Richard Gonzalez
Psych 613
August 29, 2022

# SPSS handout

This handout will help you get started with SPSS syntax. There are obviously many details that I cannot cover in these short notes but these these pages will help you start using SPSS. This handout was adapted from similar handouts prepared by Hannah d'Arcy (UM) and John Miyamoto (UW).

## How to get SPSS?

SPSS is available on most PC and Mac computers on campus (including most lab and floor computers in East Hall). It is available through the virtual sites website `http://virtualsites.umich.edu/.` If you want to buy SPSS for your own computer, you can check out the special rate through the Computer Showcase in the Student Union building (see syllabus). The annual license is about $45 for the premium student version.

## SPSS syntax

SPSS syntax is relatively simple though not straightforward if you are new to programming. Syntax provides a concise way to tell people exactly which analyses you ran, easily recreate analyses, or run analyses multiple times.

Some of you may want to use the pull down menus in SPSS. For some analyses the menu system is a fine interface, but the menu system will not permit you to do everything you need for this course. You MUST USE THE SYNTAX WINDOW in order to perform some of the later problem sets (the menu system won't be able to do everything you will be asked to do). You might as well learn the syntax now when things are relatively easy rather than later when the syntax will be more complicated.

For some features, however, it will be okay to use the menu system. For example, for most GRAPHS the menu system is fine.

You may prefer to enter data through the DATA EDITOR in SPSS rather than the syntax window method I will show below.

In these notes I will use capital letters for SPSS keywords. All SPSS commands must end in a period.

I can't promise that everything listed here will work exactly as advertised across all versions of SPSS and all platforms (MAC, PC, and UNIX). Everything here has been checked on all versions up to 21 for windows. I have not carefully checked all these commands on the Mac version. Let me know if any of the commands in this document no longer work as advertised in more recent versions of SPSS.

1. General tips

   (a) Each command must begin on a new line and end with a period.

   (b) SPSS syntax is case insensitive. Capitalized words are used to help the reader identify keywords and command names.

   (c) The values of string variables must be enclosed in single quotes.

   (d) Commands, sub-commands, and keywords can be abbreviated to 4 letters. Variable names must always be typed in full.

   (e) SPSS syntax lines can never be longer than 80 characters across.

   (f) All variable names must be unique. The use of capitals will not distinguish two variable names, so defining "name" and "NAME" results in an error because SPSS thinks you are trying to use the same variable name to define two different columns of data.

   (g) Subcommands are always preceded by a forward slash (/).

2. Opening the syntax window

   To open a new syntax window, just click on File/New/Syntax. A new window will popup and you can enter commands directly into that window. To execute the commands just click on the "play" button (just like on a DVD player) or click on Run. You can also save the commands into a file (click on File/Save or File/Save As). This simplifies life because you can come back to the syntax file at a later date and quickly make any changes.

3. Entering data in the syntax window

   Three SPSS keywords are needed: DATA LIST, BEGIN DATA, and END DATA. For example, the following syntax reads in data from 5 subjects:

```
DATA LIST FREE / subnum height.

BEGIN DATA
1 62
2 68
3 66
4 65
5 70
END DATA.
```

   The syntax above labels the first column "subnum" and the second column "height". Those are names of variables and you should choose names that make sense given the column of data they name. Later, when you use the word subnum in syntax, SPSS will use the column of data named subnum.

   The FREE option in DATA LIST allows any number of spaces between the numbers so, for example, this would work too:

```
BEGIN DATA
1    62
2         68
3    66
4        65
5   70
END DATA.
```

Try to keep data lined up in columns to avoid clerical errors. Some people like to use DATA LIST LIST because that forces you to specify the number of columns each variable uses (see the manual for details).

4. Labeling variables

   SPSS limits variables names to be no longer than 8 characters (that is why I used "subnum" rather than "subjectnumber" above). It is recommended that you attach labels to the variable names so that you remember what the variables are. For example, the following SPSS command will assign sensible labels to the variables:

```
VARIABLE LABELS subnum "Subject number"
       /height "Subject height in inches".
```

   This way when we forget what the variable name "subnum" means or we forget how "heights" was measured (e.g., feet, centimeters), we can look at the VARIABLE LABELS. This feature is a good way to document your variables.

5. Labeling values of a variable

   In the same spirit, it is a good idea to label codes used for group membership. For instance, if you used a variable that had a code of 1 to refer to men and a code of 2 to refer to women, you should enter labels so you have a record of which code refers to which sex. The VALUE LABELS feature allows you to do enter sensible labels.

   Here is a complete example of the ideas we have presented so far:

```
DATA LIST FREE / subnum height sex.

BEGIN DATA
1 62 1
2 68 1
3 66 2
4 65 2
5 70 2
END DATA.

VARIABLE LABELS subnum "Subject number"
       /height "Subject height in inches"
       /sex "Sex of subject".

VALUE LABELS sex 1 "Male" 2 "Female".
```

   The VALUE LABELS feature is very helpful in labeling the responses on a scale. For instance, if you use a seven-point scale where 1 is "strongly agree" and 7 is "strongly disagree" you can

attach the labels with the numbers so you won't get confused about the direction of the scale ("what did a rating of 1 mean again?").

```
VALUE LABELS rating 1 "Strongly agree" 7 "Strongly disagree".
```

6. Defining missing values

   Suppose that for some subjects you don't have their heights recorded and for some subjects you don't have their sex recorded. You may choose to make up arbitrary codes for missing data. For example, in the former you could choose to enter a height of 0, and for the latter you choose to enter a sex of 9. SPSS needs to be told that a height of 0 means missing, not that the subject has a height of 0 inches. Those two possibilities need to be treated differently.

   The MISSING VALUES command allows you to specify which values are to be treated as missing.

```
DATA LIST FREE / subnum height sex.

BEGIN DATA
1 62 9
2 68 1
3  0 2
4 65 2
5 70 2
END DATA.

MISSING VALUES height (0) sex (9).
```

   You can assign many variables the same missing value code at once by using this syntax (e.g., suppose you want to assign all number 4s in the data to represent missing values):

```
RECODE
  subnum height sex (4=SYSMIS).
EXECUTE .
```

   This version sets the number 4 to be the "system-wide missing code" so any number 4 appearing in any of the variables listed will be converted to missing data. A cool version of this command is to say "all" and all number 4's on any variable will be recoded to missing data, as in

```
RECODE
  all (4=SYSMIS).
EXECUTE.
```

   You should be careful using such sweeping commands because you may wipe out as "missing" those cases where number 4 is a legitimate observation or code. Select missing value codes wisely.

   The menu version of this command is "transform/recode/into same variable", paste the variables you want, then click old and new values, and assign to "system missing".

7. Reading external files into SPSS

   You may want to keep the data in a separate file and have the SPSS syntax call that file. Syntax such as this will do the trick (where filename.txt is a text file with three columns of data corresponding to subnum, height and sex).

```
DATA LIST FREE FILE="filename.txt"
 / subnum height sex.
```

You can search the web for additional details on how to read a tab or comma delimited file, an Excel file, or files from other statistics programs such as SAS.

8. Performing transformations

   Sometimes you will need to transform your data, e.g., you may want to change the scale of your data. You can use the COMPUTE command for this. Suppose you wanted to convert the height scores you entered in inches into height in feet. The following command creates a new variable called "heightft" (you are free to choose any name under the 8 character limit), which is the original height data divided by 12. The EXECUTE command is needed to make SPSS perform the operation.

   ```
   COMPUTE heightft = height/12.
   EXECUTE.
   ```

   Another transformation you may want to do $X^p$ (discussed later in the class). This can also be accomplished with the COMPUTE command

   ```
   COMPUTE xsquare = x**2.
   COMPUTE xcube = x**3.
   COMPUTE xsqrt = x**0.5.
   EXECUTE.
   ```

   There are also several functions built into SPSS, e.g., the sqrt(x) function which is identical to x**.5, ln(x) is the natural log, and log10(x) is log base 10. They are used the same way as above:

   ```
   COMPUTE newvar = ln(x).
   EXECUTE.
   ```

9. Rank transformations

   SPSS has a special command to convert a column of data into ranks. To convert variable x into ranks, and place the ranks into a new column named "newvar" do the following:

   ```
   RANK x
    /RANK INTO newvar.
   ```

   If you need the ranking done separately by a grouping variable, just change the first line to

   ```
   RANK x BY groupingvariablename
   ```

   Using the "BY" followed by a variable name is a common method in SPSS to define grouping or predictor variables.

10. Saving files and accessing existing files.

    With large data sets you will have lots of VALUE LABEL lines, VARIABLE LABEL lines, MISSING lines, lots of COMPUTE lines, etc. To avoid having all that clutter in your analyses, it is a good idea to save your data into an SPSS data file (the usual suffix is .sav).

```
SAVE OUTFILE='C:\FOLDERNAME\file.sav'.
EXECUTE.
```

where FOLDERNAME is the path to the directory that contains the file "file.sav". The next time you want to work on that file all you have to do is type the following syntax

```
GET FILE='C:\FOLDERNAME\file.sav',
EXECUTE.
```

and SPSS can work from a dataset just the way you left it so you don't have to deal with all that clutter.

This is a convenient feature when sending your data to other people. The *.sav file contains all the variable descriptions, variable labels, missing value codes, etc.

11. Recoding the values of a variable

    Sometimes you'll want to change the coding from how the original variable was entered. This syntax writes over the original variable.

```
RECODE
 ethnic (1=0) (2=1) (3=1).
EXECUTE.
```

12. Recoding into a new variable keeping the original variable intact

```
RECODE
 ethnic (1=0) (2=1) (3=1) INTO newethn.
EXECUTE.
```

    This version maintains the original variable ethnic and creates a new variable newethn that contains the recoded variables.

13. The RECODE command can also be used to create new categories from a variable. The following example creates a new categorization according to three height values.

```
RECODE
 height (LOWEST THRU 65 = 1) (66 THRU 67 = 2) (68 THRU HIGHEST = 3) into
  h_cat.
```

14. Obtaining descriptive statistics for continuous variables

```
DESCRIPTIVES
 VARIABLES = height
 /STATISTICS = MEAN STDDEV VARIANCE MIN MAX SEMEAN.
```

15. Getting frequencies for categorical variables

```
FREQUENCIES
 VARIABLES = sex.
```

16. Obtaining a cross-tabulation between two categorical variables

```
CROSSTABS
 /TABLES = sex BY h_cat
 /STATISTICS=CHISQ
 /CELLS=COUNT ROW COLUMN.
```

17. Producing a histogram

```
GRAPH
 /HISTOGRAM(NORMAL)=height
 /TITLE='Histogram of height'.
```

18. Creating side-by-side boxplots

```
EXAMINE
 VARIABLES=height by sex
 /PLOT = BOXPLOT
 /STATISTICS=NONE
 /NOTOTAL.
```

The subcommand /NOTOTAL prevents the printing of the single boxplot for both groups combined into a single sample.

19. Generating a scatterplot.

```
GRAPH
 /SCATTERPLOT(BIVAR)=height WITH income
 /MISSING=LISTWISE.
```

20. Obtaining a line graph

```
GRAPH
 /LINE(SIMPLE)=MEAN(height) by sex.
```

21. Getting a line graph with multiple lines (e.g., one for men, one for women)

```
GRAPH
 /LINE(MULTIPLE)=MEAN(height) BY sex BY h_cat.
```

22. SPSS has recently modified how graphs are produced. The commands I give above are called "legacy commands." The new approach to graphics is more flexible but leads to much longer commands. I suggest starting with the legacy commands and moving to the newer commands later in the semester.

23. Choosing a subset of the data to analyze

Suppose you only want to analyze the data for the male subjects

```
COMPUTE filt_var=(sex=1).
FILTER BY filt_var.
EXECUTE.
```

Now any data that is not associated with sex=1 is "filtered out" from any subsequent analyses.

24. Returning to the entire sample

    This is how you turn off filtering to have all the data available again (or to filter out something else, such as analyzing data for only the female subjects)

```
FILTER OFF.
EXECUTE.
```

25. Splitting observations into strata, so that future analyses will be repeated for each stratum separately.

```
SORT CASES BY sex.
SPLIT FILE BY sex.
```

    To return to the entire data file, type

```
SPLIT FILE OFF.
```

26. Using "if then" logic (note that the word "then" is never used in SPSS)

```
COMPUTE tallman = 0.
IF (sex=1 and height >=68) tallman=1.
EXECUTE.
```

    The first line creates a new column of all zeros called "tallman". The second line changes the value of tallman to 1 for those cases that satisfy the condition (sex=1 and height >= 68).

27. Reading characters in a data file

    SPSS can handle nonnumeric data as well. For example, instead of using codes 1 and 2 for sex you can use M and F. The (A1) below means "alpha data point with 1 character". If one column of data was 2-character state abbreviations of the subject's birth state (e.g., MI, CA, NY), you would use (A2) instead. Thus, A followed by the number of characters used. Note that because now we need to specify number of columns, we can't use FREE anymore. We need to specify the number of columns for the other variables too. Do that with "fortran statements" such as (F1.0) which means "Fortran statement with one digits and no decimal". As another example, the number 12.34 would be denoted as (F4.2) because it consists of four digits, two of which are to the right of the decimal.

```
DATA LIST LIST / subnum(F1.0) sex(A1) income(F4.2).

BEGIN DATA
1 M 11.11
2 M 22.22
3 F 33.33
4 F 44.44
5 M 55.55
END DATA.
```

Sometimes the data point itself is a long character with spaces, as when inputting subjects first and last name. In this case, use quotation marks to make sure that the first and last name are read in as a single entry. Otherwise, SPSS will think that the first name is one entry and the last name is a separate entry (though in some applications you may want to code first and last name into two separate variables).

```
DATA LIST LIST / subnum(F1.0) name(A17) sex(A1).

BEGIN DATA
1 "Richard Gonzalez" M
2 "William Brenneman" M
END DATA.
```

Note that I entered (A17) for the number of characters associated with the variable name. Assign the number of characters associated with the longest name.

It is important to use DATA LIST LIST rather than DATA LIST FREE. Otherwise peculiar things will happen. For instance, the following code will not run and produce an error message:

```
DATA LIST FREE / x y name(A8).
BEGIN DATA
1 2 "John"
3 4 "Mary"
5 6 "Bill"
END DATA.
```

but the following code will run (the difference is that the string variables are placed before the numeric variables):

```
DATA LIST FREE / name(A8) x y.
BEGIN DATA.
"John" 1 2
"Mary" 3 4
"Bill" 5 6
END DATA.
```

If you use DATA LIST FREE, then all the string variables need to appear before the numerical variables. These kinds of somewhat arbitrary aspects are common in all types of programming.

28. Assigning value labels to character variables

    The original character value labels must be in single quotes; the labels that are being associated with the values are in double quotes. Another SPSS peculiarity....

```
VALUE LABELS sex 'M' "Male" 'F' "Female".
```

29. Recoding the character values of a variable

    Sometimes you may want to change the characters M and F back to numeric data (or vice versa).

```
RECODE
 sex ('M'=1) ('F'=2) INTO newsex.
EXECUTE.
```

You must recode INTO a new variable because the original variable is of a different type (character or string) than the recoded variable (numeric).

Some versions of SPSS complain if there are spaces on either side of the equal sign. That is, ('F'=2) is okay but ('F' = 2) produces an error. I don't know why this is so; just another SPSS peculiarity.

30. Interacting with a word processor

Most of this section applies to all versions of SPSS, but there may be peculiar things that may happen depending on your specific setup. The variables include whether you are on a Mac or PC, the version of the operating system you are using (Windows, Mac OS, linux, etc), the version of SPSS you have, and the word processor you use (Word, Wordperfect, Notepad, etc).

SPSS works nicely with Word and Wordperfect. You can cut and paste sections of the output (including graphics) directly into your word processor[1].

You should always organize the SPSS output into coherent sections; the easiest way to do that is to cut and paste the relevant pieces into your word processing document. You should never turn in pages and pages of printouts directly from SPSS without thoroughly editing and commenting the output.

You can drag and drop graphs from SPSS into Word. On some versions of SPSS, you may have to click on "drag to copy" which is an option under the Edit folder once you double click on the SPSS Graph object.

Tables in the output may require some tinkering in order to copy. You may experience long tables that span more than 80 columns so that when you copy to your word processor the table is chopped off. Try the following. In spss syntax window type the one line command

```
SET width= 80.
```

and then run it. You could also do this through the menu system (under Edit/Options/Viewer and Edit/Options/Draft Viewer you will see a place to set line width of the output). Unfortunately, on some versions of SPSS this is not sufficient. You ALSO need to double click on the table you want to copy. After doubleclicking, go to the format folder and click on autofit. That will force the table to fit within the width=80 columns. The table will then be reformatted to fit within the width specification. Now, you can copy to the word processor.

Another thing you could try is to copy the table as an object (rather than lines of text). To do this, make sure you set width=80 and autofit the table as described above. Then put the mouse over the table, click once, go to Edit, click on "Copy Objects" (those of you with a multi-button mouse can click on the rightmost button instead), and then paste the object into your word processor. This method has the advantage of reproducing the nicely formatted table. However, for the table to fit into your wordprocessor you need to use the autofit procedure above so that the table doesn't get chopped off.

Yet another thing to try if all else fails, open up a draft viewer session in SPSS by clicking on File/New/Draft Output. This opens up a new window to which all text output will go. The text will not be formatted, but it is standard text (save for cutting and pasting into email or a word processor). You can go into View/Edit/Draft Viewer to do some further tweaking (such

[1]Warning: a few students reported that cutting and pasting from SPSS to Word on the Windows system unexpectedly and for no apparent reason caused the system to crash and the word file was permanently corrupted so that it could no longer be opened. I am not aware of what produced this problem or how to solve this problem. So if you cut and paste from SPSS to Word, backup frequently. It would be terrible to spend hours on a problem set only to lose everything after a paste from SPSS.

as unclicking "Display Box Characters" and putting a blank space for the row and column separators–this eliminates any annoying box graphics that may not transfer well in e-mail). To revert back to the regular output, simply close the Draft Output Viewer and return to the original SPSS Viewer.

If after pasting into your word processor the tables look jagged, you might try the Courier font. With this font your columns should line up correctly. If you use proportional fonts such as TimesNewRoman, the columns may not line up correctly. Within your word processor, you could use Courier for output and your favorite font, say TimesNewRoman, for your own annotations and comments.

There are other tricks to solving some of these problems but I tried to offer simple solutions that would work across different versions of SPSS and different platforms. If you discover some neat tricks, please send them to the e-mail group so that others can benefit.

31. Forcing the commands to print in the output

    One advantage of using a syntax file is that you have a record of all the commands you ran. If the data set changes at a future date, all you need to do is re-run the syntax file.

    It is a good idea to force SPSS to print out the syntax commands in the output (so the output contains a record of what commands produced it). This is done by the command

    ```
    SET printback=on.
    ```

    You could also set the printback option through the menu system instead of the SET command. Click on Edit/Options and go to the Viewer or Draft Viewer tab. Just click on the box that is labeled "Display commands in log". Depending on what version of SPSS you use, this menu may be slightly different.