STA/OPR 9750
Introduction to SAS

Víctor Peña

Baruch College

OUTLINE

1. Welcome to 9750!
    a. Blackboard
    b. Tentative syllabus
    c. Remote access to SAS
2. Getting SAS
3. Introduction to reading data with SAS
    a. Manual input
    b. Reading from a text file
    c. Import wizard
4. 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!

Basic logistics:

- The course website is http://vicpena.github.io/sta9750/spring19/.
- Make sure you can get into Blackboard and access the course. You'll turn in your assignments there.
- I'll upload handouts to the website before or after class (depending on the topic).
- Bring your laptop to class if you can.
- Starting next week, CIS/STA will provide laptops that you can use in class.

You can find the syllabus on Blackboard. The topics we cover are subject to change.

We'll go through more logistics (such as grade assessment) in our first lecture.

You'll notice that people with very different backgrounds are taking this course: software engineers that don't know much about statistics, people with a strong mathematical and statistical background but no background in programming, etc. This is inevitable: 9750 is a required course for the MS program in statistics, and it is usually taken during the first semester in the program.

The MS program attracts people with very different backgrounds/stories. While there are a lot of great things that come with that, there are also some difficulties. For example, it's likely that you think I'm going too slowly or too fast at some point of the semester.

The main goal of this course is getting to a point where everyone in the MS program has a common background in computing and statistics that will be used throughout the program.

2. GETTING SAS

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.

You can download a version of SAS as a student following the steps herein:

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

3. 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 [some traditional courses on SAS spend 5-7 lectures on this topic.] We'll just cover some basics.

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 (you can also use CARDS).
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.

4. 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

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;
```

If you want to print the first 10 observations, the following code will do it:

```
PROC PRINT data=pol4 (obs=10);
      VAR POP SIZE;
RUN;
```

If you want to print, say, the subset of rows that starts at observation number 5 and ends at observation number 10, you can use

```
PROC PRINT data=pol4 (firstobs = 5 obs=10);
      VAR POP SIZE;
RUN;
```

You can also print subsets according to values of variables!

```
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;
```

4

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

There are many SAS libraries that produce graphics. My favorite is SGPLOT. 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 scatterplots by some categorical variables, or stacked bar-plots to visualize 2-dimensional 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;
```