EXST SAS Lab
Lab #4: Data input and dataset modifications

Objectives

1. Import an EXCEL dataset.
2. Infile an external dataset (CSV file)
3. Concatenate two datasets into one
4. The PLOT statement will be introduced
5. Apply several previously used procedures (FREQ and CHART)
6. Subset the data with a WHERE statement
7. Use a CLASS statement and OUTPUT statement in PROC MEANS

More details on basic SAS Statements and Procedures

One of the strengths of SAS as a data analysis tool is its ability to read data from many sources, subset or combine data sets, and modify the datasets to accomplish various tasks. The most common types of external data sets used in SAS are EXCEL files (XLS extent), comma separated value files (CSV extent) and various space separate text files (PRN or TXT extent). A CSV file is actually a text file and can be read in any text reader (NOTEPAD or WORDPAD in Windows). In fact, the SAS files themselves, as well as the LOG and the LST files produced by a SAS by a batch submit, are also simple text files. After installing SAS you may find that clicking on a file with a LOG or LST extent opens them in SAS. You can request that Windows open these files by default in WORDPAD, which is much faster.

Reading external files: It seems that in recent years EXCEL, or a similar spreadsheet, has been the most popular program for data entry. Normally each observation is placed in a row and the variables in columns. At the top of each column there is usually a variable name. This arrangement is perfect for entry into most SAS programs. I would recommend a single row of variable names at the top where each column where each variable name is suitable for use a SAS variable name. SAS variable names can have up to 32 characters, letters or numbers, but cannot have any special characters except underscores (i.e. _). If the excel column heading has a blank SAS will substitute an underscore. SAS will not distinguish between variable names with upper or lower case. For example, LOCATION, location, Location and LoCatIon would all be recognized as the same variable name.

In the assignment this week there are two files to read into the program. One is an EXCEL file (.xls extent) and the other a comma separated values file (.csv extent). Start by saving these files into a directory. You could use the desktop directory (C:\Users\Lecture\Desktop), but you should probably create a new directory for this week's assignment as there are quite a few files involved.

Giving the full path of the data set to be accessed (e.g. C:\Users\Lecture\Desktop) always works. However, SAS keeps track of the currently active folder in windows. This folder is listed in the lower right hand margin of the SAS window and can be changed by clicking on the folder name and changing the location with windows menus. If data to be accessed is in the current folder the full path is not necessary. If the current folder in the example above is "C:\Users\Lecture\Desktop" then you would only need to list the data set name "Sharks (1998).csv" to access it and not the full path. The same is true of most statements that read or write output from SAS, such as INFILE and ODS statements. This is another reason why it is so useful to create a new subdirectory for each SAS program.

Image from wild life world website http://wild-life-world.blogspot.com/2013/11/shark.html;

Using SAS IMPORT: Open a SAS session and initialize your program with the usual statements. Then we will IMPORT the first of the two data sets. Using the menus at the top of the SAS window, do the following: [FILE > Import Data]. SAS will show a window giving a choice of file types for import. Keep the default, which is MS Excel and click the NEXT> button. Enter the name of the file you want to enter. I chose to browse, chose "COMPUTER", "Local Disk C" and then the directory where the data was saved. The example XLS file was called "Sharks (1995-97).xls" and was saved as a SAS file named SHARK95_97. There is only one sheet within our files so choose the first one, "Sheet1$". By default our file will go into the work library and will be lost when we finish. We could choose a permanent SAS dataset (e.g. SASUSER), but for his assignment the WORK library is adequate. As a member name I used the same name as the data file, "SHARK95_97".

At this point SAS is ready to create the dataset. SAS will offer to write a file containing the code needed to repeat the import process without using the menus. You do not need to save these as I have included the code in the example. If you prefer to use the code instead of the menus, you only need to change the name of the data file for input and change the name of the output file going to the WORK library.

Using INFILE**:** The INFILE statement applied to a CSV file is my favorite way to enter external data sets. Any EXCEL data, set up with the suggestions above (observations in rows with simple column headings), can be saved as a CSV file by simply choosing the EXCEL menu options "FILE > Save As" and then choosing the "Save as type" called "CSV (comma delimited)(.csv)". Avoid the other CSV types for Macintosh and MS-DOS.

The CSV file was called "Sharks (1998)" and I also stored it in the newly created MP directory. It is an update to the previous "Sharks95-97" and can be read with the following statements.

```
data Sharks98; length wt 8;
   INFILE 'Sharks (1998).csv' dlm=',' dsd missover firstobs=2;
   input Shark_No Year Month Day Sex $ PCL FL STL Wt;
datalines;
run;
```

The INFILE statement reads the dataset stored in the dataset named. The option "dlm=',' indicates that the "delimiter", or separator, for the data values is a comma. The option dsd indicates "delimiter-sensitive data", which means that if a comma exists within a variable, then values for that variable will be contained in quotes. For example, if the data set is comma separated, but one of the variables was NameLastFirst and was coded as "Geaghan, James", there would be a comma imbedded within the variable, and it is not intended as a delimiter. The dsd option causes the variable NameLastFirst to be contained within quotes in the dataset so SAS would recognize the variable value including the comma. The last part of the INFILE statement (firstobs=2) indicates that the first line contains variable names and is not to be read as data, so the first observation of data is actually in the second row.

Concatenation versus merging: There are two common ways of combining datasets, merging and concatenation. If you have two datasets that have the same observations, but different variables, the recombination would be a merge. For example, if one dataset has 100 observations with the date and

location of samples with data on catch of fish (the number in each species) and a second dataset has the same 100 observations with date and location information and has environmental information (temperature, depth, salinity), a merge is indicated. The following SAS statements would match the 100 observations in each dataset one on one and result in a combined total of 100 observations with all variables, date, location, catch of fish and environmental information all on the same record or data line.

```
PROC SORT DATA=FISHDATA; by date location; run;
PROC SORT DATA=ENVDATA; by date location; run;
DATA combined; merge fishdata envdata; by date location; run;
```

Concatenation is a simpler process. Two datasets that have the same variables, but different sets of samples, can be combined by concatenation. For example, if we have data for different years or different locations or different samplers and we want to combine the data into a single sample then concatenation is indicated. The following SAS statements would cause two data sets in my example (sharks95_97 and sharks98) to be concatenated with the second dataset appended to the end of the first dataset. In this example the combined dataset is called combined.

```
data combined; set Sharks95_97 Sharks98; run;
```

Concatenation would also work for more than 2 data sets.

This completes the data sets manipulation part of the assignment. These datasets are large and I would not recommend printing them for output as there are too many pages.

The remaining procedures are mostly procedures that you have used before with the exception of PROC PLOT.

```
proc plot data=Combined; plot stl * fl = sex;
   Title4 'Scatter plot';
run;
```

This PROC produces a scatter plot with STL on the Y-axis and FL on the X-axis. The usual symbols plotted are A for a single observation, B if a second observation occurs at the same (Y, X) coordinates, C for a third observation, etc. However, if you prefer a different symbol there are other options. The plot above will take the first letter of the variable SEX and use it as the plot symbol.

We have used PROC FREQ before. It is a useful procedure to explore datasets and get familiar with the distribution of observations in the data. It is also the procedure we will eventually use to do Chi square tests.

```
proc freq data=Combined;
   Title4 'Two-way frequencies of age and sex';
   table month * sex;
run;
```

The next task is to run two PROC CHART statements. The first should have a reasonable set of midpoints (look at the means statement to decide on the range) and will introduce the “SUBGROUP=” option. Notice that there were some observations that were not classed as either M (Male) or F (Female). Those individuals that were missing a value for sex were represented with a period. Missing values of quantitative variables are represented with a period in SAS data sets. Missing categorical variables are often represented as a blank space, but are show here as periods.

```
proc chart data=Combined;
   Title4 'Histogram (horizontal) of total lengths';
   Title5 'Sex specified as subgroup';
   hbar stl / midpoints=35 to 95 by 10 subgroup=sex; run;
```

The second PROC CHART introduces the "WHERE" statement to subset the data. In this case adding the statement "where sex = 'Female';" will produce output only for females with males omitted.

```
proc chart data=Combined; where sex eq 'Female';
   Title4 'Histogram (horizontal) of total lengths';
   Title5 'WHERE statement used to plot females only';
   hbar stl / midpoints=35 to 95 by 10 subgroup=sex;
run;
```

The equal in the where statement can be expressed as an "=" or as the character string "eq" (e.g. "where sex eq 'Female';"). Other options are "ne" (not equal to), so the missing values could have been eliminated with the statement or "where sex ne '';".

We have used PROC SORT to alter the order and appearance of printed data. However, another common use is to repeat a SAS procedure multiple times for a number of categories. The statements below will first sort the data by SEX and then produce a separate PROC MEANS outputs of the variable STL for each SEX.

```
proc sort data=combined; by sex; run;
proc means data=combined; by sex;
   var stl;
   Title4 'Proc MEANS by sex';
run;
```

A similar effect is obtained with the CLASS statement in PROC MEANS, but some procedures do not have class statements.

```
proc means data=combined; class sex;
   var stl;
   Title4 'Proc MEANS classed by sex';
run;
```

The default output statistics for PROC MEANS are: N, MEAN, STD, MIN, and MAX. However, the SAS OnlineDoc® 9.3 has a large list of other statistics that can be requested including most measures of variability and central tendency, sums of squares, percentiles and quartiles and confidence limits.

The final task for assignment 4 will be to use an OUTPUT statement. Most procedures, in addition to the output listed in the output window or the results viewer window, can output a SAS data set.

```
proc means data=combined noprint; class sex;
   var stl;
   Title4 'Proc MEANS OUTPUT classed by sex';
   output out=next01 n=n mean=mean range=range median=median;
run;
proc print data=next01; run;
```

The use of the class statement yields the requested statistics for each class (male and female) as well as a set of the same statistics for the two sexes combined. Two SAS variables, _type_ and _freq_ are also produced. Note that the observations with missing values for sex are omitted from the calculations.

If we had used a BY statement instead of a CLASS statement the PROC MEANS output would have included the requested statistics for the 3 levels of the by variable (i.e. missing, male and female) and no combined statistics for all sexes would have been produced.

Citation for SAS online documentation:
SAS Institute Inc. 2011. SAS OnlineDoc® 9.3. Cary, NC: SAS Institute Inc.

Assignment 4

We have data on fishes from several years. We want to concatenate the two years and perform a series of analyses. Complete the following assignment and pass in your LOG and you results from the OUTPUT window or RESULTS viewer.

1) You will want to include in your program the "usual statements" with option, comments and titles similar to those in previous assignments. (1 point)

As usual, include appropriate title statements. (1 point)

The dataset is from a MS Thesis from Nicholls State University in 2006 by Johnathan G. Davis titled "Reproductive Biology, Life History and Population Structure of a Bowfin, *Amia calva*, Population in Southeastern Louisiana. The data set (some variables have been omitted) was obtained on 9/11/2012 from: http://www.nicholls.edu/bayousphere/GraduateStudents/JDavis/Davis_Thesis.pdf.

2) The dataset is in two parts. The first part is an EXCEL dataset from 2005 and the second part is a CSV dataset from 2006. You can see the variable names in the datasets.

They are: Month Day Year Age Sex TL Wt. Note that sex is a categorical variable.

We want to read in the two datasets and concatenate them into a single data set. (2 points)

You probably want to label some variables. FYI, Age is in Years, TL is total length (mm) and Wt is weight (g).

3. Produce a scatter plot of weight by total length with sex plotted as the symbol (1 point)

4. Produce a two-way frequency table of age by sex. (1 point)

5. Sort by sex and get a PROC MEANS for each sex by using the "BY sex;" statement. Get means for the variables TL and WT using "VAR TL WT;". (1 point)

6. Produce a PROC CHART (horizontal bar) for the variable TL using midpoints (use the "MAX" and "MIN" values from PROC MEANS to get the range and use a midpoint interval of 50). Specify the option "subgroup=sex". Do not use a "BY sex;" statement. (1 point)

7. Using the same PROC CHART (horizontal bar) for the variable TL, add the statement "where sex = 'F';" Note that in this data set the sex is indicated as a single letter unlike the example data which was coded as "Male" and "Female". (1 point)

8. Get a PROC MEANS for the classes "sex" for the variable TL and output to a new data set. Then print that new data set. Include in the print data set the following variables: n, mean, min, max, median. The names given match the SAS keyword for the statistics. (1 point)

Image from the public domain image website:
http://www.public-domain-image.com/full-image/fauna-animals-public-domain-images-pictures/fishes-public-domain-images-pictures/bowfin-fish-image.jpg-free-picture.html