Using SPSS on UNIX

Last revised: 3/27/00

SPSS is a comprehensive, integrated system for statistical data analysis that provides a wide variety of applications including data access, data management, data analysis, and data presentation.

SPSS provides several user interfaces ranging from interactive processing through SPSS Manager to batch processing. The version of SPSS is version 6.1 and it only runs on the SSCC UNIX computer called NORMAN.

Invoking SPSS

You have three ways of invoking SPSS: interactive SPSS Manager mode, interactive prompt mode, and noninteractive mode. The following examples show you how to invoke SPSS in each of these modes.

Interactive Mode

In interactive mode, SPSS lets you create, modify, and save command files and listing files; submit commands to the SPSS Processor; and view SPSS output. SPSS can be invoked in interactive mode from X displays simply by typing `spss` at the UNIX prompt. If you want to run SPSS in interactive mode from a non-X display, type:

```
> spss +m
```


and SPSS will appear with its character based manager. The interfaces of the SPSS character based manager and X are very different.

Note that it may be necessary to define an environment variable SPSSTERM prior to invoking SPSS in SPSS Manager mode. An example of defining the environment variable SPSSTERM for a terminal in Soc. Sci. 2470 is:

```
> setenv SPSSTERM xterm
```

Follow these steps to terminate SPSS from the character based manager:

1. Press escape, then `0`.

2. Press `e` followed by a Return. The following message appears on your screen:

   ```
   Unsaved changes in output file - OK to exit?
   ```

3. Press `y` followed by a Return and SPSS returns control to the operating system.

To terminate the X version of SPSS, select Exit from the File menu of either the Data Editor or Output windows.

Warning: The Motif version of SPSS does not work with monochrome NCD X-terminals. SPSS expects to fix this bug in the next release.

Interactive Prompt Mode

Typing this command:

```
> spss -m
```

at the UNIX prompt brings up SPSS in interactive prompt mode. An `SPSS>` prompt appears on your screen and you can begin entering SPSS commands. After each command another prompt appears. Enter `FINISH` to terminate your SPSS session and return control to the operating system.

Noninteractive Mode

Noninteractive or "batch" mode is by far the most common way of running SPSS at the SSCC. To invoke SPSS in noninteractive mode, you use the SPSS command for interactive prompt mode in combination with UNIX conventions for redirecting standard input and standard output. For example, suppose you have stored your SPSS commands in a file named `foo.sps` and you want your output written to a file called `foo.spsout`. To execute your program, type the following at the UNIX prompt:

```
spss -m <foo.sps >foo.spsout
```

You do not get another UNIX prompt until SPSS finishes executing the program. If you run this program in the background, however, you do not have to wait until your SPSS program finishes execution before you get the UNIX prompt. Your shell is available for other work. To run a noninteractive program in the background, simply add an `&` at the end of the SPSS command. For the example above, you would type the following:

```
spss -m <foo.sps >foo.spsout &
```

Submit only one SPSS background job at a time. This is because unlike batch mode on VMS, UNIX runs all submitted jobs simultaneously.

Increasing Memory

SPSS allocates 512K of memory to its data areas by default. You can request additional memory by adding the `-s` option to the `spss` command line. For example, to request two megabytes of memory for the example above, you would type the following:

```
spss -m -s2m <foo.sps >foo.spsout &
```

SPSS Programs

SPSS programs consist of one or more commands which are processed sequentially. Below is an example of a simple SPSS program that reads in data and then computes a crosstabulation table.

```
data list file=heart.dat / smoke 1-2 coronary 4-6 count 20-25
list
weight by count
crosstabs tables=smoke by coronary / statistics=all
```

SPSS Command Syntax

An SPSS command is an instruction that tells SPSS to perform some task or gives it information. Commands begin with a keyword (which may contain more than one word) that is the name of the command and often have additional specifications, such as subcommands and user specifications. In the example below

```
DATA LIST   FILE=MYDATA.DAT
```

`DATA LIST` is the keyword and `FILE=MYDATA.DAT` is the specification.

The following rules apply to all SPSS commands:

- All commands in the SPSS command file must begin in column one. If you use multiple lines to complete a command statement, column one of each continuation line must be blank.
- A command line cannot exceed 80 columns.
- Commands can be entered in upper and lower case.
- Spaces can be added between specifications at any point where a single blank is allowed. In addition, lines can be broken at any point where a single blank is allowed.

Both of the following are acceptable ways to write the same SPSS statement:

```
DATA LIST
   FILE=mydata.dat

Data List      File=mydata.dat
```

If you would like SPSS to check your syntax without actually executing any commands or reading any data, include

```
EDIT
```

as the first SPSS command in your program.

Data Access

SPSS can access data from a variety of sources including:

- raw data ready for data entry,
- data stored in files formatted by other software products, such as database

management software,

- data stored in other file formats including hierarchical files, variable-length fields and records, and multiple records.

Once SPSS accesses and reads the data, it organizes them into an SPSS data file. The data values are arranged in a rectangular table-like structure. The columns of the table are called variables and the rows are called cases. SPSS data files also contain descriptor information. During processing, SPSS reads the descriptor information to determine the attributes of the data set and the number and characteristics of each variable.

SPSS provides data definition commands as its means of getting your data into SPSS data files. The most basic component of SPSS data definition is the `DATA LIST` command. It tells SPSS where the data can be found, indicates how many records there are for each case, defines the variable names, and specifies their column locations. Following is an example of a `DATA LIST` command.

```
DATA LIST FILE=MYDATA.DAT /ID 1-4 VAR1 6-8 VAR2 12-15
```

The `FILE` subcommand specifies the name of the file containing the data. This is followed by a list of variable names and their column locations.

Use the `FILE` subcommand to specify the file containing the data described in the `DATA LIST` command. If the file is in the current directory, you can simply type the filename and filetype on the subcommand as shown in the example above. If the file is not in the current directory, the entire file specification must be enclosed in apostrophes or quotation marks, as shown in the example below:

```
DATA LIST FILE='mydata.dat'
 /ID 1-4 VAR1 6 VAR2 12-15 CHARVAR 17-25(A) VAR3 18-30
```

If you want to include the raw data in the same file as your SPSS commands, you can omit the `FILE` subcommand. This method requires two additional commands `BEGIN DATA` and `END DATA` which are discussed later in this handout.

The information following the `FILE` subcommand in the example above is referred to as the variable definition portion of the `DATA LIST` command. It is used to assign names to variables and provide information about the location and format of each variable. The variable definition portion begins with a "`/`" and is followed by a variable name (`ID`) and then the column location for the variable, and so on. If the variable is two or more columns wide, specify the number of the first column, followed by a dash (`-`), and then the number of the last column. The (`A`), in the example above indicates the variable is a string variable and is placed after the variable name and column location.

Variable names followed by column locations, as in the example above, are referred to as fixed-format data. Another method of entering raw data is by listing variable names without column locations. This method is referred to as free-field format. You can use free-field format when the variables are recorded in the same order for each case, but

not necessarily in the same locations. Values may be separated by blanks or commas. A blank field for a value will cause all values from that point on to be assigned to the wrong variable. So, make sure missing data are keyed as "." or some other special numeric value when using free-field format. When using free-field format in SPSS you must specify the keyword FREE on the DATA LIST command as in the example below:

```
DATA LIST FILE=mydata.dat FREE / ID VAR1 TO VAR4
```

VAR1 TO VAR4 is a shortcut way of specifying consecutive variables. TO is the keyword and defines the variables VAR1, VAR2, VAR3, and VAR4.
Sometimes, instead of keeping your data in an external file, you may prefer to enter your data along with your SPSS commands. In such cases, omit the FILE subcommand from the DATA LIST command and separate the inline data from the command lines with the BEGIN DATA and END DATA command. For example,

```
DATA LIST FREE /ID VAR1 TO VAR4
BEGIN DATA
1 .7 1 0
2 1.0 1 1
3 .09 1 0
4 1.3 1 1
5 2.1 0 0
6 3.1 0 1
END DATA
```

Missing Values

Sometimes information for a particular variable is not available for a case. You can assign a special value for these cases, and instruct SPSS to flag these values with the MISSING VALUES command. The SPSS statistical procedures and transformation commands recognize this flag, and those cases with missing values are handled in a special way.

The specification on the MISSING VALUES command consists of a variable name (or variable list) and the specified missing value(s) in parentheses, as in the example below:

```
MISSING VALUES ID (-1, -99) VAR3 (-99) CHARV1 ('X') CHARV2 (' ')
```

Missing values for string variables must be enclosed in apostrophes or quotation marks and include any leading or trailing blanks. You can specify up to three individual missing values for each variable.

MISSING VALUES defines user-missing values, which should be distinguished from system-missing values (indicated by a period in the output). SPSS assigns a system-missing value whenever it encounters a value other than a number for a variable defined as numeric on the DATA LIST command. System-missing values are also assigned when new variables created with transformation commands have undefined values.

Transformations

You can perform data transformations ranging from simple tasks, such as collapsing

categories for analysis, to creating new variables based on complex equations and conditional statements.

The RECODE command tells SPSS to change the values for a variable as the data are being read. The command

```
RECODE VAR1 (0=9)
```

instructs SPSS to change all 0's found for VAR1 to 9's. The variable(s) to be recoded must already exist. You can specify as many value specifications as needed, enclosing each specification within parentheses, as in:

```
RECODE VAR1 (0=1) (1=0) (2=-1)
```

To recode the values of one variable and store them into another variable, use the keyword INTO, as in:

```
RECODE VAR1 (0=1) (1=0) (2=-1) INTO NEWVAR1
```

The recoded variables can be existing or new variables. If you use an existing variable, cases with values not mentioned in the RECODE specification are not changed. If you recode a variable into a new variable, cases with values not specified for recoding are assigned the system-missing value.

The COMPUTE command creates new variables through numeric transformations of existing variables. COMPUTE names the variable you want to create followed by an expression defining the variable. For example,

```
COMPUTE NEWVAR = VAR1 + VAR2 + VAR3
```

defines a new variable, NEWVAR, which is the sum of VAR1, VAR2, and VAR3.

This is a very brief overview of RECODE and COMPUTE. Refer to the SPSS documentation listed at the end of this handout for more details.

Variable and Value Labels

VARIABLE LABELS and VALUE LABELS commands supply information that is used for labeling SPSS output. Use the VARIABLE LABELS command to assign an extended descriptive label to variables. Specify the variable name followed by at least one comma or blank and then the label (up to 120 characters) enclosed in quotation marks, as in:

```
VARIABLE LABELS  TIME 'TIME TO DEATH IN DAYS'
                 AGE  'AGE IN MONTHS'
```

Use the VALUE LABELS command to provide descriptive labels for values. The VALUE LABELS command is followed by a variable name, or variable list, and a list of the values with their associated labels (up to 60 characters) enclosed in quotation marks, as in:

```
   VALUE LABELS AGE 1 '0-12' 2 '13-24' 3 'over 24'
```

```
/ SEX 1 'FEMALE' 2 'MALE'
```

SPSS System Files

Once you have defined your data file in SPSS, you do not need to repeat the data definition process every time you access the data in an SPSS program if you save the data as an SPSS system file. An SPSS system file is a self-documented file containing the data and descriptive information (referred to as the dictionary). The dictionary contains variable names and locations, variable and value labels, print and write formats, and missing-value indicators.

The `SAVE` command is used to save the data file you have defined. The only required specification on the `SAVE` command is the `OUTFILE` subcommand which identifies the name of the new SPSS system file. For example, the command

```
SAVE OUTFILE=spssdata.sav
```

added to the end of the example above, would create an SPSS system file in the current default directory called `spssdata.sav`. If the file is to be saved in a directory other than the current directory, the entire file specification must be enclosed in quotations.

Once you have created an SPSS system file, you can read it in subsequent SPSS programs with the `GET` command. The only specification required is the `FILE` subcommand, which identifies the name of the system file you want to use:


```
GET FILE=spssdata.sav
```

accesses the SPSS system file created above and `DISPLAY` causes the system file's dictionary to be displayed. If the file is to be saved in a directory other than the current directory, the entire file specification must be enclosed in quotation marks.

There are several advantages to working with SPSS system files over the raw data files:

- SPSS reads system files faster than raw data files.
- SPSS system files are self-documenting.
- SPSS system files reflect transformations applied to the data.
- SPSS system files use less disk space.

There are two disadvantages to using SPSS system files:

- SPSS system files cannot be read directly by software other than SPSS (with a few exceptions).
- SPSS system files can only be read by SPSS on the operating system they were created on. I.E. An SPSS system file created on Digital UNIX cannot be read by SPSS on VMS.

The second disadvantage can be overcome by converting your SPSS system file into a portable file. Portable files are discussed below.

SPSS Portable Files

At some point it may become necessary to read an SPSS system file on some operating system other than the one that created the data set. The process of moving an SPSS file from one operating system to another is referred to as *transporting* the file. In order to transport an SPSS portable file, you must first convert it to a format that can be recognized by the operating system that will read the SPSS system file. In fact, even if you are not planning to transport an SPSS system file, it is still an excellent idea to convert the system file to a portable file before moving the data to some other location like a 4mm DAT tape or CD for long term storage.

Converting an SPSS system file to a portable file requires the following two steps:

1) Issue a GET command to identify the SPSS system file you want to convert into a portable file. The following example illustrates a GET command for an SPSS system file with a filename of comp.sav. This command will be similar to the one described in the previous section:

```
get file="~/spssstuff/dissert.sav"
```

2) Issue an EXPORT command to identify the portable file. The following example illustrates an EXPORT command for a portable file to be saved with a filename of archive.por:

```
export outfile="archive.por"
```

Once you have created a portable file, you can then move the file to any other operating system and read the file into SPSS (called *importing* the file). *Make sure to use text mode when transferring the file.* The SPSS statement for importing an SPSS portable file is IMPORT. For example:

```
import file=”archive.por”
```

Reading ASCII Data with a Record Length Greater than 1024

In order to read ASCII data whose records are greater than 1024 columns, you must include a FILE HANDLE command containing the LRECL subcommand. LRECL specifies the width (in columns) of the records in the file. For example:

```
file handle school /name=”school90.txt” / lrecl=1275
data list file=school / caseid 1-5 yrscomp 1274-1275
```

Writing External Files from SPSS

Suppose you need to create an external file from an SPSS system file so you can process the records with another software package. The `WRITE` command may be used for this

purpose. Use the OUTFILE subcommand to specify the file for the output from the WRITE command. You must also specify a variable list as in the example below:

```
WRITE OUTFILE='~/mydata.dat'
 /ID 1-4 VAR1 6 VAR2 12-15 CHARVAR 17-25(A) VAR3 18-30
```

The OUTFILE subcommand and the variable list follow the same rules as the FILE subcommand and variable list of the DATA LIST command described above.

The following example retrieves the SPSS system file, SPSSSAVE.SPS, and from it, creates an external data file in the [.rawdata] directory named raw.dat:

```
GET FILE=SPSSSAVE.SPS
WRITE OUTFILE="~/rawdata/raw.dat" TABLE
EXECUTE.
```

The TABLE subcommand in the example above requests a format table on the listing file showing how the variable information is formatted.

Reading and Writing Compressed Data

Working with compressed files can save you a lot of disk space. This section describes how to read and write compressed raw (ASCII) data, compressed export files, and compressed system files with SPSS.

Reading Compressed Data

You can read compressed raw (ASCII) data, compressed export files, and compressed system files with SPSS on Digital UNIX computers using the INPIPE subcommand on the FILE HANDLE command. The FILE HANDLE command takes the following general form when used to read compressed data:

```
file handle fileref /inpipe="UNIX-command"
```

where *fileref* is the name by which you reference the file and *UNIX-command* is the name of the UNIX command for reading compressed data.

zcat is the UNIX command used to write out the contents of a compressed file in uncompressed format. Use zcat in conjunction with the FILE HANDLE command to read compressed ASCII data as in the following example:

```
file handle pipeit /inpipe="zcat nsfh01.asc.Z"
data list file=pipeit records=1 /
     mcaseid        1-  5
     momftlt5       6-  6
     sex           18- 18
     races        312-313
```

pipeit is the *fileref* in the above example and must also be specified on the FILE= subcommand of the DATA LIST command as illustrated.

Use zcat in conjunction with the IMPORT command to read a compressed export file as in the following example:

```
file handle pipeit /inpipe='zcat comp.por.Z'
import file=pipeit
```

Use zcat in conjunction with the GET command to read a compressed system file as in the following example:

```
file handle pipeit /inpipe='zcat comp.sys.Z'
get file=pipeit
```

Writing Compressed Files

You can write compressed system files with SPSS using the outPIPE subcommand on the FILE HANDLE command. Due to a bug in SPSS you cannot *write* compressed raw (ASCII) data or compressed export files. SPSS hopes to have this bug fixed in their next update.

The FILE HANDLE command takes the following general form when used to write compressed data:


```
file handle fileref /outpipe="UNIX-command"
```

where *fileref* is the name by which you reference the file and *UNIX-command* is the name of the UNIX command for writing compressed data.

compress is the UNIX command used to write a compressed file. Use compress in conjunction with the save command to write a compressed system file as in the following example:

```
file handle pipeit /outpipe='compress > comp.sys.Z'
save outfile=pipeit
```

SPSS Documentation

SPSS is documented in the following manuals:

SPSS Syntax Reference Guide

SPSS Advanced Statistics User's Guide

SPSS 6.1 Base System User's Guide, Part 1 and 2

SPSS 6.1 Syntax Reference Guide

SPSS Professional Statistics 6.1

SPSS Advanced Statistics 6.1

These manuals are circulated by the CDE Print Library, 4457 Social. For online documentation, type `man spss` at the UNIX prompt.

SPSS's web site, http://www.spss.com also has a lot of information including a support section which has a searchable database for finding answers to common questions.

To subscribe to the SPSS listserver, visit http://www.stattransfer.com/lists.html. This web site provides a subscription service to all the major statistical software listservers including SPSS. The SPSS listserver provide a depth of information and support that is essentially impossible for staff at any one institution (like ours) to duplicate.