Session 1, SPSS[1]

Class overview. Hints on error avoidance and detection. The importance of documenting programs. Tips on record-keeping. Reading simple raw data. Reconfiguring SPSS so that it gives more and clearer information. Some debugging techniques.

COURSE OVERVIEW

This class will mainly be an introduction to using SPSS. Since it is an introduction, I will start at a very basic level. Please don't be insulted if you already know what I tell you. Some of the other students in class may not, and I hope you will feel free to help them. I will also give suggestions about organization and documentation that will help with your research projects. And, I will present ways more easily make tables from your output.

This is a one credit class. Grades will not be assigned, and credit will be given to everyone who attends class and completes the assignments. I will give assignments that will give you the opportunity to practice what you have learned in class. I encourage you to work together on assignments, and to come to me for help. There will also be a few readings that I will assign.

SPSS has a nice "point and click" interface, but this class will focus on writing syntax. You can use "point and click" and then paste the code created by it into a syntax window. But, pointing and clicking only is not a good practice for serious work. You can create variables, or output files, and then not remember the how they were created. Saving a syntax file creates an important record for you.

This is not a statistics class, and I may not be able to help you with statistical problems that you have. We will cover some procedures that produce statistics frequently used by social scientists, but there are many procedures that we will not cover, and that I have not used.

Before we start talking about SPSS soon, I want to talk about topics that apply to programming in general.

ERROR AVOIDANCE AND DETECTION.

Programming is a mine field. There are hundreds of ways to make mistakes. (I know from first-hand experience.) Your goal is to catch them earlier rather than later. Creating code that look neat helps with this. Lining up variables, etc, will help you catch mistakes.

When you create code to do something, TEST IT. A single error in a keystroke can change make a HUGE difference. If you create a new variable, look at it and the old variable to make sure it is what you want and expect.

Beware of missing values. They provide multiple opportunities for mistakes. Sometimes data sets have values that always mean missing, but they have not yet been set to missing. Be sure to deal with them! For example, a missing value might be -99999 for numbers that are usually positive. Including these as valid values could result in very wrong answers. When creating new variables, make sure that you do not unintentionally include missing values in a valid value. TEST! Understand and examine your variables.

Examine your results critically. Is the number of cases what you expect? Do the results have face validity? If your findings strongly contradict what previous literature has shown, it could be because you have an earth-shattering finding, or you may have a programming error.

By default, SPSS for Windows is very stingy with information. You can edit the configuration so that it gives you more information. I recommend that you do this, and I will describe below how to do it. LOOK at the

[1]Prepared by Patty Glynn, University of Washington.

notes. It is boring, and you are anxious to push back the frontiers of science by looking at your output, but notes can hold very important information. Pay special attention to warning messages, and the number of observations.

DOCUMENTING YOUR PROGRAMS, AND ORGANIZING.

Documentation is very important. When you write your code, you know what you are doing and why. But, when you go back to it six months later, it may not be as clear. Use labels, titles, and documentation lines. The first line of your program should be a comment that includes the name of the program. This should be followed by as many comment lines as are necessary to help you remember where the data came from, and what the purpose of the program is.

Directory structure and supplemental notes are other important ways of helping with organization. When you start a project, you might want to create a directory for it. For example, when you start a new paper, choose a short word that you will associate with the paper. Create a directory on the disk where you will do your work for that paper, using that word. In that directory, create subdirectories as needed - one for your text and tables, one for your SPSS programs, and beneath that create a subdirectory for raw data if necessary, and your systems files, etc. Create a file called “anote”, in which you will store important information. Such as, when you create a table, include information about what SPSS programs created the output for the table. If other information was used for a table, document it in this note.

Another way to help with documentation is the file naming scheme that you use. You will probably run many programs before you finally arrive at the results you want to publish. When you have made this decision, name the programs that create the results so that you will be able to find them when it is time to revise. For example, the program that creates results for table 1 and 2 could be named “tabs1_2.sps”. When it is time to revise the paper, you will be glad that you took these steps.

We will spend the rest of the class today writing a short SPSS program, and running it. But, there are many menu items that we will not cover. Please spend a little time looking at the menus.

Now, lets bring up SPSS. Find the icon, and double click on it. You will see a something that looks like a spreadsheet. Explore and familiarize yourself with the menu options.

CONFIGURING SPSS TO GIVE MORE INFORMATION IN A MORE READABLE FORMAT.

You can configure SPSS. If you work on a machine that others use, they may to be more verbose (which I recommend). First, change the font to Courier New 10 wherever you have the opportunity. It will make your output more readable. Please see http://staff.washington.edu/glynn/config_spss.pdf for information about how to configure SPSS so that it will give you more information.

WRITING SYNTAX

Type the following in a syntax window. (Open a new syntax window with FILE, NEW, SYNTAX.) Once you have typed in the program and data, click on Run, and then All in the menu across the top. That submits the program. After submitting the program, examine contents of the output window. If you find that you made errors, correct them and run the program again. Save the program. Once everyone is finished, we will discuss what the program does, and interesting features of SPSS.

```
title 'prog1.sps' .

DATA LIST FREE / ID * name (A9) AGE (f3) hinch (f3) weight (f3) gender (A) .
BEGIN DATA
1 George 22 72 203 m
2 Frank 67 65 180 m
```

```
3 Sally 27 62 120 f
4 Michelle 33 66 145 f
END DATA .
variable label
 name   'First name'
 age    'Age in years'
 hinch  'Height in inches'
 weight 'Weight in pounds' .

list var = all .
freq var = all .
```

Note the periods. Periods are the main punctuation in SPSS. Putting a period in the wrong place, or not putting one where there should be one is a common mistake.

`title 'prog1.sps' .`

This line causes the words “prog1.sps” to be printed at the top of your output page.

`DATA LIST FREE / ID * name (A9) AGE (f3) hinch (f3) weight (f3) gender (A) .`

The data list line tells SPSS what data to expect, and what format it will be in.

`ID *` tells SPSS to read the variable ID and to use the default format (which is 8.2 - width of 8 and 2 to the right of the decimal).

`name (A9)` tells SPSS to read the variable “name” as a character variable, and to allow up to 9 spaces for the variable.

`AGE (f3)` tells SPSS to read the variable “age” as a number, and to allow a width of 3, with no information to the right of the decimal.

Homework: Before class next week, please do the following:

1. Read the document “Helpful Hints on Record Keeping for People Working on Projects” found at: http://staff.washington.edu/glynn/other.html

2. Read the document, “AN INTRODUCTION TO USING SPSS” which can be found at: http://staff.washington.edu/glynn/stat.html

3. Go to https://depts.washington.edu/csde1/ and look at the SPSS documentation that is available.

4. Practice using SPSS Look at the menu items. Be sure to look at the help available (a book with a question mark on the top right of the screen). Create a practice data set like the one included in this lesson. Make deliberate mistakes, look at the log so that you can become familiar with what errors look like, and debug your program.

5. Try to figure out how you would modify the program that we wrote in class today so that it would read raw from an external file.