

# STAT 5100 Handout #1: Course Overview & SAS Crash Course

## Course Overview

- Want to predict a response  $Y$  from one or more predictors  $X_1, X_2, \dots$ 
  - Note that  $Y, X_1, X_2, \dots$  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
  - Is  $Y$  continuous or discrete?
  - Are  $X$ ’s continuous or discrete?
  - Has  $Y$  been observed sequentially, or are observations independent?

(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.

SAS is a software package used by many companies and researchers. We will focus in this course on the statistical tools available in SAS Studio (a free online version available for academic use). This document will give you a quick crash course in some of the basics of SAS operation. SAS has three main tabs or windows:

- CODE (or Editor): type and run commands here; “color” matters; .sas files open here
- LOG: tells what was “done”; helps with debugging; “color” matters
- RESULTS (or Results Viewer): see results in HTML format. On desktop version (not SAS Studio), run this code at the beginning of .sas file to get ‘nice’-looking output there:

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

To “run” a SAS program (or a highlighted section of SAS code), click on the “running man”:



### Read in Data

- By “hand”:  
(Easiest way when using free online version of SAS)

<pre>data a1;   input x y;   cards;   1 2   1 4   2 3   3 2   ; run;</pre>	<p>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”.</p> <p>Notice that data are entered in columns (but they don’t need to line up; it just looks nice); line breaks matter.</p>
--	---

- Infile statement: (file must be uploaded for SAS Studio; local files require local SAS)

<pre>../datafile.txt 1 2 1 4 2 3 3 2</pre>	<pre>data a1;   infile '/home/jrstevens/STAT5100/datafile.txt';   input x y; run;</pre> <p>[If local file, using desktop SAS]</p> <pre>data a1;   infile 'C:\folderpath\datafile.txt';   input x y; run;</pre>
--	--

- PROC IMPORT: (file must be uploaded for SAS Studio; local files require local SAS)

<pre>../datafile.xlsx</pre> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>x</th> <th>y</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>2</td> </tr> <tr> <td>1</td> <td>4</td> </tr> <tr> <td>2</td> <td>3</td> </tr> <tr> <td>3</td> <td>2</td> </tr> </tbody> </table>	x	y	1	2	1	4	2	3	3	2	<pre>proc import   datafile=   "/home/jrstevens/STAT5100/datafile.xlsx"   dbms=xlsx out=work.a1 replace;   getnames=yes;   datarow=2; run;</pre>
x	y										
1	2										
1	4										
2	3										
3	2										

### 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

↳ PROC IMPORT

### Sample SAS code to create and view a new data set

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

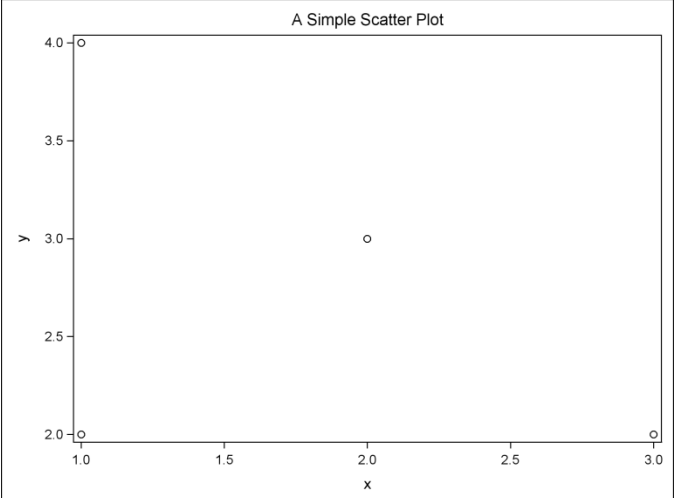
Smaller Set			
Obs	x	y	xeq1
1	1	2	1
2	2	3	0
3	3	2	0

## Procedures

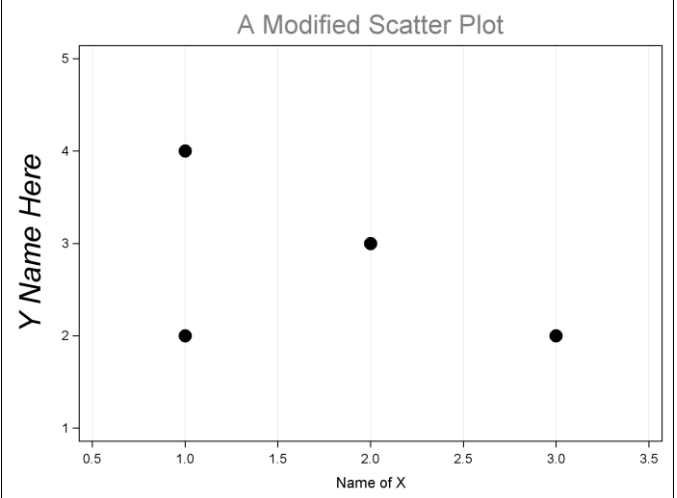
Data steps are used to create, read in, and manipulate data sets. Procedures (PROC's) are used to perform specific analyses or to create specific types of output (usually more than you need). We will introduce several PROC's in this class (we've already seen PROC IMPORT and PROC PRINT). Some PROC's we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

- Fitting models: PROC REG, PROC LOGISTIC, PROC ARIMA, others
- 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 (sometimes helps to use SAS Studio's "Open in a new browser tab" or "Download results as an RTF file" after enabling that option) and pasting into a word-processing document (control format of pasted objects with "Paste Options" and "Paste Special"; paste as "Device Independent Bitmap" seems to work best for plots). In Windows, the Snipping Tool is very helpful to copy plots.
- 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;
/* start comment
proc means data=a1;
  var y;
  title1 "Means Output";
run;
*/ end comment
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

## Accessing SAS (for free; paid license versions also available)

1. [SAS Studio; probably most convenient for most students, link from class website] SAS OnDemand for Academics is a free version of SAS available for coursework. It has limited functionality for data input, but should be usable for everything in this class. It requires an Internet connection every time you run it. Use the links on the class website (one to create an account, and another to login). This includes limited cloud storage, so you should be able to save homework code files there. Note that you must first “enroll” in a SAS course I created, using a USU-specific course code. A file in Canvas has enrollment instructions.
2. Campus Labs: SAS is available in the following student computer labs on the USU campus: Sci-Tech Library room 002, Ag Science room 135, and Family Life room L201. Graduate students in some departments have additional labs available to them.

I hope SAS Studio (or other free versions of SAS) work out for students, but I have no control over them, and cannot really help with installation or access issues. Depending on a variety of factors outside my (and your) control, sometimes they take a few minutes to open, and are noticeably slower than the licensed desktop versions. If these issues become a distraction for you, just plan on using the on-campus labs.

A note on older versions of SAS: In order to efficiently create many of the diagnostic plots we’ll rely on this class, you’ll need to have access to SAS version 9.3 or later. (These versions are automatic in all of the options listed above.)

## Getting Started with SAS

For most students, the best way to learn SAS is to start with a program that works (like from class examples in the notes), and then modify it as needed. See the “Starting Homework” section in the class syllabus, especially the warning against blindly modifying available sample SAS code. ALWAYS SAVE YOUR CODE!!! (You can save code in the free versions, too.)

- ① understand what you need to do
- ② Find an example that does something similar
- ③ Modify that example's code for your data
- ④ Get necessary output