STAT 5100 Handout #0.a: Course Overview & SAS Crash Course

Course Overview

- Want to predict a response Y from one or more predictors X_1, X_2, …
    - Note that Y, X_1, X_2, … are variable names, and correspond to different characteristics measured on each ‘individual’
    - In addition to getting good prediction, we also want to understand the contribution of each predictor
        - Is the contribution (or effect) real, or could it have been just due to chance?
        - What is the magnitude of each predictor’s effect?
- The nature of response and predictors determines the appropriate model
    - Units 1-7: Y is continuous, and X’s usually are
    - Unit 4: Some discussion of non-continuous X’s
    - Unit 7: Y is observed sequentially, causing some dependence
    - Unit 8: Y is discrete

    (See syllabus’s ‘Course Description’ and ‘Anticipated Order of Topics’ sections)
- NOTE: This course will not prepare you to analyze all types of data, just some very common types found in a wide range of application areas.

About SAS

SAS is a software package used by many companies and researchers. We will focus in this course on the statistical tools available in SAS. This document will give you a quick crash course in some of the basics of SAS operation. SAS has three main windows:

- Editor (or Code): type and run commands here; “color” matters; .sas files open here
- Log: tells what was “done”; helps with debugging; “color” matters
- Results Viewer: see results in HTML format. Run this code at the beginning of .sas file to get ‘nice’-looking output there:

```
        ods html image_dpi=300 style=journal;
```

- [Output]: see results in simple text format
- [Graph]: shows nice graphs

To “run” a SAS program (or a highlighted section of SAS code), click on the “running man” icon, or hit F8 (which will unfortunately resize windows).

Read in Data

- By “hand”:

```
data a1;
   input x y;
   cards;
    1 2
    1 4
    2 3
    3 2
    ;
run;
```

Define a data set called a1.
Into this data set, input variables x and y.
Here come the data; could use “datalines” instead of “cards”.

Notice that data are entered in columns (but they don’t need to line up; it just looks nice); line breaks matter.

- Infile statement:

```
C:\folder\datafile.txt

    1 2
    1 4
    2 3
    3 2
```

```
data a1;
  infile 'C:\folder\datafile.txt';
  input x y;
run;
```

- PROC IMPORT (After saving C:\folder\datafile.xlsx as C:\folder\datafile.csv)

C:\folder\datafile.xlsx

x	y
1	2
1	4
2	3
3	2

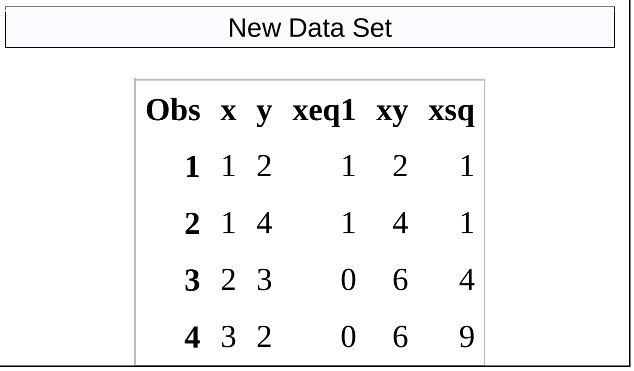
```
proc import
   datafile="C:\folder\datafile.csv"
   dbms=csv out=work.a1 replace;
  getnames=yes;
  datarow=2;
run;
```

Necessary components to run a program

- semi-colon at the end of every line (except for datalines)
- a data statement that names your data set (unless you import the data set)
- input statement (unless you import the data set)
- at least one space between each word or statement
- a run statement

Sample SAS code to create and view a new data set

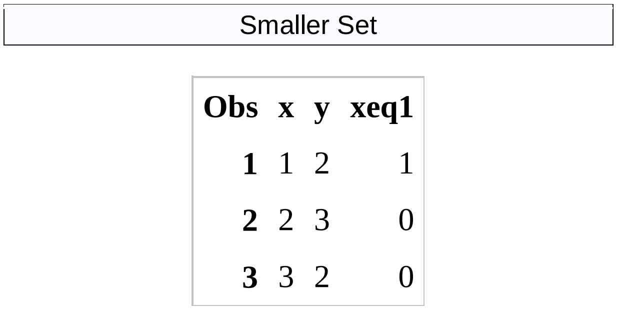
```
data a2; set a1;
  xy = x*y;
  xsq = x**2;
  xeq1 = 0;
  if x = 1 then xeq1=1;
run;
proc print data=a2;
  var x y xeq1 xy xsq;
  title1 'New Data Set';
run;
```



```
New Data Set

Obs  x  y  xeq1  xy  xsq
  1  1  2     1   2    1
  2  1  4     1   4    1
  3  2  3     0   6    4
  4  3  2     0   6    9
```

Sample SAS code to only keep certain observations in a new data set

```
data a3; set a2;
  if y < 3.5;
    /* Default 'then' is keep */
run;
/* Same as: */
data a3; set a2;
  if y >= 3.5 then delete;
run;
proc print data=a3;
 var x y xeq1;
 title1 'Smaller Set';
run;
```



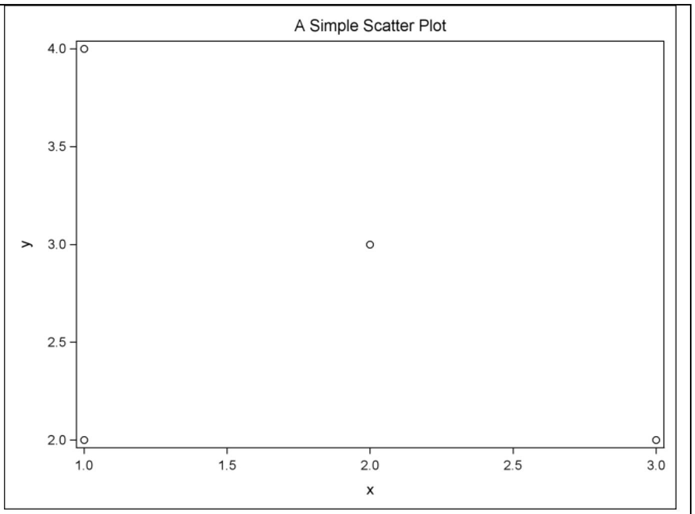
```
Smaller Set

Obs  x  y  xeq1
  1  1  2     1
  2  2  3     0
  3  3  2     0
```

Procedures

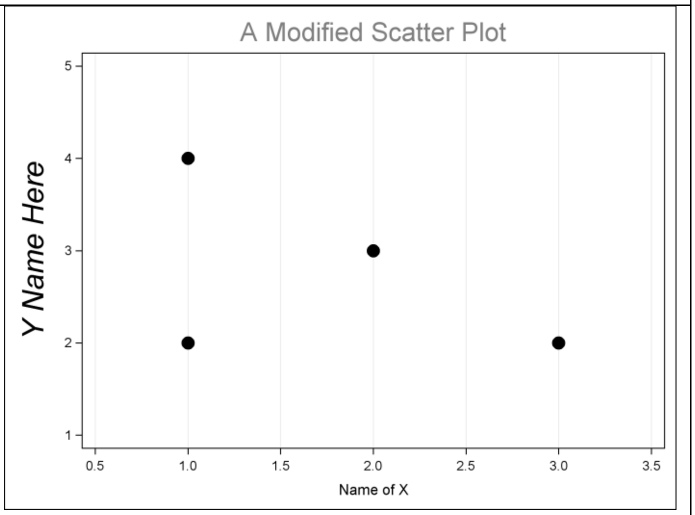
Data steps are used to create, read in, and manipulate data sets. Procedures (PROC's) are used to perform specific analyses or to create specific types of output (usually more than you need). We will introduce several PROC's in this class (we've already seen PROC IMPORT and PROC PRINT). Some PROC's we will use quite a bit include the following:

- Fitting models: PROC REG, PROC ARIMA, PROC LOGISTIC
- Graphical checks: PROC SGPLOT, PROC SGSCATTER, PROC BOXPLOT, PROC UNIVARIATE

```
proc sgplot data=a2;
  scatter x=x y=y ;
  title1 'A Simple Scatter Plot';
run;
```



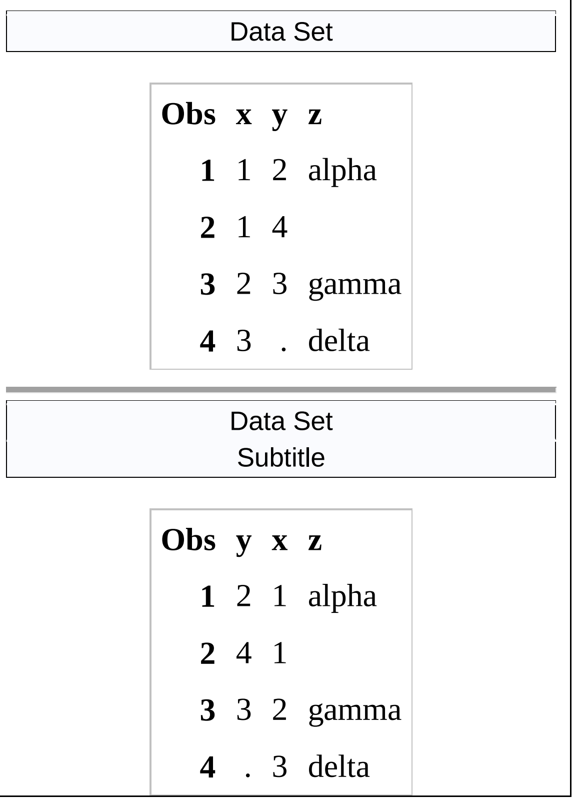
```
proc sgplot data=a2;
  scatter x=x y=y /
    markerattrs=(symbol='CIRCLEFILLED'
                 size=12);
  xaxis min=.5 max=3.5 grid
    label='Name of X';
  yaxis values=(1 to 5 by 1)
    label='Y Name Here'
    labelattrs=(size=20 style=ITALIC);
  title1 height=2 color=grey
    'A Modified Scatter Plot';
run;
```



Miscellaneous SAS Notes

- Export output by copying from Results Viewer and pasting into a word-processing document (control format of pasted objects with “Paste Options” and “Paste Special”).
- Help in SAS: The help facility in SAS (drop-down menu or the question-mark-book icon) can be useful if you have a good idea of what you want to do. You can see appropriate syntax and possible options for specific procedure, see examples of SAS code for specific analyses or graphics, and learn more about what a specific procedure option does.
- Code Continuity: Code can be written across lines; SAS only looks for semicolons to break up code, except for data lines. To read in data “continuously”, use @@.
- Missing Values: SAS procedures will completely ignore an observation if one of the called variables is missing; to code a value as “missing”, use the period (.) character.
- “Strings”: Read in character variables with $ after the name in the input line.
- Comment Lines: To comment out a line up to the next semi-colon, put an asterisk (*) before it. To comment out an entire section, start it with /* and end it with */
- Selective Output: SAS will usually give you more than you want, so you'll need to know what you want in order to do anything useful with the output.
- Save Code: Save your code (from the Editor window) as a .sas file in whatever location you choose; double-click on file icon to open SAS; run to recreate all results.

```
 data a1; input x y z $ @@; cards;
   1 2 alpha    1 4 .
   2 3 gamma    3 . delta
   ;
 run;
 proc print data=a1;
  * var y x z;
  title1 "Data Set";
 run;
 /*
 proc means data=a1;
   var y;
   title1 "Means Output";
 run;
 */
 proc print data=a1;
   var y x z;
   title2 "Subtitle";
run;
```



```
Data Set

Obs  x  y  z
  1  1  2  alpha
  2  1  4
  3  2  3  gamma
  4  3  .  delta

Data Set
Subtitle

Obs  y  x  z
  1  2  1  alpha
  2  4  1
  3  3  2  gamma
  4  .  3  delta
```

SAS Web Editor

This is a free version of SAS available for coursework, and it is currently in Beta testing. It has been enabled for this course (and may only work for Windows), but may still have some bugs.

Before you can use it, you'll need to register for "SAS OnDemand for Academics" and then access SAS Web Editor (Beta). Start by following the instructions here:
http://support.sas.com/ondemand/manuals/SASWebEditorStudent.pdf
Somewhere in there you'll need to indicate your affiliation with this course. Also, I think your screen will look a little different on page 14, since you shouldn't have to install the Enterprise Miner. When you get to that point, you'll just have to click on 'Run Client', which will take you to the actual SAS Web Editor login page.

After you have completed this registration process, you can log in at the following website:
https://academic93.oda.sas.com/SASWebEditor/index

I think the first time you log in there, it might take a few minutes to start up. You might also get an error message about the server not being available; just try again a few minutes later (this is running on a SAS server in North Carolina). Since this Web Editor in 'Beta', there might still be some bugs, but I was able to log in and run code there just fine (eventually). It seems to work better in some browsers (like Firefox) than others (Explorer). I don't plan on putting any data on the SAS server, and the INFILE statement won't read in any files on your local machine (or on my website), so any data sets you create in SAS Web Editor will have to be input 'by hand' using a CARDS or DATALINES statement (which will make for longer code for large data sets. I also don't know how permanently you can save your code on the server (I do know you can create ‘Project’ folders with saved .sas code), so make sure to keep a local copy of your .sas files.

There is additional support information here:
http://support.sas.com/ondemand

I hope this SAS Web Editor is a helpful addition to the course, but if it becomes a distraction for you, just go back to using the on-campus labs. I'll formally ask for feedback on this later in the semester (to see how it went and how things could be made smoother for future classes), but feel free to let me know how this goes in the meantime.

Citrix Receiver

You can also run SAS by going to http://apps.usu.edu and install the Citrix Receiver after logging in. You may need to be on campus or else running a VPN from off campus to get this to work. See more about the VPN at http://it.usu.edu/service-catalog/vpn. Some students have reported that running SAS this way is noticeably slower than other interfaces.

USU Software Licensing

Students can purchase a SAS license (http://software.usu.edu) for $70 (plus $25 for disks). The license runs 11/1-10/31 annually, NOT a year from your purchase date. I am not recommending this (the campus computer labs are already paid for [by you] and should be sufficient), but some students will see it as an attractive option. You must create your account online and add the product to your cart. Then, you need to go to the Service Desk (JQL North Entrance) to fill out a student use agreement and pay for the product.

SAS 9.3 and HTML Output Files

Beginning with version 9.3, SAS avoids the Output and Graphics windows in favor of the Results Viewer window. This window is actually a .htm file in the background. If you want to clear out the Results Viewer window during a SAS session (like if you have done a lot of back and forth with changing code and there is a lot of clutter in that window), run this:

```
/* Clear current windows and start a new (clean)
   Results Viewer window (may need to run this twice) */
ods html close;
dm 'log; clear; output; clear' continue;
dm 'log; next results; clear; cancel' whostedit continue ;
ods html newfile=none;
run;
```

After that, you can send new output to this clean Results Viewer file.

```
proc sgplot data=a2;
  scatter x=x y=y ;
  title1 'A Simple Scatter Plot';
run;
```

However, this really will generate a .htm (and for graphs, .png) file in the working directory (where your .sas file is located). After you save your .sas file and exit SAS, these files will not stay in the background, but will remain in that working directory. If you want to automatically delete them (and there might be a lot of them), you can run this:

```
/* Clear current windows and remove files created by Results
   Viewer  (WARNING: this will remove ALL .png and .htm files in
   the working directory */
options noxsync noxwait;
ods html close;
dm 'log; clear; output; clear' continue;
dm 'log; next results; clear; cancel' whostedit continue;
dm gsubmit "x 'del *.htm'; x 'del *.png' ";
run;
```