDENT         N=     4405

     CODE VALUE:         1       2       3       4       5       6       7       8       9      10      11      12      13      14      15
     FREQUENCY:        313    1357     332    1357       0       0       0       0     213       0     145      45       9       8       0

     CODE VALUE:        16      17      18      19      20      21      22      23      24      25      26      27      28      29      30
     FREQUENCY:          0       0       0      22       0      13       6       0      13       0       0       0       0       0       0

     CODE VALUE:        31      32      33      34      35      36      37      38      39      40      41      42      43      44      45
     FREQUENCY:         10      29       4      17       0       0       0       0       3       0     253      75      15      11       0

     CODE VALUE:        46      47      48      49      50      51      52      53      54      55      56      57      58      59      60
     FREQUENCY:          0       0       0      37       0      46      35       0       9       0       0       0       0       0       0

     CODE VALUE:        61      62      63      64      65      66      67      68      69      70      71      72      73      74      75
     FREQUENCY:          0       0       0       0       0       0       0       0       5       0       0       1       0       0       0

     CODE VALUE:        76      77      78      79      80      81      82      83      84      85      86      87      88      89      90
     FREQUENCY:          0       0       0       3       0       0       2       0       1       0       0       0       0       0       0

     CODE VALUE:        91      92      93      94      95      96      97      98      99
     FREQUENCY:          0       0       0       0       0       0       1       2       7

***VARIABLE    8 VEHICLE IMPACT           N=     4405

     CODE VALUE:         1       2       3       4       5       6       7       8       9      10      11      12      13      14      15
     FREQUENCY:        323    1386     336    1374     226     399     121      24      19      52      59      43       0      23       0
```

FIGURE 6 Sample Means and Marginals Output

I) THE ANALYSIS OF VARIANCE PROGRAM

ANALYSIS VALUE A=5

This program produces one-way analysis of variance tables and the usual statistics associated with such tables. The control variable may only take the values 0,1,2, ...,11. Up to fifteen tables can be constructed in one pass.

The program offers the additional capability of selecting, for each table in a run, independent of all other tables, a subset of the data on the basis of up to two table filter variables. These table filters are in addition to the global filter which applies to the entire data set and thus affects all the tables.

TABLE FILTERS:

You may specify up to two filter variables with each table for use only with that table.

If no filters are specified for a table, the total data sample is used; that is, all cases that satisfy any filter conditions of the filter statement are used. If one filter variable is specified, a case is selected for the table provided that the value of the filter variable is within the specified selection range in the table parameters. If two filter variables are specified, the case is entered into the table provided that the value of the first filter variable is within the selection range for the first filter and the value of the second filter variable is within the selection range for the second filter.

NOTE: The case is not selected for the table unless it satisfies the conditions of both table filters. The case is not even considered if it does not satisfy the conditions of the global filter card.

RESTRICTIONS:

1. Only single-column control variables may be used.
2. Cases with values greater than 11 for the control variable will be omitted from the Table without notation.

0. $SOURCE HSRI:SPAD(D) D=Data File Value (Tape Mount)

1. $SOURCE HSRI:SPAD(D+5)

2. FILTER(RECODE OR LABEL)=

   Filter, Recode, or Label (See Section 3)

3. RECODE(OR LABEL)= (Prompted for only if the above was a filter)

   Recode or Label (See Section 3)

4. LABEL= (Prompted for only if the above was a recode)

   Label (See Section 3)

5. FORMAT FOR TABLES?
   NO no description printed out
   YES table line parameters are described

6. TABLE CARDS ON UNIT 4?
   NO must answer "no"

7. ....5....10....15....20....25...30...35....40...45...50

TB NO	CONT VAR	DEP VAR	WT VAR	FV1	FV1 MIN	FV1 MAX	FV2	FV2 MIN	FV2 MAX
Unique table identification no. (integer)	Control variable number (noV)	Dependent variable # (noV)	Weight variable # (noV) or 0 no blank	Variable # of first filter for this table. 0 or blank no filters for this table	Left end point for first filter	Right end point for first filter	Variable # of second filter for this table 0 or blank no second filter	Left endpoint of second filter	Right endpoint of second filter

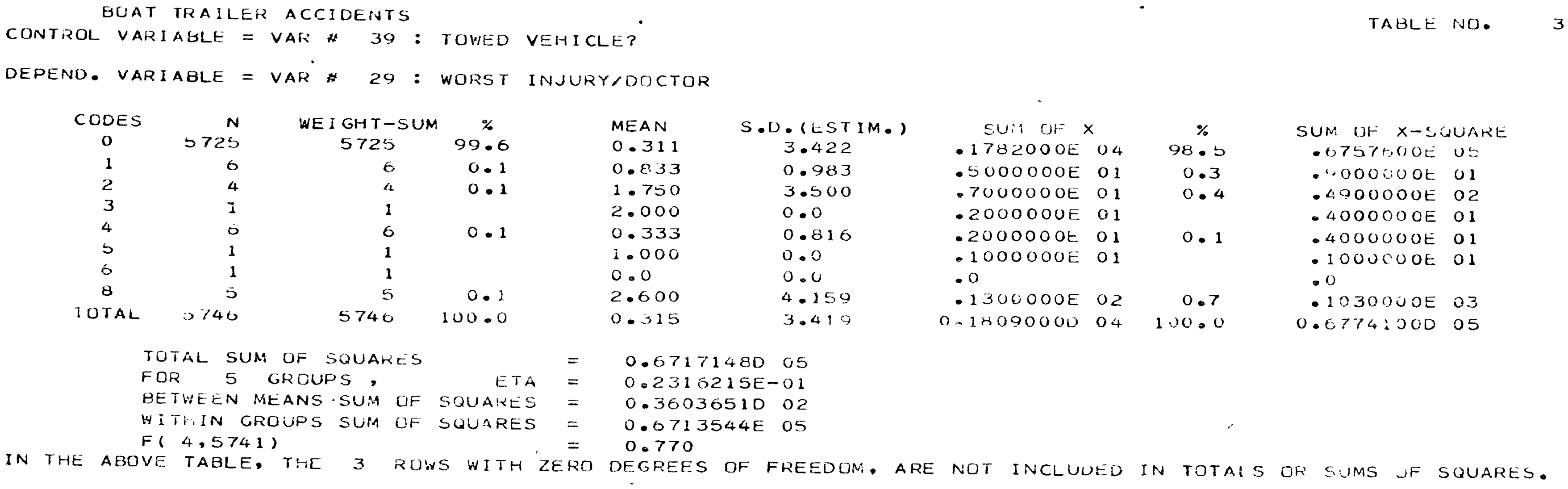
Terminal only:

Col 1 = 1 Eta Statistic printed at terminal

Col 2 = 1 F-Ratio Statistic printed at terminal

The user should supply one line of parameters for each table.

8. $ENDFILE



```
      BOAT TRAILER ACCIDENTS                                                                                   TABLE NO.    3
CONTROL VARIABLE = VAR #  39 : TOWED VEHICLE?

DEPEND. VARIABLE = VAR #  29 : WORST INJURY/DOCTOR

     CODES      N     WEIGHT-SUM    %        MEAN     S.D.(ESTIM.)     SUM OF X        %       SUM OF X-SQUARE
       0      5725       5725     99.6      0.311       3.422       .1782000E 04    98.5      .6757600E 05
       1         6          6      0.1      0.833       0.983       .5000000E 01     0.3      .9000000E 01
       2         4          4      0.1      1.750       3.500       .7000000E 01     0.4      .4900000E 02
       3         1          1               2.000       0.0         .2000000E 01              .4000000E 01
       4         6          6      0.1      0.333       0.816       .2000000E 01     0.1      .4000000E 01
       5         1          1               1.000       0.0         .1000000E 01              .1000000E 01
       6         1          1               0.0         0.0         .0                        .0
       8         5          5      0.1      2.600       4.159       .1300000E 02     0.7      .1030000E 03
     TOTAL    5746       5746    100.0      0.315       3.419      0.1809000D 04   100.0     0.6774100D 05

         TOTAL SUM OF SQUARES              =   0.6717148D 05
         FOR   5  GROUPS ,          ETA    =   0.2316215E-01
         BETWEEN MEANS SUM OF SQUARES      =   0.3603651D 02
         WITHIN GROUPS SUM OF SQUARES      =   0.6713544E 05
         F( 4,5741)                        =   0.770
IN THE ABOVE TABLE, THE  3  ROWS WITH ZERO DEGREES OF FREEDOM, ARE NOT INCLUDED IN TOTALS OR SUMS OF SQUARES.
```

FIGURE 7 Sample Analysis of Variance Output

J) THE BIVARIATE ANALYSIS PROGRAM

ANALYSIS VALUE A=6

The basic output of this program is a bivariate frequency table. Each frequency table is entirely independent of all other tables, and for any particular table, each output option is entirely independent of all other options.

A table is described to the program by naming two variables to be used as the independent (ROW) and dependent (column) variables, specifying whether the table shall contain weighted or unweighted frequencies, and specifying whether the entire sample or a subset of the sample shall be used for that table. The bivariate frequency table is limited to column variables with a code range of 0 to 11.

Options include:

Percentage options (each of these produces an additional page of output)

Row — A table in which cells and marginals are expressed as percentages of the row totals.

Column — A table in which cells and marginals are expressed as percentages of the column totals.

Corner — A table in which cells and marginals are expressed as percentages of the grand total.

Delete — An option to delete the frequencies of invalid codes before computing percentages based on the sum of code frequencies. The range of valid codes is defined by the minimum and maximum codes specified for the independent and dependent variables. Thus, if the codes for inappropriate responses and no answer are at the low or high end of the scale, these responses can, by the delete option, be eliminated from the percentages and from the total N upon which these are based.

Statistics

Any statistic options specified are printed out under the frequency distribution and are computed using the range of code values designated by the minimum and maximum values of the control and spread variables. Any or all of the following can be computed:

1) Chi-Square

2) Contingency Coefficient

3) Goodman-Kruskal's Gamma

4) Kendall's Tau (Alpha) and Tau (Beta) including intermediate terms:

   A) Numerator of Tau

   B) S.D. of the numerator of Tau

   C) BI

   D) BJ

5) Lambda Statistics

RESTRICTIONS:

Number of Variables

50 or less unique variable numbers can be called for in any one job; however, each number can be used as often as desired.

Number of Tables

The user may request as many tables as desired. However, 15 tables are done per pass over the data; each additional pass will increase execution time.

Range of Codes

A. Column Variable
   The range of codes must be from 0 to 11; any other codes will fall in a wild code category.

B. Row Variable
   The row variable has no range restriction.

C. Filter Variables (fourlocal filters can be specified for each table).

EXECUTION:

0. $SOURCE HSRI:SPAD(D) D=Data File Value (Tape Mount)
1. $SOURCE HSRI:SPAD(D+6)
2. FILTER(RECODE OR LABEL)=

   Filter, Recode, or Label (See Section 3)
3. RECODE(OR LABEL)= (Prompted for only if the above was a filter)

   Recode or Label (See Section 3)
4. LABEL= (Prompted for only if the above was a recode)

   Label (See Section 3)
5. ENTER TABLE ID:

   Supply up to 8 characters as the table headings. No numbers.
6. NEED TABLE DESCRIPTION?

   YES terminal table parameters are printed out
   NO terminal table parameters are not printed
7. TYPE IN TERMINAL TABLE PARAMETERS (not prompted)

   Type "DONE" to end the table parameters and begin execution.

One set of parameters is needed to describe each table. Thus, each frequency table is entirely independent of all other tables, and for any particular table, each option is entirely independent of all other options with the exception of the "DELETE" and the statistics options (OPT) which require specification of the valid minimum and maximum values of the row and column variable. For any one table only one row and one column variable are required; any options not used should not be entered.

(XX,YY,ZZ and X(#) represent values to be supplied by the user; the other symbols are keywords that must precede the value, including the "=" sign.)

PARAMETER DESCRIPTION

TN=XX

XX represents the table number (to be assigned by the user) of the current table being requested. Use this parameter to signal the start of a new table. Any parameters appearing before the first "TN=..." are ignored.

RV=XX

XX represents the row variable number; optional form for specifying a code range is:

RV=XX,YY to ZZ

YY to ZZ represents the range of codes (inclusive) to be used in calculation of statistics. Option X(1) must=1 and ZZ > YY.

CV=XX

XX represents the column variable number; specification of valid code range same as for "RV"

CV=XX,YY to ZZ

YY to ZZ represents the range of codes (inclusive) to be used in calculations. Option X(1) must=1, ZZ > YY.

WV=XX

XX represents the weight variable number (optional).

FV1=XX,YY to ZZ

XX represents the first filter variable number. YY to ZZ defines the range of codes comprising the subset of data to be included in the current table (inclusive).

NOTE: This filter applies only to this table; the global filter card before the label applies to all tables. If a case is excluded because of the global filter conditions, it won't even be considered by the table filter.

FV2=XX,YY to ZZ — XX represents the second filter variable; YY and ZZ are as above for the first filter but apply only to the second filter.

FV3=XX,YY to ZZ — Same as above.

FV4=XX,YY to ZZ — Same as above.

OPT=X(1)X(2)X(3)X(4)X(5)X(6)X(7)X(8)X(9)X(10)X(11)X(12)

where X(i) corresponds to the following options:

X(1) =DELETE(restricts range of codes for calculations, see RV,CV). Ranges for both RV and CV must be specified when this option is used.

X(2) =ROW PERCENT(cells and marginals percent based on row totals)

X(3) =COLUMN PERCENT(cells and marginals based on column totals)

X(4) =CORNER PERCENT(cells and marginals based on grand total)

X(5) =CHI SQUARE

X(6) =CONTINGENCY COEFFICIENT

X(7) ±TAU A(Kendall's)

X(8) =TAU B(Kendall's)

X(9) =GAMMA (Goodman-Kruskal's)

X(10) =LAMBDA 1

X(11) =LAMBDA 2

X(12) =LAMBDA 3

These options are normally off. To request any one, set the appropriate X(i) to "1".

EXAMPLE: OPT=0111 would compute the row, column and corner percentages without deleting any data. The other options would not be computed.
Since these option requests are positional, if any option is desired, the intervening positions must be filled with zeros.
If only option X(11) (Lambda 2) is desired, then supply:

```
OPT=000000000010
```

OPT=ALL1

If the user wishes all options to be set to 1, he may use "OPT=ALL1". This is equivalent to:

```
OPT=111111111111
```

SAME

This keyword causes all otherwise undefined parameters in the current table to be set to the values used in the previous table. If the parameters were not defined in the previous table, the result will be disastrous.

DONE

This keyword causes table entry to end and execution to begin. Place it only at the end of all tables you wish to enter.

```
HEAD-ON TWO CAR ACCIDENTS                                                                   TABLE #    9          PAGE #   17

                                                   BIVARIATE FREQUENCIES

                           SELECTING ONLY THOSE CASES CODED (    2-    3) ON VARIABLE NO.  .51:  TYPE OF VEHICLE #1
                                 AND ONLY THOSE CASES CODED (    0-    1) ON VARIABLE NO.  102:  TYPE VEH. #2

ROW (CONTROL) VARIABLE NO.  148                                                          COLUMN (SPREAD) VARIABLE NO.  149
  SPEED CAR #1, BRACKET-A                                                                 SPEED CAR #2, BRACKET-A

                  ( 0)    ( 1)    ( 2)    ( 3)    ( 4)    ( 5)    ( 6)    ( 7)    ( 8)    ( 9)    ( 10)   ( 11)   WILD      TOTAL
( 0)                0       0       0       1       0       0       0       0       0       0       0       0       0          1
( 1)                2      10       0       6       0       2       0       0       0       0       0       0       0         20
( 2)                8       6       5       1       0       2       0       0       0       0       0       0       1         23
( 3)                3       9       3       4       0       1       0       0       0       0       0       0       0         20
( 4)                2       2       6       3       3       0       0       0       0       0       0       0       1         17
( 5)                1       2       0       0       4       2       0       0       0       0       0       0       0          9
( 6)                1       0       0       0       0       1       1       0       0       0       0       0       0          3
( 7)                0       0       0       1       0       1       0       0       0       0       0       0       0          2
( 9)                0       0       0       0       0       0       1       0       0       0       0       0       0          1
(99)                0       1       0       1       0       0       0       0       0       0       0       0       0          2
 TOTAL             17      30      14      17       7       9       2       0       0       0       0       0       2         98
FOR DELETIONS, IMIN=    0  IMAX=   10  JMIN=    0  JMAX=   10     GRAND TOTAL=     94

AFTER DELETIONS, REVISED ROW TOTALS=
( 0- 9)             1      20      22      20      16       9       3       2       0       1
(90-99)             0       0       0       0       0       0       0       0       0       0                       0
REVISED COLUMN TOTALS=
                   17      29      14      16       7       9       2       0       0       0       0       0       0
```

FIGURE 8 Sample Bivariate Frequencies Output

WARNING: On statistics of weighted data.

In the analysis of weighted samples, the program assumes that for each observation a weighting factor has been coded into some field in the data. The user should be aware of some limitations in the use of summary statistics produced from weighting frequency tables.

The program generates only summary descriptive statistics. It does not go on to compute probability values for tests of significance (e.g., against the null hypothesis of no underlying bivariate association in the population from which the sample was drawn). Even though the program does generate the raw materials from which such estimates may often be readily calculated (such as the N or the X^2 quantity), it does not perform such a calculation. The reason for the limitation is that the form of a test of significance which is appropriate for a given bivariate distribution depends, among other things, upon the precise nature of the sample design which generated the data. Most textbook formulas in inductive statistics presuppose a simple random sample, or at least a sample in which the elements have an equal probability of selection. In many areas of research, these conditions are frequently not fulfilled. The consumer using the output of this program as raw materials for significance testing should modify his calculations in a manner suited to his sample design.

One example may clarify this limitation. Observations may be given different weightings in order to reproduce the original population structure, and these weightings should be taken into account (as this program takes them into account) in order to compute a descriptive statistic of degree of association such as the tau-beta. However, the N presumed for purposes of significance testing is not the weighted N given by this program, but rather the "effective" N, a quantity which lies somewhere between the weighted N and the raw (unweighted) number of observations. Where input has called for some weighting, and significance tests are required, the consumer must take the peculiarities of his weighting scheme into account on his own in order to arrive at the effective N appropriate for his significance-test calculations.

The program prints a reminder of these limitations whenever a user requests a summary statistics from a weighted frequency table.

K) THE USE OF SETUP FILES

Each of the programs describe above requires the correct entry of user responses from the terminal to a number of program prompts. These responses are supplied while the program is in execution on the computer. If all the required responses are put into a private file (called a SETUP file) before the program is executed, the following advantages are obtained:

1) The actual analysis program is not loaded in the computer memory (and thus accruing residency charges) while the slow job of manually typing in information is performed. That is, the operation is more cost-effective.

2) Many potential errors in the input information can be found by editing the setup file before it is used. This decreases the possibility of an error in execution and can save greatly on processing costs; especially for expensive jobs.

3) If an error does occur in the setup file, the input information is not lost when execution terminates. Only the error in the setup file need be corrected by entering the correct specification.

The technique involves the creation of a user file with any convenient name. For discussion purposes, this file will be called "SETUP." The user supplied input information, beginning with the filter statement, are entered in the setup file using a line in the file for each input line. The form and order of the information in the setup file must correspond exactly to the data format required by the program. (Remember that $$ENDFILE must be used to enter $ENDFILE in a line file.)

When the setup file has been created and edited, and program execution has been initiated via the appropriate SPAD command, the information stored in "SETUP" can be fed to the program in response to the filter request by the following sequence:

```
FILTER(RECODE OR LABEL)=
$CONTINUE WITH SETUP RETURN
```

This will cause MTS to look to the file "SETUP" as a source of input information. As an example, a setup file for the Data Set List program might appear as in Example 30.

```
#$LIST SETUP
       1      INCLUDE V25=01-14*
       2      ACCIDENT STUDY
       3      NO
       4      1
       5      1
       6      YES
       7      0
       8      0
       9      0
      10      V1, V5, V24-36, V52, V86*
      11      $ENDFILE
# END OF FILE
#
```

EXAMPLE 30

Using this technique, it is possible to create and edit a setup file before mounting a tape and executing an analysis program. This greatly increases the probability of correct inputs and speeds the actual data processing step.

L) CHECK OF OUTPUT RESULTS

Before printing out the entire analysis results that are stored in TBBL, it is advisable to check this file to determine that the program execution was successful and that the desired data was obtained. This is most conveniently done by using the $COPY TBBL(J,K) command to "browse" through portions of the file and look for expected results. In particular, it is useful to $COPY TBBL(LAST-5) to see if the program successfully completed its analysis task.

M) DISMOUNTING THE ACCIDENT FILE TAPE

When the user is satisfied that his data is reasonable and that no data files on the currently mounted tape will be used shortly, the tape should be dismounted. The Computing Center charges for tape mounts and for tape drive usage. At present rates, it is cheaper to dismount and remount a tape than to leave it idle for a period longer than three minutes. When tape drive usage is high, however, maintaining possession of a drive may be an overriding consideration.

Before dismounting a tape, remember to check that the next file to be accessed is not on the same tape. A summary of equivalent SPAD mount commands is shown in Section 5-P.

Dismounting a tape is accomplished by a SPAD command using N=1 (i.e., D=0, A=0, U=1). The action of this SPAD command is shown in Example 31.

```
#$SOURCE HSRI:SPAD(1)
#$RUN *DISMOUNT PAR=*T*
#EXECUTION BEGINS

 *T*: DISMOUNTED
#EXECUTION TERMINATED
#
```

EXAMPLE 31

N) PRINTING THE ANALYSIS RESULTS

The simplest way to print out the analysis results is to copy TBBL to the terminal using the $COPY, %LEN, and %RMAR commands (see Section 2).

Some analysis results may be fifty to one hundred pages in length so that copying the quantity of data to a teletype can be a time consuming task. In addition, since the SRS output line is 132 characters long, teletypes (in combination with the device control commands) result in an output that is difficult to read.

Several alternatives exist for printing out the results. If the user is part of a large facility that can utilize its own computer as a remote job entry terminal, then MTS can be instructed to print out at that computer center. Obviously, arrangements for this must be made through HSRI.

Several institutional printout facilities of this nature already exist. Data stored in "TBBL" may be printed out at HSRI on a line printer by using the SPAD command

```
$SOURCE HSRI:SPAD(2).
```

Although the hardware exists for such printout, there are no established procedures for billing the printout, handling, and mailing costs to the user. Interested parties should contact HSRI for a resolution of these factors before using this facility.

Users at the National Highway Traffic Safety Administration (NHTSA) can have their data output in file "TBBL" printed out in Washington, D.C. by issuing the SPAD command

```
$SOURCE HSRI:SPAD(3).
```

O) ERROR CONDITIONS

In the process of executing SPAD operations, a large number of possible errors can be generated as the result of program operation, incorrect user input specifications, or machine failures. Depending on how these error conditions come about, one of four major points of origin may generally be ascribed to the diagnostic message received by the user.

1) Computer Operating System

   Some condition occurring during program execution completely defies the basic operating structure of the computer. For instance, the program might ask the computer to store a number in an impossible location or to perform an arithmetic calculation with undefined outcome. These conditions usually result in a PROGRAM INTERRUPT that terminates execution of the program at the point where the error occurs.

2) MTS

   Incorrect file names, invalid commands, and many other user-input syntax errors will result in error messages from MTS. Errors in the MTS command structure are usually caught and the user is prompted for replacement information.

3) FORTRAN

   Since many of the SRS programs are written in FORTRAN, it is possible to obtain input/output errors from the Fortran Operating System. These errors are usually associated with user supplied input where the computer expects numbers to be entered at the terminal, but finds other characters instead.

4) SRS

   The SR system has a large number of error conditions that relate to the syntax used in filter, recode, and variable list statements; to the names and locations of the dictionary and data files used to store the accident data; to the field widths of the variables used in the analysis program; and in general, to program specifications. Diagnostic messages that point to the source of the trouble are usually returned to the user.

While it is nearly impossible to identify and document all potential error statements that the user might receive, a summary of the more common errors is presented in Table VII. This table also suggests user action in each case to alleviate or remove the error.

The task of assembling user encountered error commands is a continuing task. HSRI welcomes the contribution of user's error messages to the collection presented here, but cautions the user to follow two main courses of action when they have received a puzzling error comment.

First, the bulkfile "TBBL" into which program output is routed, will often contain error messages not echoed during program execution. The user is strongly urged to"$COPY TBBL" for further information related to errors encountered.

Second, and most important, HSRI recommends that the user contact us whenever an error command that is not entirely clear is received. For users' convenience a comment relay command $SOURCE HSRI:SPAD(5)--has been added. Users may route comments or copies of problem output to the HSRI printer via this command, or call HSRI Systems Analysis at (313) 764-0248.

Table VII

Common Error Conditions

ERROR COMMENT OR CONDITION	REASON FOR ERROR/RECOMMENDED ACTION
T DOES NOT EXIST	User has neglected to mount a tape.
PERMANENT READ ERROR ON *T* or SENSE DATA ERROR	There is a "dirty" spot on the data tape or a hardware problem with the tape drive. Try dismounting the tape and remounting it at a later time. If the error persists, notify HSRI.
ERROR IN EITHER THE CALL TO OPEN THE DICTIONARY FILE or AN I/O AND OR ROUTINE ERROR	Usually generated by an incorrect variable field width entered by user. The best action here is to re-run the program after having checked the field widths and the data codebook.
ENTER REPLACEMENT OR CANCEL	Usually generated because a file or program requested does not exist as specified, this comment prompts the user for only that portion of the command which is incorrect. (Such as a file name).
ILLEGAL OR MISSING PARAMETER	The user has probably violated a syntax rule, and should check the structure of the command or statement he is using.
"TBBL" EXPANSION UNSUCCESSFUL or "TBBL":SIZE EXCEEDED	The size of the bulkfile, into which SPAD program output is routed, is too small to hold all the output data so that some portion is lost. The user should copy "TBBL" to see how much of the output was lost, and then decide whether the program needs to be run again. Also run *CATALOG to see if your total file space has been exceeded. Destroy any unnecessary files. (Contact HSRI for details on obtaining larger files.)
FCVTH :INVALID CHARACTER IN NUMERIC FIELD or ERROR:ALPHA CHARACTER IN NUMERIC FIELD	User has probably placed an alphabetic character in a space where a numeric character belongs, due to improper construction of a statement.

TABLE VII CONTINUED

ERROR COMMENT OR CONDITION	REASON FOR ERROR/RECOMMENDED ACTION
VALUE MEANING 1 ALPHA TYPE VARIABLE ACCESSED FOR NUMERICAL ANALYSIS 2 VARIABLE NUMBER NOT IN THE DICTIONARY ACCESSED 3 SUBROUTINE ERROR 4 FILTER CARD ERROR PROBABLY FIELD WIDTH ON CARD DOESN'T MATCH THAT IN THE DICTIONARY ERROR IN FUNCTION LSTVAR RETURN. K=2.	This rather complete error comment documents four commonly encountered errors. In the example, the user has generated error #2, by specifying a variable not in the dictionary being read. In the case of errors #1 and #4, the users should follow the instructions for alpha variables and filter statements in this section. Error #3 should never be encountered but should be reported to HSRI if it is received.
FILTER ERROR: VAR. FIELD WIDTH or FILTER ERROR: VARIED FIELD WIDTH FOR VAR.#---	User input of an incorrect field width for a variable in the filter statement will produce this comment. User should consult data codebook and resubmit a correct filter statement after restarting the program.
FILTER ERROR: ON SYNTAX LAST VALID CHAR. FOUND = , CURRENT CHAR. FOUND = ,	User should rerun program and check filter statement, because a syntax rule has been violated (incorrectly placed character such as a dash, comma, asterisk, parenthesis, or number).
EXCEEDED VAR. LIMIT = 15	The present capability of the filter allows only fifteen variable expressions ("Vx=yy") per filter statement. The user has exceeded this limit, and needs to rerun the program, specifying fifteen or less.
RECODE SYNTAX RULE VIOLATED LAST VALID CHAR. FOUND = CURRENT CHAR. FOUND INVALID =	In entering his recode statement, the user has supplied an improper construction and should restructure the statement and rerun the program.
RECODE ERROR: VARIABLE FIELD WIDTHS VARY IN LENGTH TRY AGAIN ON CONTINUATION CARD (41-50)=04, (51-80)=05,......	Generated by the user's violation of a recode syntax rule on his second line of a recode statement, this error gives an example of the correct continuation card syntax, after which the user is given a chance to "try again" on the incorrect line.

Table VII Continued

ERROR COMMENT OR CONDITION	REASON FOR ERROR/RECOMMENDED ACTION
PROGRAM INTERRUPT. PSW=xxxxyyyy xxzzzzzz	yyyy interruption code zzzzzz location causing the interrupt The conditions that cause program interrupts are often complicated. A list may be found in Vol. 1 of the MTS manual. Users should notify HSRI when these errors occur if no apparent cause of the difficulty is evident.
INPUT (OR OUTPUT) ERROR TYPE N N=800 through 824	These are SRS data file input and output error messages. SPAD commands were established to avoid these errors. If they occur, the user probably forgot to mount a tape before conducting analysis on the data.

P) SPAD COMMAND SUMMARY

```
Command Format:   $SOURCE HSRI:SPAD(N)    N=D+A+U
Tape Mount    :   $SOURCE HSRI:SPAD(D)
Analysis -    :   $SOURCE HSRI:SPAD(D+A)
Utility       :   $SOURCE HSRI:SPAD(U)
```

DATA FILE	D
Washtenaw Accident	100
Washtenaw Driver	180
Washtenaw Vehicle	190
Oakland Accident 1968	200
Oakland Accident 1969	210
Oakland Accident 1970	220
Oakland Accident 1971	230
Oakland Driver	280
Oakland Vehicle	290
CPIR Rev 3 Vehicle	300
CPIR Rev 3 Occupant	310
CPIR Rev 3 Injury	320
CPIR Rev 2 Vehicle	330
CPIR Rev 2 Occupant	340
Cornell Level 1 1970	400
Cornell Level 2 Veh.	500
Cornell Level 2 Occ.	510
Cornell Level 2 Accident	520
Bexar Accident 1969	600
Bexar Vehicle 1969	610
Bexar Accident 1970	620
Bexar Vehicle 1970	630
Dade Accident 1969	700
Dade Accident 1970	710
Denver Accident 1969	800
Denver Accident 1970	810
Seattle Accident 1969	900
Seattle Accident 1970	910

ANALYSIS	A
Tape Mount or Utility	0
Data Set List	1
Histogram	2
Means and Marginals	4
Analysis of Variance	5
Bivariate Frequencies	6

UTILITY	U
Tape Mount or Analysis	0
Dismount Tape	1
Print at HSRI (Ann Arbor, Michigan)	2
Print at NHTSA (Washington, D.C.)	3
Message Relay	5

EQUIVALENT MOUNT COMMANDS

HSRI TAPE NO.	FILE	SPAD INDEX D
4	CPIR 2 VEHICLE	330
	CPIR 2 OCCUPANT	340
6	BEXAR COUNTY ACCIDENT 1969	600
	BEXAR COUNTY VEHICLE 1969	610
8	CAL LEVEL 2 VEHICLE 1970	500
	CAL LEVEL 2 OCCUPANT 1970	510
10	DADE COUNTY 1969	700
	DADE COUNTY 1970	710
11	BEXAR COUNTY ACCIDENT 1970	620
	BEXAR COUNTY VEHICLE 1970	630
25	CPIR 3 VEHICLE	300
	CPIR 3 OCCUPANT	310
	CPIR 3 INJURY	320

SECTION 6
EXAMPLES OF OPERATION

In order to put the entire accident data retrieval operation in perspective, the analysis of a simple problem will be carried through from problem inception to the printout of analysis results. It is hoped that this procedure will clarify many of the operations involved.

A. THE PROBLEM

With the current concern over occupant protection, we can consider the problem of determining injury severity to front seat occupants for all seated positions and for different configurations of seat belt usage. Three levels of usage can be investigated: 1) no restraints; 2) lap but no torso restraints; and 3) both lap and torso restraints. As an added restriction, it is desired to limit the investigation to passenger cars only.

To obtain the most detailed information on seat belt usage and injury severity, the Revision 3 Collision Performance and Injury Report (or CPIR3) file will be utilized; because occupant information is desired, the occupant file must be employed in the analysis.

An examination of the codebook for the CPIR files shows the following variable numbers for the information of interest:

Overall Injury Severity	V600
Seat Position (i.e., left, right, etc.)	V579
Seat Location (i.e., front, rear, etc.)	V578
Lap Belt Usage	V592
Torso Restraint Usage	V596
Body Style	V124

Although the information can be obtained in many ways, three bivariate tables will be processed corresponding to the three conditions of seat belt usage. In each table, the overall injury severity (as row variable) will be tabulated by seated position (as column variable).

B. TERMINAL OPERATION

The complete record of an actual run to retrieve the data discussed above is presented in Figure 9. This record is unretouched except in the following aspects:

1) The signon ID has been changed
2) User inputs have been underlined and annotation has been added
3) For editorial convenience the paper was advanced manually in the teletype or the copy was cut for paging purposes

Annotations in Figure 9 are by letter; the discussion below will follow the same order as these annotations. The symbols H^c, N^c, and ATTN refer to the character delete character (CTRL-H), the line delete character (CTRL-N), and the attention break, respectively. These are non-printing characters and hence do not show on the teletype record.

a) User MYID signs on normally obscuring has password with the duplex switch.

b) The attempt to create a file called "SETUP" results in the diagnostic message that a file by this name already exists.

c) Remembering that "SETUP" was used in a previous run and is no longer needed, MYID issues the command to empty this file and confirms the command with "OK". MTS responds with a done noting the action has been accomplished.

d) In order to enter data into "SETUP", MYID issues the number command. There is some confusion when this is not followed by the line number 1 for input, but assuming that the computer knows best, the data line is entered anyway. MTS, of course, does know best and the data line is rejected as an invalid command.

e) Using his keen perception, MYID remembers that he just signed on and that there is no active file to put the data in. This is accomplished with the $GET command. The number command is remembered by MTS, however, and as soon as "SETUP" is made the active file numbering begins automatically, and the bivariate setup information is entered in line number order.

f) Having entered the TN keyword before the NO, a CTRL-N is issued to delete the line. The request for line 4 is repeated.

g) The typing mistake of entering an "8" instead of "9" is corrected in text, with CTRL-H. A similar situation occurs in line 18.

h) To stop numbering, "UNNUMBER", is entered in line 21 and, much to the consternation of the user, numbering continues. The assumption here is often that the command that was entered is absolutely correct and if it didn't work, you can't believe anything. Consequently, "STOP" is entered but that doesn't work much better. Well, an attention break always works to stop whatever is going on. However, it doesn't always work and numbering continues merrily. To cut an interminable story short, "$UNNUMBER" does stop numbering.

i) The setup file is listed to check its contents for accuracy.

j) It is noticed that the asterisk at the end of the filter statement was forgotten and that lines 21 and 22 contain extraneous data. These lines are corrected by entering the appropriate data lines.

k) Since "SETUP" is the active file, it can be listed by the one letter command "L".

l) After checking to see that the contents are corrected, the active status of "SETUP" is modified with "$RELEASE".

m) The data storage file "TBBL" is created before making the actual analysis run.

n) With the setup and data files ready, the magnetic tape containing the CPIR3 occupant file is mounted with the SPAD source command.

o) A confirmation comment indicates that the tape was successfully mounted on a drive with the designation T903.

p) The source command for bivariate analysis is entered.

q) This is a temporary information statement.

r) After the filter request, MTS is instructed to look to the file "SETUP" for further input by using the $CONTINUE WITH command. This means that the questions that follow (Enter Table ID, etc.) need not be answered, since the answers are being read from "SETUP" by MTS.

s) After termination of analysis, a command to copy TBBL at line 32 (near the beginning of Table 1) is partially entered. However, to speed up this checking it is decided to set the right margin to 25 with the device command. Then the command is given to copy 4 lines of "TBBL".

t) Since the beginning of the table seems all right, the last 11 lines are printed.

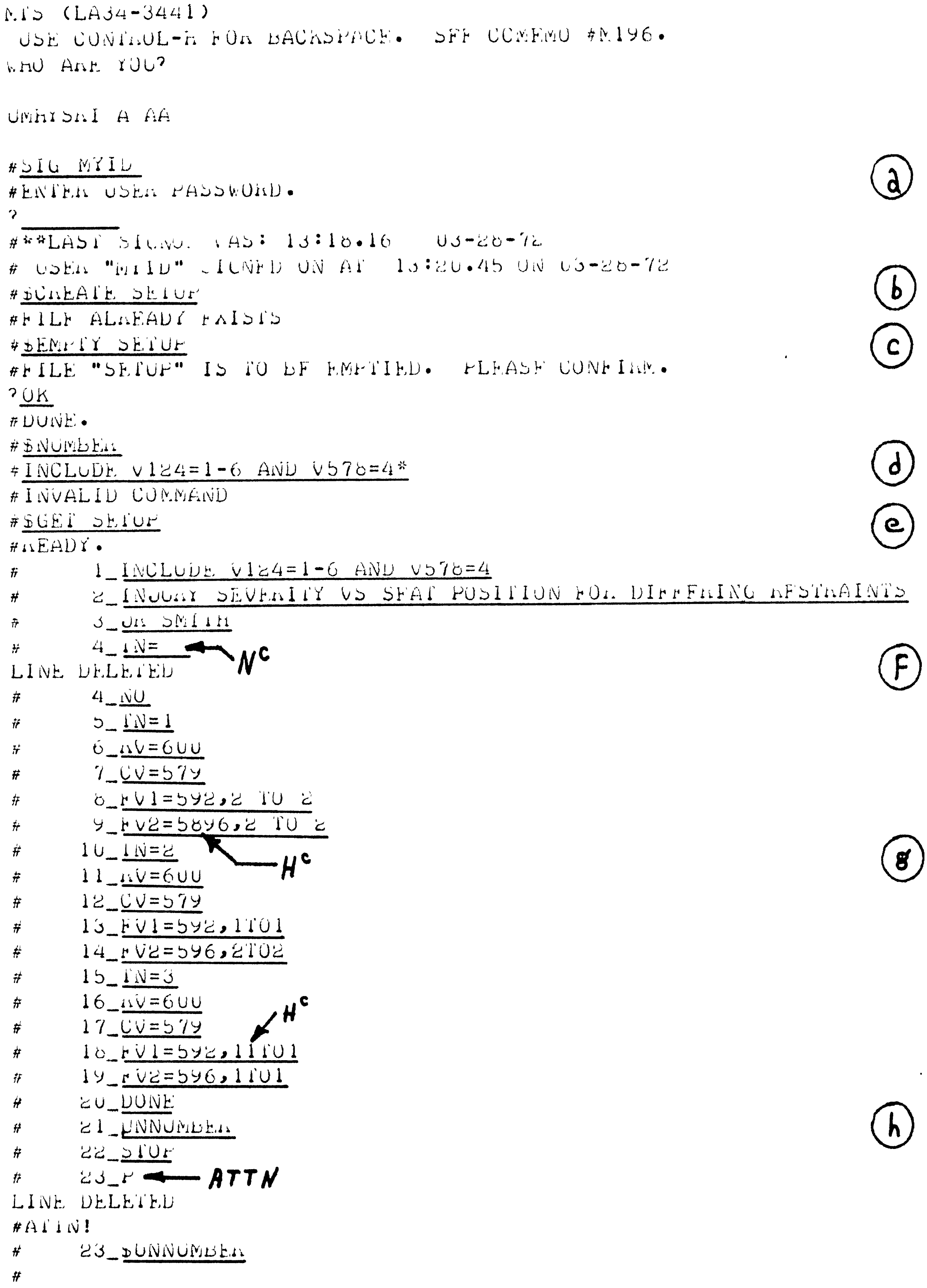
u) Again the output looks all right, so that the tape is dismounted with the SPAD source command. The dismount is confirmed by a comment.

v) In order to copy out an entire table, the margin is set to 68 and the maximum line length to 255. This is a convenient set of values for bivariate tables since it offsets the folded lines as shown in the printout. Note that the parts of lines that are folded back are indicated by an asterisk at the beginning of each line.

w) After using ATTN to stop the copy command, the file "TBBL" is printed out in batch with the SPAD source command. Remember that this facility should only be used <u>after</u> making arrangements with HSRI. The complete results of the analysis that result from this printout are shown in Figure 10.

x) The data in "TBBL" is not needed after the information is printed out so the file is destroyed.

y) The "SIG" command terminates the runs and shows that the entire operation lasted 29 minutes and cost approximately $5.28.



```
MTS (LA34-3441)
 USE CONTROL-H FOR BACKSPACE.  SEE CCMEMO #M196.
WHO ARE YOU?

UMHYSRI A AA

#SIG MYID
#ENTER USER PASSWORD.
?
#**LAST SIGNON WAS: 13:18.16    03-28-72
# USER "MYID" SIGNED ON AT  13:20.45 ON 03-28-72
#$CREATE SETUP
#FILE ALREADY EXISTS
#$EMPTY SETUP
#FILE "SETUP" IS TO BE EMPTIED.  PLEASE CONFIRM.
?OK
#DONE.
#$NUMBER
#INCLUDE V124=1-6 AND V578=4*
#INVALID COMMAND
#$GET SETUP
#READY.
#       1_INCLUDE V124=1-6 AND V578=4
#       2_INJURY SEVERITY VS SEAT POSITION FOR DIFFERING RESTRAINTS
#       3_OR SMITH
#       4_TN=
LINE DELETED
#       4_NO
#       5_TN=1
#       6_RV=600
#       7_CV=579
#       8_FV1=592,2 TO 2
#       9_FV2=5896,2 TO 2
#      10_TN=2
#      11_RV=600
#      12_CV=579
#      13_FV1=592,1TO1
#      14_FV2=596,2TO2
#      15_TN=3
#      16_RV=600
#      17_CV=579
#      18_FV1=592,11TO1
#      19_FV2=596,1TO1
#      20_DONE
#      21_UNNUMBER
#      22_STOP
#      23_P
LINE DELETED
#ATTN!
#      23_$UNNUMBER
#
```

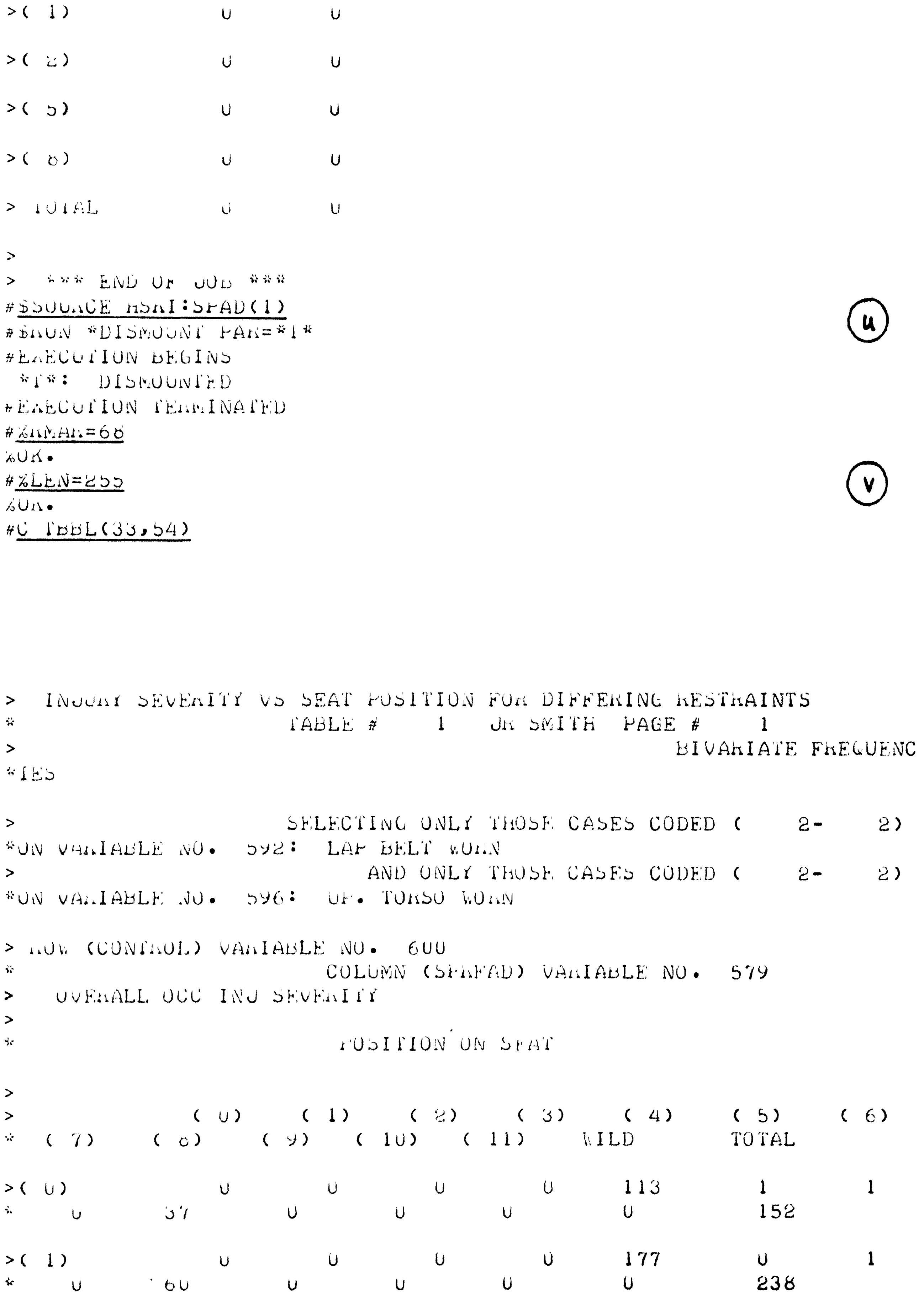
Figure 9
Sample Computer Run for Data Retrieval

```
 $LIST SETUP                                                        (i)
>        1       INCLUDE V124=1-6 AND V578=4
>        2       INJURY SEVERITY VS SEAT POSITION FOR DIFFERING RESTRAINTS
>        3       JR SMITH
>        4       NO
>        5       TN=1
>        6       RV=600
>        7       CV=579
>        8       FV1=592,2 TO 2
>        9       FV2=596,2 TO 2
>       10       TN=2
>       11       RV=600
>       12       CV=579
>       13       FV1=592,1TO1
>       14       FV2=596,2TO2
>       15       TN=3
>       16       RV=600
>       17       CV=579
>       18       FV1=592,1TO1
>       19       FV2=596,1TO1
>       20       DONE
>       21       UNNUMBER
>       22       STOP
#END OF FILE
#1,INCLUDE V124=1-6 AND V578=4*                                     (j)
#21,
#22,
#L                                                                  (k)
>        1       INCLUDE V124=1-6 AND V578=4*
>        2       INJURY SEVERITY VS SEAT POSITION FOR DIFFERING RESTRAINTS
>        3       JR SMITH
>        4       NO
>        5       TN=1
>        6       RV=600
>        7       CV=579
>        8       FV1=592,2 TO 2
>        9       FV2=596,2 TO 2
>       10       TN=2
>       11       RV=600
>       12       CV=579
>       13       FV1=592,1TO1
>       14       FV2=596,2TO2
>       15       TN=3
>       16       RV=600
>       17       CV=579
>       18       FV1=592,1TO1
>       19       FV2=596,1TO1
>       20       DONE
#END OF FILE
#$RELEASE                                                           (l)
#CR TBBL
# FILE "TBBL" HAS BEEN CREATED.                                     (m)
#
```

Figure 9 continued...

```
 $SOURCE HSRI:SPAD(310)                                                        (n)
#$RUN *MOUNT PAR=C0004 9TP *T* SIZE=3600 LP=OFF VOLUME=GMLF 'GMLF'
#EXECUTION BEGINS
 C0004 9TP *T* SIZE=3600 LP=OFF VOLUME=GMLF 'GMLF'

 *T*: MOUNTED ON T903                                                          (o)
#EXECUTION TERMINATED
#$COMMENT             PLEASE COPY SAV3:NEWS FOR RECENT CORRECTION MESSAGES.
#$SOURCE HSRI:SPAD(316)                                                        (p)
#$EMPTY -OATS
#DONE.
#$EMPTY -WHEY
#DONE.
#$RUN HSRI:PC650 3=-OATS 4=-WHEY 5=*SOURCE*+*MSOURCE* 6=TBBL
#EXECUTION BEGINS
  DDEF STATEMENT=

 UNIT  1 READING TAPE *T*                  , GMLF     , FILE IS OCC3DIC

 UNIT  2 READING TAPE *T*                  , GMLF     , FILE IS OCC3DAT

     **** OPTIONAL RECODING CAPABILITY NOW IN EFFECT.
    IF RECODE OPTION IS NOT DESIRED, CONTINUE EXECUTING                        (q)
    PROGRAM AS USUAL.
 FILTER (RECODE OR LABEL) =
 $CONTINUE WITH SETUP RETURN                                                   (r)
 RECODE (OR LABEL) =
 ENTER TABLE ID:
 NEED THE TABLE DESCRIPTION?
 TABLE      1: LINE   32.000
 TABLE      2: LINE   55.000
 TABLE      3: LINE   78.000
#EXECUTION TERMINATED
#C TBBL(32 ←── N^c
LINE DELETED
#ZRMAR=25                                                                      (s)
%OK.
#C TBBL(32,35)
>                          NOW W




 >   INJURY SEVERITY VS SEA
 >
#C TBBL(LAST-10)                                                               (t)

 >
 >                   ( 0)       ( 1

 >( 0)                 0         0
```

Figure 9 continued...

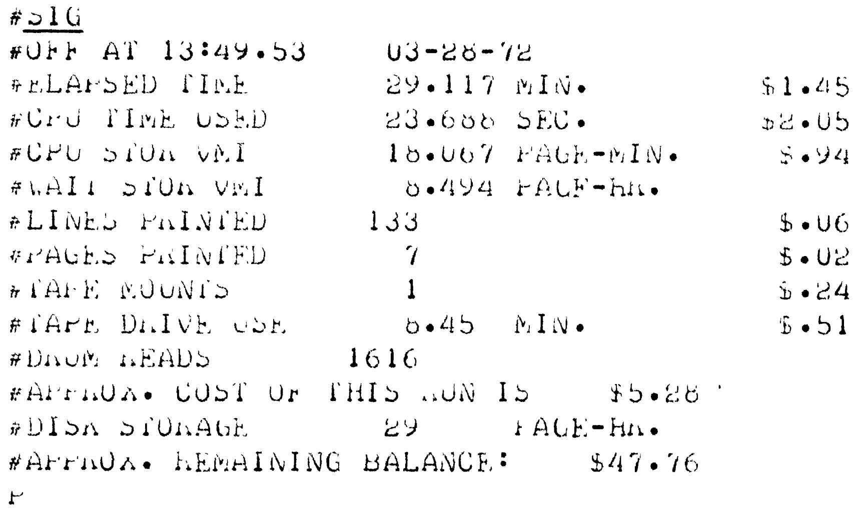


```
>( 1)                  0        0

>( 2)                  0        0

>( 5)                  0        0

>( 8)                  0        0

> TOTAL                0        0

>
>   *** END OF JOB ***
#$SOURCE HSRI:SPAD(1)
#$RUN *DISMOUNT PAR=*T*                                        u
#EXECUTION BEGINS
 *T*:   DISMOUNTED
#EXECUTION TERMINATED
#%RMAR=68
%OK.
#%LEN=255                                                      v
%OK.
#C TBBL(33,54)




>  INJURY SEVERITY VS SEAT POSITION FOR DIFFERING RESTRAINTS
*                         TABLE #     1    JR SMITH   PAGE #     1
>                                                    BIVARIATE FREQUENC
*IES

>                        SELECTING ONLY THOSE CASES CODED (     2-     2)
*ON VARIABLE NO.  592:  LAP BELT WORN
>                             AND ONLY THOSE CASES CODED (     2-     2)
*ON VARIABLE NO.  596:  UP. TORSO WORN

> ROW (CONTROL) VARIABLE NO.  600
*                         COLUMN (SPREAD) VARIABLE NO.  579
>   OVERALL OCC INJ SEVERITY
>
*                           POSITION ON SEAT

>
>                 ( 0)      ( 1)      ( 2)      ( 3)      ( 4)      ( 5)      ( 6)
*   ( 7)      ( 8)      ( 9)     ( 10)     ( 11)      WILD        TOTAL

>( 0)              0         0         0         0       113         1         1
*       0        37         0         0         0         0         152

>( 1)              0         0         0         0       177         0         1
*       0        60         0         0         0         0         238
```

Figure 9 continued...

```
>( 2)                 0        0        0        0       52        0        2
*    0      30        0        0        0        0            84

>( 3)                 0        0        0        0       31        0        1
*    0      19        0        0        0        0            51

>( 4)                 0        0        0        0       17        0        0
*    0       8        0        0        0        0            25

>( 5)                 0        0        0        0       11        0        0
*    0       5        0        0        0        0            16

>( 6)                 0        0        0        0       22        1        0
*    1       8        0        0        0        0            32

>( 7)                 0        0        0        0        5        0        0
*    0       1        0        0        0        0             6

>( 8)                 0        0        0        0       16        0        0
*    0       6        0        0        0        0            22

>( 9)                 0        0        0        0        3        0        0
*    0       7        0        0        0        0            10

>(98)                 0        0        0        0        1        0        0
*    0       1        0        0        0        0             2

>(99)                 0        0        0        0        6        0        0
*    0       4        0        0        0        0            10
#C TBBL(55,57)

> TOTAL               0        0        0        0      454        2        5
*    1     166        0        0        0        0           648




>  INJURY SEVERITY VS SEAT POSITION FOR \ ←ATTN
#ATTN!
#SOU HSRI:SPAD(2)                                                W
#$SET ECHO=OFF
#*PRINT* ASSIGNED RECEIPT NUMBER 620735
#*PRINT* 620735 RELEASED
#DES TBBL
#FILE "TBBL" IS TO BE DESTROYED.  PLEASE CONFIRM.                X
?OK
#DONE.
```

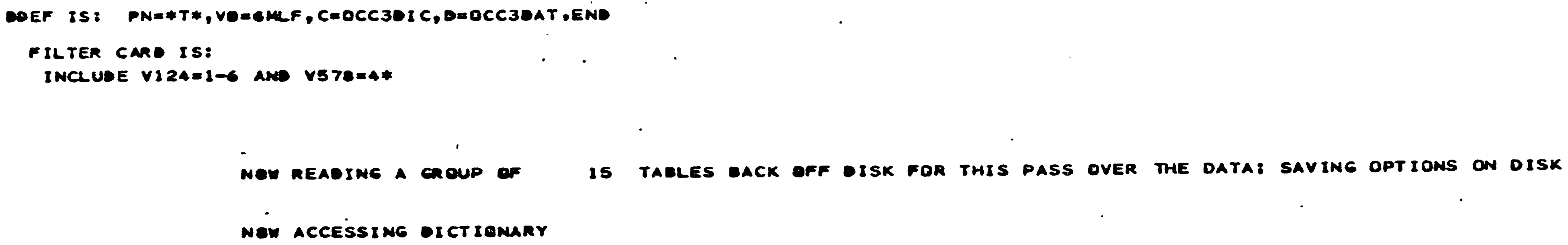
Figure 9 continued...



```
#SIG
#OFF AT 13:49.53      03-28-72
#ELAPSED TIME         29.117 MIN.            $1.45
#CPU TIME USED        23.688 SEC.            $2.05
#CPU STOR VMI         18.067 PAGE-MIN.         $.94
#WAIT STOR VMI         8.494 PAGE-HR.
#LINES PRINTED       133                       $.06
#PAGES PRINTED         7                       $.02
#TAPE MOUNTS           1                       $.24
#TAPE DRIVE USE        8.45  MIN.              $.51
#DRUM READS         1616
#APPROX. COST OF THIS RUN IS      $5.28
#DISK STORAGE         29       PAGE-HR.
#APPROX. REMAINING BALANCE:     $47.76
P
```

Y

Figure 9 continued...



```
DDEF IS:  PN=*T*,VO=6MLF,C=OCC3DIC,D=OCC3DAT,END
  FILTER CARD IS:
   INCLUDE V124=1-6 AND V578=4*


                    NOW READING A GROUP OF     15  TABLES BACK OFF DISK FOR THIS PASS OVER THE DATA: SAVING OPTIONS ON DISK

                    NOW ACCESSING DICTIONARY
```

Figure 10 Output from Sample Computer Run.

```
LISTING OF DICTIONARY IN SAME ORDER AS RECORDS
VAR.#       VARIABLE NAME       LOC. WID DEC RESP CTYPVTYP MDCODE1 MDCODE2
 579 POSITION ON SEAT            840  1   0    1    0    0  0000000
 592 LAP BELT WORN               862  1   0    1    0    0  0000000
 596 UP. TORSO WORN              866  1   0    1    0    0  0000000
 600 OVERALL OCC INJ SEVERITY    871  2   0    1    0    0  0000098

                 TOTAL # VARIABLES FOR JOB STEP=        4
                 TOTAL # RESPONSES FOR JOB=        4

                 NOW READING A CASE AND FILLING EACH TABLE FOR THIS PASS

                 TOTAL NUMBER OF CASES READ=    1200

                 NOW WRITING TABLES FOR THIS PASS
```

Figure 10 continued...

```
INJURY SEVERITY VS SEAT POSITION FOR DIFFERING RESTRAINTS                    TABLE #    1   JR SMITH  PAGE >    1
                                        BIVARIATE FREQUENCIES

                SELECTING ONLY THOSE CASES CODED (    2-    2) ON VARIABLE NO.  592:  LAP BELT WORN
                    AND ONLY THOSE CASES CODED (    2-    2) ON VARIABLE NO.  596:  UP. TORSO WORN

 ROW (CONTROL) VARIABLE NO.  600                                                COLUMN (SPREAD) VARIABLE NO:  579
   OVERALL OCC INJ SEVERITY                                                      POSITION ON SEAT

              ( 0)    ( 1)    ( 2)    ( 3)    ( 4)    ( 5)    ( 6)    ( 7)    ( 8)    ( 9)    ( 10)   ( 11)    WILD      TOTAL
 ( 0)           0       0       0       0     113       1       1       0      37       0       0       0       0         152
 ( 1)           0       0       0       0     177       0       1       0      60       0       0       0       0         238
 ( 2)           0       0       0       0      52       0       2       0      30       0       0       0       0          84
 ( 3)           0       0       0       0      31       0       1       0      19       0       0       0       0          51
 ( 4)           0       0       0       0      17       0       0       0       8       0       0       0       0          25
 ( 5)           0       0       0       0      11       0       0       0       5       0       0       0       0          16
 ( 6)           0       0       0       0      22       1       0       1       8       0       0       0       0          32
 ( 7)           0       0       0       0       5       0       0       0       1       0       0       0       0           6
 ( 8)           0       0       0       0      16       0       0       0       6       0       0       0       0          22
 ( 9)           0       0       0       0       3       0       0       0       7       0       0       0       0          10
 (98)           0       0       0       0       1       0       0       0       1       0       0       0       0           2
 (99)           0       0       0       0       6       0       0       0       4       0       0       0       0          10
  TOTAL         0       0       0       0     454       2       5       1     186       0       0       0       0         648
```

Figure 10 continued...

```
INJURY SEVERITY VS SEAT POSITION FOR DIFFERING RESTRAINTS                        TABLE #    2   JR SMITH   PAGE >    2
                                              BIVARIATE FREQUENCIES

                    SELECTING ONLY THOSE CASES CODED (    1-     1) ON VARIABLE NO.   592:  LAP BELT WORN
                         AND ONLY THOSE CASES CODED (    2-     2) ON VARIABLE NO.   596:  UP. TORSO WORN

ROW (CONTROL) VARIABLE NO.  600                                                   COLUMN (SPREAD) VARIABLE NO:  579
  OVERALL OCC INJ SEVERITY                                                          POSITION ON SEAT

               ( 0)    ( 1)    ( 2)    ( 3)    ( 4)    ( 5)    ( 6)    ( 7)    ( 8)    ( 9)    ( 10)   ( 11)    WILD       TOTAL
( 0)            0       0       0       0       45      0       0       0       10      0       0       0        0          55
( 1)            0       0       0       0       64      0       1       0       16      0       0       0        0          81
( 2)            0       0       0       0       12      0       0       0       3       0       0       0        0          15
( 3)            0       0       0       0       11      0       1       0       2       0       0       0        0          14
( 4)            0       0       0       0       2       0       0       0       1       0       0       0        0          3
( 5)            0       0       0       0       1       0       0       0       0       0       0       0        0          1
( 6)            0       0       0       0       4       0       0       0       1       0       0       0        0          5
( 7)            0       0       0       0       3       0       0       0       0       0       0       0        0          3
( 8)            0       0       0       0       2       0       0       0       0       0       0       0        0          2
( 9)            0       0       0       0       1       0       0       0       1       0       0       0        0          2
(19)            0       0       0       0       1       0       0       0       0       0       0       0        0          1
(99)            0       0       0       0       0       0       0       0       2       0       0       0        0          2
 TOTAL          0       0       0       0       146     0       2       0       36      0       0       0        0          184
```

Figure 10 continued...

```
  INJURY SEVERITY VS SEAT POSITION FOR DIFFERING RESTRAINTS                                TABLE #    3   JR SMITH  PAGE >    3
                                                     BIVARIATE FREQUENCIES

                    SELECTING ONLY THOSE CASES CODED (    1-     1) ON VARIABLE NO.  592:  LAP BELT WORN
                         AND ONLY THOSE CASES CODED (    1-     1) ON VARIABLE NO.  596:  UP. TORSO WORN

 ROW (CONTROL) VARIABLE NO.  600                                                           COLUMN (SPREAD) VARIABLE NO:  579
   OVERALL OCC INJ SEVERITY                                                                 POSITION ON SEAT

                 ( 0)     ( 1)     ( 2)     ( 3)     ( 4)     ( 5)     ( 6)     ( 7)     ( 8)     ( 9)     ( 10)    ( 11)     WILD       TOTAL
 ( 0)              0        0        0        0        5        0        0        0        1        0        0        0        0           6
 ( 1)              0        0        0        0       15        0        0        0        2        0        0        0        0          17
 ( 2)              0        0        0        0        2        0        0        0        0        0        0        0        0           2
 ( 5)              0        0        0        0        1        0        0        0        0        0        0        0        0           1
 ( 8)              0        0        0        0        0        0        0        0        1        0        0        0        0           1
  TOTAL            0        0        0        0       23        0        0        0        4        0        0        0        0          27

  *** END OF JOB ***
```

Figure 10 continued...

APPENDIX A

PHYSICAL CHARACTERISTICS OF COMMON TERMINAL DEVICES

A. TELETYPES

The Model 33 and 35 Teletypes have a transmission rate of 10 characters per second and have a carriage width of 72 positions. These models do not provide lower-case characters nor several of the special symbols used in programming. Some models have paper tape reading and punching equipment built into them.

The model 37 Teletype transmits at a rate of 15 characters per second and has a carriage width of 75 positions. Both upper and lower-case alphabetics and many special symbols can be used with the Model 37. Because the speed of the Model 37 is faster than other teletypes, this model can only be connected to MTS through the Data Concentrator (i.e., phone line 763-1500).

Most keys and controls are common to all Teletypes. The position and shape vary considerably, but the functions are identical.

Teletypes are capable of both half-duplex and full-duplex mode of operation. In half-duplex mode, the keyboard is connected to the printer so that whenever a key is pressed it is immediately printed, as well as sent to the computer. In full-duplex mode, the keyboard and printer are not connected. A character is printed only when sent from the computer. In this mode of operation, the computer will usually "echo" each keystroke so the user can see what he is typing. The control for HDX-FDX operations may be located in two places. On some Model 35 Teletypes (the ASR models), there is a switch to the left of the keyboard. Twisting it clockwise will put it into half-duplex mode. On the Model 33 and some Model 35 Teletypes (the ASR models), there is a toggle switch located above the telephone dial. Switching it to the left will put it into the half-duplex position.

Under the telephone dial, six buttons are found which are primarily concerned with the connection of the Teletype with the telephone lines and MTS. The ORIG, or "originate", button is located at the extreme left. This button must be pressed to connect the Teletype with the telephone line.

The CLR or 'clear' button is generally the second from the left. This button disconnects the telephone line and severs communication with MTS. This is pressed if, for example, the user gets a busy signal in response to his call, or if he gets no answer at all. Using the CLR button is not the recommended way to sign off.

The button labelled LCL or 'local' is used to operate the Teletype as a regular typewriter. This button enables the keyboard to work but makes no connection to the telephone lines and MTS.

The ordinary character keys on the keyboard act much like typewriter keys except that as their symbol is printed (if operating in half-duplex mode), the code for that symbol is sent to the computer. The SHIFT key selects the upper character on dual character keys. Note that some keys do not have an upper-shift function and may not be pressed in combination with the SHIFT.

A special key labelled CTRL for 'control' is similar to the SHIFT key in that it selects an alternate function for a key. The code sent to the computer by a key pressed in combination with the CTRL key usually does not represent a printable character and thus no symbol is printed. Some CTRL combinations are used for editing purposes by the device support routines (e.g., backspace and 'end-of-line'). These combinations have been designated in this guide by 'CONTROL-x' where x is the character key operated in conjunction with the CTRL key. Note that the labels on the upper part of the alphabetic keys generally refer to the control function, not to the shift function. Many Teletypes have color-coded lables: white lables for shift functions and red labels for control functions.

The LINE FEED key causes the paper to move up one line without changing the lateral position of the typing element. It also sends a LINE FEED character to the computer. This key is not used for any special purpose.

The RETURN key causes the typing element to return to the beginning of the line without spacing the paper. It sends a RETURN character to the computer. This signifies 'end of line'.

The RUBOUT key has meaning only when the Teletype is connected through the Data Concentrator. It signals a line-delete (and a "#" is echoed as a response).

The REPT key, when pressed in conjunction with any character key, causes that character to be repeated until the key is released. This is useful for spacing forward or for multiple 'backspaces'.

The BREAK key causes an attention signal to be sent to the computer to interrupt the current operation. This key is located to the left of the keyboard on a Model 35 Teletype, and at the right end of the Model 33. Any input or output in progress will be terminated. If the Model 35 is being used, the BRK RLS key, located at the left of the keyboard, must be operated after the BREAK key has been pressed if it lights up.

The ESC key is not found on all Teletypes. It is used only with the Data Concentrator to stop printing. If the Teletype is not equipped with the ESC key, the same function is provided by the combination of "CONTROL-SHIFT-K".

Two other keys are available on the Model 35. They are LOC LF and LOC CR. These allow the user to space the paper up and to perform a carriage return, respectively, without sending signals on to the communications line.

B. THE IBM 2741

The IBM 2741 is similar to the IBM Selectric typewriter. The table on which the IBM 2741 stands contains the control electronics for the typewriter and communications line. A mode switch on the left side of the table has two positions: LCL and COM, for 'local' and 'communicate'. In COM mode, the terminal is connected to the communications line, while in LCL mode, it may be used only as an ordinary typewriter. Power to the terminal is controlled by an ON-OFF rocker switch on the right side of the keyboard. This switch should be turned OFF when the terminal is not in use.

Most of the normal typewriter controls exist on an IBM 2741. There is a lever on the right rear of the typewriter cover which lessens the pressure on the paper and allows paper adjustment. On a pin-feed IBM 2741, this lever must be set so there is no pressure on the paper or it will not feed properly. A lever on the left adjusts the typing-head pressure to compensate for varying thicknesses of paper. The "golf ball" typing elements may be changed to provide a variety of character sets.

Left and right margin stops limit the travel of the typing element. The MAR REL key will temporarily release these stops. The margins may be used in conjunction with the margin editing facility of the device support routines to position the printing on the paper. A small pointer rides in a slot between the margin stops and indicates the current printing element position.

Physical tab stops may be set using the CLR-SET rocker switch on the left side of the keyboard. These may be used in conjunction with the input tabulation editing facility of the device support routines.

The IBM 2741 keyboard is very similar to that of an ordinary electric typewriter. When a character key is pressed, the symbol is printed on the paper and at the same time the internal code for that character is transmitted to the computer. The shift key allows upper and lower-case alphabetic characters to be produced and selects the upper or lower symbol on dual-character keys.

The TAB key causes the typing element to move to the next tab stop while sending the tab character code to the system. The backspace key moves the typing element back one position while sending a backspace character code to one system. Note that several keys, among them the space bar and the backspace key, may have a 'typamatic' mode. By pressing the key and holding it fully down, the user may have its function repeated until the key is released.

The RETURN key causes the typing element to return to the left margin and spaces the paper up one line. It also sends a return character to the system and locks the keyboard. The RETURN code informs the computer that the user has completed a line. The keyboard remains locked until the system again requests input. While locked, all keys except ATTN are inoperative.

The ATTN key is located on the upper right keyboard. When this key is pressed, an attention interrupt is signalled to MTS.

C. THE GE TERMI NET 300*

The GE TermiNet 300 is a high-speed (30 characters/sec.) impact-printing terminal with a carriage width of 118 characters that is utilized by some users

* This terminal is not optimally supported by MTS but is included here because it is employed by SPAD users at NHTSA in Washington, D.C.

of the SPAD system. Although not generally supported by MTS, it may be operated like a low-speed teletype at 10 characters per second by utilizing the teletype phone line (763-0300). Using a special signon procedure that is described below, this terminal may be operated at 30 characters/sec. by using the inherent flexibility of the Data Concentrator (763-1500).

The keyboard is similar to that of a standard typewriter with the addition of a number of extra control keys and switches. A detailed description may be found in the GE publication GEH-2184A entitled "Operations Manual for the TermiNet 300 Prints."

To use the TermiNet 300 at 30 cps, instead of the normal 10 cps, do the following:

1. Turn on terminal, set duplex switch to "HALF" on back of coupler, Push "ON LINE", set rate to "30", all else as normally used.

2. Dial (313) 763-1500 for the UM Data Concentrator, wait for tone.

3. Place phone in coupler, Press "INTERRUPT" until READY light stays on.

4. Hit "RETURN" key. If "LF" and "SPECIAL" are in the computer response you are on!

5. Enter "CTL-A CONTROL OFF=DUPLEX" to stop the double printing of your input.

6. To utilize the full 118 character page width, at 30 cps, enter the following two device commands:

"CTL-A CTL-A RM=118"

"CTL-A CTL-A LEN=255"

7. Proceed with terminal operations.

APPENDIX B
ONE-WAY ANALYSIS OF VARIANCE

I. INTRODUCTION

The method of one-way analysis of variance is used to test the equality of two or more population means (r,r>2), when the data is composed of r independent random samples. Thus, we might state our hypotheses as:

H_0(null hypothesis): $\mu_1=\mu_2=\cdots\cdot=\mu_r$

H_1(alternative hypothesis): $\mu_i\neq\mu_j$ at least two means are not equal where $\mu_j;j=1,2,\ldots.,r$ is the mean of the jth population, etc. for μ_i.

In many situations it may be of interest to compare a number of means. For example; a company testing 3 brands of tires may want to know if the average life of each brand is the same, by measuring tread depth after some specified hours of testing.

II. STATISTIC METHOD:

Under the assumptions: (a) The r columns represent r random samples from r populations. (b) Each of the r populations is normally distributed. (c) Each of the r populations has the same variance:

The observations can be arranged as follows:

	Conditions 1	 2	 3--------	 r	
Observations	x_{11} x_{21} ⋮ $x_{n_1 1}$	x_{12} x_{22} ⋮ $x_{n_2 2}$	x_{13} x_{23} ⋮ $x_{n_3 3}$	x_{1r} x_{2r} ⋮ $x_{n_n r}$	
Totals	$T._1$	$T._2$	$T._3$	$T._r$	$T..$
Means	$x._1$	$x._2$	$x._3$	$\bar{x}._r$	$\bar{x}..$
Variances	$S_1{}^2$	$S_2{}^2$	$S_3{}^2$	Sr^2	S^2

The first column represents a random sample of size n, from the population that has received condition 1; the second column represents a random sample of size n_2 from the population that has received condition 2, etc.

The total of the j$\underline{th}$ column is represented by $T._j$.

$$T.j = \sum_{i=1}^{n_j} X_{ij}$$

where

X_{ij} is the i^{th} observation under condition j

The mean of the j^{th} column is expressed by $\overline{X}._j$ where

$$\overline{X}._j = \frac{T._j}{n_j}$$

The variance of the j^{th} column S_j^2 is defined as:

$$S_j^2 = \frac{\sum_{i=1}^{nj} (X_{ij}-\overline{X}._j)^2}{n_j-1}$$

The total of all observations T.. is evaluated by:

$$T.. = \sum_{j=1}^{r} T._j$$

The mean of all observations $\overline{X}$.. can be computed by:

$$\overline{X}.. = \frac{T..}{N}$$

where

N the total number of observations $= \sum_{j=1}^{r} n_j$

The variance of all N observations regarded as a single sample is defined as S^2 where

$$S^2 = \frac{\sum_{j=1}^{r} \sum_{i=1}^{n_j} (X_{ij} - \overline{X}..)^2}{N-1}$$

One-way Analysis of Variance Table

Soruce of Variation	SS	d.f.	MS	EMS	F
Among Groups	SS_A	r-1	$SS_{A/(r-1)=MS_A}$	$\sigma^2+\sum^{r}\frac{n_j(\mu_j-\mu)^2}{r-1}$	MS_{A/MS_W}
Within Groups	SS_W	N-r	$SS_{W/(N-r)=MS_W}$	σ^2	
Total	SS_T	N-1			

The total sum of squares is represented by SS_T.

$$SS_T = \sum_{j=1}^{r} \sum_{i=1}^{n_j} X_{ij}^2 - \frac{T..^2}{N}$$

The within sample or within groups sum of squares is represented by SS_W.

$$SS_W = \sum_{j=1}^{r} \sum_{i=1}^{n_j} X_{ij}^2 - \sum_{j=1}^{r} \frac{T_j^2}{n_j} = SS_T - SS_A$$

The among samples or among groups sum of squares is referenced by SS_A.

$$SS_A = \sum_{j=1}^{r} \frac{T._j^2}{n_j} - \frac{T..^2}{N}$$

The expected mean square (abbreviated EMS) is defined as:

$$= \alpha^2 + \sum_{j=1}^{r} \frac{n_j(\mu_j-\mu)^2}{r-1}$$

The "ETA" statistic is known as the correlation ratio. The correlation between X and Y can be defined as:

$$\rho_{xy} = \frac{\text{Cov }(X,Y)}{\sqrt{(\text{VarX})(\text{VarY})}} = \frac{E[XY]-\mu_x\mu_y}{\sigma_x\sigma_y}$$

ETA, the "correlation ratio" is a measure of the total variance of the dependent variable accounted for (both linear and non-linear variance) by the difference between groups in the independent variable. Eta gives a rough indication of the correlation between the independent and dependent variables.

A necessary assumption for valid analysis of variance is the homogeniety or equality of the group variances. If one cannot meet this assumption any significant or non-significant difference between groups is suspect because of the possible invalidity of the analysis of variance. In order to test this assumption, one can perform the F max test by simply dividing the variance of the largest group by the variance of the smallest group (Fmax = $\frac{\text{Vlarge}}{\text{Vsmall}}$) and examining a table of Fmax distributed with d_{flarge}=N-1 (where N=the # of observations in the large group) and d_{fsmall}= L-1 (where L is the # of observations in the small group). Tables of FMAX are available in most statistical texts (C.F. Winer). In general the ANOVA is known to be robust with regard to violations of homosoelasticity (equal variances, normally distributed) and thus can tolerate mild disturbances of these parameters.

III. STATISTICS - REFERENCES:

1. The one way analysis of variance
William C. Guenther: ANALYSIS OF VARIANCE,
1964 by Prentice - Hall, Inc., Englewood Cliffs, N.J.,
pp. 31-43

2. CORRELATION COEFFICIENT
Harold J. Larson: INTRODUCTION TO PROBABILITY THEORY
and STATISTICAL INFERENCE, 1969 by John Wiley & Sons, Inc.,
pp. 156, 163

3. Analysis of Variance (General)
B.J. Winer: Statistical Principles In Experimental
Design 2nd Ed. 1971 McGraw-Hill.

APPENDIX C
BIVARIATE FREQUENCIES

I. Introduction

This program produces bivariate frequency distributions in tabular form with associated non-parametric statistics. Each such table can be expressed as a set of ordered pairs $\{(X_1,Y_1),\ldots,(X_1,Y_n),\ldots,(X_n,Y_m)\}$ where the X_i's and Y_j's represent the code values of dependent variables used for the distribution. Each of the cell (X_i,Y_j)'s then contain a count or frequency of the number of occurrences of the code value pair (X_i,Y_j) in the sample as shown in Fig. C-1. The following set of non-parametric statistics are also available with each table:

a) Chi-Square

b) Standard Deviation of S

c) Contingency Coefficient

d) Kendall Rank Correlation Coefficient

e) Gamma Statistic

f) Lamda Statistics

II. Statistical Methods

The following sections define the methods used to compute the various statistics generated by this program. The method used to generate the actual bivariate frequency tables is a simple count.

A. Chi-Square: χ^2

When frequencies in discrete categories constitute the data, a χ^2 test may be used to determine the significance of the differences among the k independent categories.

	X_1	X_2	...	X_i	...	X_n
Y_1	f_{11}					
Y_2						
⋮						
Y_j				f_{ji}		
⋮						
Y_m						f_{mn}

FIGURE C-1. FORMAT FOR BIVARIATE FREQUENCY TABLES.

$$\chi^2 = \sum_{i-1}^{r} \sum_{j=1}^{c} \frac{(f_{ij}-e_{ij})^2}{e_{ij}}$$

where f_{ij} = the actual frequency of the category (i,j)

e_{ij} = the expected frequency of the category (i,j)

$$N = \sum_{i=1}^{r} \sum_{j=1}^{c} f_{ij}$$

$$e_{ij} = \frac{\left(\sum_{i=1}^{r} f_{ij}\right)\left(\sum_{j=1}^{c} f_{ij}\right)}{N \; i=1 \; j=1}$$

r = number of rows

c = number of columns

B. Standard Deviation of S: σ_s

σ_s is a large sample test of significance for tau or gamma defined below. S equals the number of agreements in order minus the number of disagreements in order. For a given cell or catagory in a table, all cases to the right and below are in agreement, all cases to the left and below are in disagreement. S is computed via

$$S = \sum_{i=1}^{r-1} \sum_{j=1}^{c} f_{ij}\left(\sum_{k=i+1}^{r} \sum_{\ell=j+1}^{c} f_{k\ell} - \sum_{m=i+1}^{r} \sum_{n=1}^{j-1} f_{mn}\right)$$

$$\sigma_s^2 = \frac{N(N-1)(2n+5) - \sum_{j=1}^{c} N_j(n_j -1)(2N_j+5) -}{18}$$

$$\underline{\sum_{i=1}^{r} N_i(N_i-1)(2N_j+5)} \quad +$$

$$\frac{\left[\sum_{j=1}^{c} N_i(N_j-1)(N_j-2)\right]\left[\sum_{i=1}^{r} N_i(N_i-1)(N_i-2)\right]}{9(N)(N-1)(N-2)} +$$

$$\frac{\left[\sum_{j=1}^{c} N_j(N_j-1)\right]\left[\sum_{i=1}^{r} N_i(N_i-1)\right]}{2(N)(N-1)}$$

and

$$\sigma_s = \sqrt{\sigma_s^2}$$

where

N_j = the marginal total of column j

N_i = the marginal total of row i

C. Contingency Coefficient: C

The contingency coefficient is a measure of the extent of the relationship. The contingency coefficient has a value of zero when there is no relationship between the variables. When there is a relationship between the variables, C takes on positive values between zero and one. C is calculated from the total sample size N and χ^2 by the following formula.

$$C = \sqrt{\frac{\chi^2}{\chi^2 + N}}$$

D. Kendall Rank Correlation Coefficient: tau

Kendall's tau measures strength of relationship ranging from minus one in the case of perfect negative relationship to zero in the case of no relationship (statistical independence) to plus one in the case of perfect positive relationship. Tau-a assumes that as two items fall - or should fall - in the same row or column of the bivariate table. In contrast to tau-a, tau-b does allow several items to fall in the same row or column of the table. The two Kendall tau's are computed by:

$$\text{tau-a} = \frac{S}{\frac{N(N-1)}{2}}$$

$$\text{tau-b} = \frac{S}{B_i \cdot B_j}$$

where

$$B_i = \sqrt{\frac{N(N-1)}{2} - T_2}$$

$$B_j = \sqrt{\frac{N(N-1)}{2} - T.}$$

$$T_1 = \frac{\sum_{j=1}^{c} N_j(N_j-1)}{2}$$

$$T_2 = \frac{\sum_{i=1}^{r} N_i(N_i-1)}{2}$$

and S, N_i and N_j are defined above

E. E. Gamma Statistic: Gamma

Goodman and Kruskal's gamma also measures strength of relationship, ranging from minus one in the case of perfect negative relationship to zero in the case of no relationship to plus one in the case of perfect positive relationship.

$$\text{Gamma} = \frac{S}{\sum_{i=1}^{r-1} \sum_{j=1}^{c} f_{ij} \left(\sum_{k-i+1}^{r} \sum_{\ell=j+1}^{c} f_{k\ell} + \sum_{m=i+1}^{r} \sum_{n=1}^{j-1} f_{mn} \right)}$$

where

r = number of rows

c = number of columns

f_{ij} = the observed frequency in cell ij

LAMBDA Statistics: Lambda

The Lambda statistics are measures of the predictive association between categories. If one were trying to predict which category an observation would fall into based upon the dependent variable, one would be right most often by predicting the dependent variable category with the greatest number of cases. If one also knew the observation's value for the independent variable, a better prediction scheme would be to predict the dependent variable category with the greatest frequency for that independent category. Lambda measures the gain in predicting the independent variable values. Lambda has the value of zero when there's no predictive association between the variables and has the value one when there's complete predictive association. Lambda-A or Lambda (control) takes the row variable as the dependent variable.

$$\text{Lambda A} = \frac{\sum_{j=1}^{c} \max_{k} f_{jk} - \max_{k} f_{.k}}{N - \max_{k} f.k}$$

Lambda-B or Lambda (spread) takes the column variable as the dependent variable

$$\text{Lambda-B} = \frac{\sum_{k=1}^{r} \max_{j} f_{jk} - \max f_{j.}}{N - \max_{j} f_{j.}}$$

Lambda or Lambda (Control and spread) assumes no distination between rows and columns.

$$\text{Lambda} = \frac{\sum_{j=1}^{c} \max_{k} f_{jk} + \sum_{k=1}^{r} \max_{j} f_{jk} - \max_{k} f_{.k} - \max_{j} f_{j.}}{2N - \max_{k} f_{.k} - \max_{j} f_{j.}}$$

where

f_{jk} = frequency observed in cell $(X_j Y_k)$

$\max_k f_{jk}$ = largest frequency in column X_j

$\max_k f_{.k}$ = largest marginal frequency among the rows Y_k

$\max_j f_{jk}$ = bigest frequency in row Y_k

$\max f_{j.}$ = largest marginal frequency among the columns X_j

REFERENCES

1. Statistics, William L. Hays, Hold, Rinehart & Winston, 1963.
2. Non-Parametric Statistics, Sidney Siegal, McGraw-Hill, 1956.

APPENDIX D

SELECTED BIBLIOGRAPHY

A) Computer Manuals

Carnahan, B. and Wilkes, J., Introduction to Digital Computing and Fortran IV with MTS Applications. The University of Michigan, Ann Arbor, Michigan, 1968.

MTS, The Michigan Terminal System, Third Edition, The University of Michigan Computing Center, Ann Arbor, Michigan. Volume 1, MTS and the Computing Center, February 1971.

MTS, The Michigan Terminal System, Third Edition, The University of Michigan Computing Center, Ann Arbor, Michigan. Volume 2, Public File Descriptions, April 1971, Revised.

MTS, The Michigan Terminal System, Third Edition, The University of Michigan Computing Center, Ann Arbor, Michigan. Volume 3, Terminals and Tapes, September 1972.

Wood, D.E., and Hafner, C.D., The Statistical Research System. Highway Safety Research Institute, The University of Michigan, Ann Arbor, Michigan, 1972.

B) Statistics

Hays, William L. Statistics, Holt, Rinehart & Winston, 1963.

Siegal, Sidney. Non-Parametric Statistics. McGraw-Hill, 1956.

Winer, B.J. Statistical Principles in Experimental Design. Second Edition, McGraw-Hill, 1971.