McMaster University

Department of Mathematics and Statistics

STATISTICS 3A03: Applied Regression Analysis with SAS

Fall 2013

SAS Lab 1. Week of September 11-15, 2017

Topics Covered in this Lab

1. The SAS Editor.
2. Libnames
3. Importing data from plain text files using `PROC IMPORT`.
4. PROC PRINT.
5. PROC MEANS.
6. Saving your SAS output.

NB: In these notes, I will assume you are using SAS 9.4 for windows which is the version installed in the BSB computer labs. There may be issues with some of the code (particularly around reading in external data) if you are using the SAS University Edition through a virtual machine. Please come to my office hours with your laptop and I can try to help you if you have such issues.

1. The SAS Editor: When you open SAS 9.4 one of the windows available is the SAS editor. This is where you can write your SAS program. Remember that every command must end in a semi-colon (;). Most things in SAS are done using procedures which start with the word `PROC`. After writing the code for a procedure together with any statements that it requires, add a line with just `run;` which tells SAS that you want to run the previous procedure. To submit the code to SAS click on the Submit icon which looks like a running man! By default this will submit every command in the SAS editor; to submit just part of it use your mouse to highlight the appropriate part and click on the Submit icon. You can save your program using Save in the File Menu. Save it as a .sas file. You can edit this file using the Windows notepad editor. Some people prefer to edit their file using notepad and then copy and paste the code into the SAS editor to submit. Either is fine as long as you save your code.

2. Libnames: SAS refers to a specific location on your computer as a *library*. You will generally give your libraries names by which you will refer to them. This is done using the `libnames` command. It is generally a good idea to run this command as soon as you open up SAS. For example suppose that I wish to use a directory (folder) on a USB key for Stat 3A03. An external USB key is typically referred to as the `E:` drive by the computer and suppose that I have a directory called Stat3A03 on that drive. Then I would set it up as a SAS library by

```
libname S3A3 "E:\Stat3A03";
run;
```

A major advantage of setting up a library in this way is that SAS will save a dataset in the specified library using a special format. In subsequent SAS sessions it is not necessary to read in the dataset again, all you need to do is to make sure you start each SAS session with the `libname` command as

above. Note that the library name you use for a directory does not need to stay the same the next time you run SAS but it is generally best if you keep the same library name to ensure the same code will still run. A SAS dataset in a library should be referred to in the form <*libname*>.<*dataname*>. For example `S3A3.Example1` used below.

3. Importing Data from plain text files: To be accessible as a SAS dataset, we first need to import the data. In this course most of the data will be made available as plain text files on the course website. These can be imported into SAS using `PROC IMPORT`. For example suppose that we want to look at the Fish Length Data that we examined in class. The data can be found at
`http://www.math.mcmaster.ca/canty/teaching/stat3a03/Data/Fish.txt`
First we would save this file on our computer. Let us suppose that we have done this and saved the file in `E:\Stat3A03\Data\Fish.txt`. We can then load it into a SAS dataset in our library `S3A3` using

```
PROC IMPORT OUT=S3A3.Example1
            Datafile='E:\Stat3A03\Data\Fish.txt'
            DBMS=DLM;
   Getnames=YES;
   Datarow=2;
run;
```

The `OUT=S3A3.Example1` says that the SAS dataset should be saved in the library `S3A3` and will be called `Example1` (it can have any valid name) in that library. The `Datafile='E:\Stat3A03\Data\Fish.txt'` tells SAS where to find the text file with the data nd its name. Replace the path that I used with the correct path to where you saved the downloaded file. The `DBMS=DLM` says that this will be a delimited file. The default for delimited files is that there is a single space between the fields. These are all options in the `PROC IMPORT` statement and as such are part of a single statement so the semi-colon comes after these to end the statement. We have two other statements that modify how `PROC IMPORT` works. The first is `Getnames=YES;` which tells SAS to read the first row and use the character strings there as variable names. The second is `Datarow=2;` which tells SAS that the first row with data that we want to read is row number 2 in the file. Finally `run;` says we are finished with statements modifying `PROC IMPORT`. When you have completed this you should see that in the directory which you used in the `libname` statement earlier there is a new file called `Example1.sas7bdat`. This is the SAS dataset and will now always be available in the library so you will not need to import the data again.

4. PROC PRINT: This procedure allows us to print out the data to the SAS output window. This can be very useful to make sure that the data were imported correctly. You need to tell the procedure the name of the dataset.

```
PROC PRINT Data=S3A3.Example1;
run;
```

By default the procedure will print the whole dataset which is quite long in this case so we may only want to print the first few rows to ensure that the data were imported correctly. We can print the first five rows using

```
PROC PRINT Data=S3A3.Example1(obs=5);
run;
```

5. PROC MEANS: This procedure will calculate and display summary statistics such as the mean, standard deviation, minimum and maximum. We need to specify which variables in a dataset we want to look at using the `Var` statement.

```
PROC MEANS Data=S3A3.Example1;
Var Length;
run;
```

We can give other options to the procedure to calculate a number of other quantities such as

```
PROC MEANS Data=S3A3.Example1 N Mean STD Min Q1 Median Q3 Max;
Var Length;
run;
```

This will give the sample size, mean and standard deviation as well as the 5-number summary for each variable.

A useful statement in `PROC MEANS` is the `BY` statment which allows us to get information on one variable for different values of another variable. It is important, however, that the dataset is sorted by values of the variable defining the classes. The original dataset is not sorted by age so we must first sort and then call `PROC MEANS`. The `PROC SORT` will sort the dataset.

```
PROC SORT Data=S3A3.Example1 OUT=S3A3.Example1;
    BY Age;
run;

PROC MEANS Data=S3A3.Example1 N Mean STD Min Q1 Median Q3 Max;
Var Length;
BY Age;
run;
```

6. Saving SAS Output:

By default SAS output is sent to the Results window where you can look at it. Often, however, you will want to save it to another file outside of SAS. In SAS 9.4 (which is the version in the labs and the one we will be using throughout this course) you can save the output as a webpage. To do this, when in the SAS output window go to the File Menu and select Save As. Browse to the appropriate directory and save your output as a webpage.

Saving your output in this way will create a new file that can be viewed outside SAS. It will also allow you to delete bits you no longer want in the output (e.g. attempts that didn't work or didn't produce the correct output). You can also add any comments you may want on the output. This will be particularly useful for your assignments. Opening the file in Microsoft Word will also allow you to save the output in other formats (such as Word or pdf formats). Always make sure that you save your output before quitting SAS or it will be lost and you'll need to redo your analysis. It's also a good idea to save the contents of the SAS editor frequently. To do this just select the File menu when the editor is active and click on Save As. Then browse to where you wish to save the file and give it a name. It is best to keep the .sas extension as that will easily allow you to load the contents back into the SAS editor when you come back to it.

Questions to try

1. Import the fish length and age data from
   `http://www.math.mcmaster.ca/canty/teaching/stat3a03/Data/Fish.txt`.
   (a) Find the sample mean and variance of the ages and lengths of these fish.
   (b) Sort the data and use the `BY` statement in `PROC MEANS` find the mean and variance of fish lengths at each age as well as the number of fish at each age.
2. Import Pearson's Height data which you can get from
   `http://www.math.mcmaster.ca/canty/teaching/stat3a03/Data/Heights.txt`.
   (a) What are the names of the variables in the dataset?
   (b) Print out the first 10 rows of the data set.
   (c) Find the 5 number summaries for mothers and daughters heights.
   (d) Find 95% confidence intervals for the mean heights of mothers and daughters. (`CLM` in the `PROC MEANS` statement).