STA9750 – Lecture I

OUTLINE

1. Welcome to STA9750!
    a. Blackboard
    b. Tentative syllabus
    c. Remote access to SAS
2. Introduction to reading data with SAS
    a. Manual input
    b. Reading from a text file
    c. Import wizard
3. Some basics!
    a. DATA vs PROC… "intuitively"
    b. Constructing new variables with IF statements
    c. Renaming variables
    d. Some PROCs:
        i. PRINT
        ii. MEANS
        iii. FREQ
        iv. SGPLOT

1. WELCOME TO 9750!

This course will cover SAS and some R: the weights are to be decided by you, to some extent.

Basic logistics:

- Make sure you can get into Blackboard and access the course!
- I'll upload handouts before class.
- Bring your laptop if you can.
- Starting Sept 5th, CIS/STA will provide laptops that you can use in class.
- The lectures will be very hands-on. I'll say a couple things at the beginning of lecture and let you work on the handouts as I walk around the room.

There is a tentative syllabus on Blackboard. The topics we cover are subject to change.

I want you to have your say in what we cover because I want this course to be useful. I'll send out polls to ask for your (anonymous) opinion somewhat frequently.

You can access SAS in your computer through a CUNY Virtual Desktop:

http://www2.cuny.edu/about/administration/offices/cis/virtual-desktop/

You can also go to computer labs on the 6th floor of the building where the library is.

As it was mentioned in class [thanks!], you can download a version of SAS as a student following the steps herein:

https://www.sas.com/en_us/software/university-edition.html

2. INTRODUCTION TO READING DATA WITH SAS

You can read data in all sorts of ways with SAS. Also, SAS has a lot of quirks in this regard: we could literally spend 10 lectures talking about how to read data, and we wouldn't be done [traditional courses on SAS spend 5-7 lectures on this topic.] I'll talk about some basics today and in Lecture II, and we'll cover some specifics as we try to read in real data. In general, I expect you to know the basics and then look for specific help as needed (which, in my experience, is the most efficient way of learning how to program).

MANUAL INPUT

You can input data "by hand":

```
DATA pol2;
      INPUT CITY $ 1-13 SO2 POP TEMP;
      DATALINES;
Phoenix        10 582 70.3
Little Rock    13 132 61
San Francisco 12 716 56.7
Denver         17 515 51.9
;
```

Steps:

1. You start with "DATA". What comes after is the name of the dataset.
2. Then, in INPUT, you name the variables.
3. Then, there's a DATALINES statement.
4. After that, input the data. I strongly recommend having "aligned" columns. Fewer headaches!
5. Categorical variables have to be entered ***carefully***. You have to put a $ after their name and, to avoid problems, specify the columns where the variables are.

If you follow these steps, you should be fine… Most of the time. If you have variables that are dates, things get a little more complicated. We won't get there now.

More on reading in categorical variables (with spaces etc.) can be found here:

https://stats.idre.ucla.edu/sas/faq/how-do-i-read-in-a-character-variable-with-varying-length-in-a-space-delimited-dataset/

READING FROM A TEXT FILE

If the columns are "aligned" (space-separated) as above, reading in the data is easy:

```
DATA pol3;
      INFILE 'C:\Users\victor.pena90\Desktop\Pollution.prn';
      INPUT CITY $ 1-17 SO2 POP TEMP;
;
```

Steps:

1. You start with "DATA". What comes after is the name of the dataset.
2. Then, in INFILE, you specify where the dataset is.
3. In INPUT, you specify the variables.
4. If there are categorical variables, make sure to include $ and the location of the variables.

Input Wizard (Recommended)

If the data are in "standard" formats (such as Excel spreadsheets, *.csv), you can use the SAS import wizard to read in the data! Simply go to File > Import data and follow the steps.

3. Some basics!

DATA vs PROC

- DATA: Input and modify data, create new variables
- PROC: Statistical analyses, printing, etc.

Constructing new variables with IF statements

Creating new variables from old ones is easy. For example:

```
DATA pol4;
      SET pol3;
      LENGTH SIZE $6 HOT $3;
      IF POP >= 1000 THEN SIZE='big';
      IF POP < 1000 and POP >= 300 THEN SIZE='medium';
      IF POP < 300 THEN SIZE='small';
      IF TEMP>=70 THEN HOT='yes';
      IF TEMP<70 THEN HOT='no';
RUN;
```

If you're creating categorical variables, don't forget to specify their length!

Renaming variables

In Lecture, I tried to rename variables on the spot, and it didn't work because I missed some parentheses. Let me show you how it's done.

Let's create some fake data set with 2 variables, which I call VAR1 and VAR2:

```
DATA test;
INPUT VAR1 VAR2;
DATALINES;
      1 10
      2 20
      3 30
      4 40
;
```

Here's how you rename the variables:

```
DATA test2;
      SET test (RENAME=(VAR1=F1 VAR2=F2));
```

Steps:

1. DATA block which starts with the name of the new dataset
2. SET, followed by the name of the dataset with the "old" column names.
3. Use the RENAME command in the (somewhat peculiar) way that is used above.

Some PROCs

You can print the dataset with **PROC PRINT** (if you don't specify anything in VAR, it prints all the variables).

```
PROC PRINT data=pol4;
      VAR POP SIZE;
RUN;
```

You can print subsets!

```
PROC PRINT data=pol4;
      WHERE SO2 > 50 or TEMP > 70;
RUN;
```

You can get basic descriptive statistics for quantitative variables with **PROC MEANS**, and more detailed analyses with **PROC UNIVARIATE**.

```
PROC MEANS data=pol4;
RUN;

PROC UNIVARIATE data=pol4;
RUN;
```

You can tabulate categorical data with **PROC FREQ**. Cross-tabulation can be accomplished by using "*".

```
PROC FREQ data=pol4;
      TABLES SIZE HOT;
RUN;

PROC FREQ data=pol4;
      TABLES SIZE*HOT;
RUN;
```

Some plots with SGPLOT

As I told you in lecture, there are many SAS libraries that produce graphics. My favorite is SGPLOT. It's pretty simple, as we saw. You can do histograms and scatter plots easily:

```
PROC SGPLOT data=pol4;
      HISTOGRAM TEMP;
RUN;

PROC SGPLOT data=pol4;
```

```
      SCATTER x=TEMP y=SO2;
RUN;
```

You can do vertical and horizontal bar plots:

```
PROC SGPLOT data=pol4;
      VBAR size;
RUN;
```

And you can do fancier things, such as scatter plots by some categorical variables, or stacked bar-plots that visualize bidimensional relationships between categorical variables.

```
PROC SGPLOT data=pol4;
      SCATTER x=TEMP y=SO2 / group = size;
RUN;

PROC SGPLOT data=pol4;
      VBAR size / group= hot;
RUN;
```