Lab 1: STAT 574S

Watch the first movie at: **http://www.ats.ucla.edu/stat/sas/notes/**
Download an example dataset "hs0.csv" from:
http://cals.arizona.edu/~anling/STAT574/datasets/
And save it into "T:\sas_data\"

1) Import data

```
 /* import data */
data hs0;
  infile 'T:\sas_data\hs0.csv' delimiter=',' dsd;
  length prgtype $10;
  input gender id race ses schtyp prgtype $ read write math
science socst ;
run;

 /* take a look at part of data*/
 proc print data=hs0 (obs=20);
 run
```

2) Draw dot plot, scatter plot, boxplot, histogram plot etc.

```
/*draw dot plot of "read" for category variable "prgtype" with
statistical limits added*/
proc sgplot data=hs0;
  dot prgtype / response=write stat=mean
            limitstat=stddev numstd=1;
run;

/* draw scatter plot with ellipse */
proc sgplot data=hs0;
  scatter x=write y=read;
  ellipse x=write y=read;
run;

/* draw vertical or horizontal boxplot*/
proc sgplot data=hs0;
  vbox write / category=prgtype;
run;

proc sgplot data=hs0;
  hbox write / category=prgtype;
run;

/* draw histogram with a normal density curve, and a kernel
density curve */
proc sgplot data=hs0;
  histogram write;
  density write;
  density write / type=kernel;
run;
```

```
/* draw bar chart*/
proc sgplot data=hs0;
  yaxis label="score";
  vbar prgtype / response=read;
  vbar prgtype / response=write
                 barwidth=0.5
                 transparency=0.2;
run;
```

3) Get basic descriptive info

```
/* get basic descriptive info for two variables "read" and "write
"*/
proc univariate data=hs0;
   var read write;
run;

/*The PLOT at the end of the line tells SAS to make additional
plots such as the stem and leaf and box plots.  */

proc univariate data=hs0 PLOT;
   var read write;
run;

/* check normality for the variable read using test and Q-Q
plot*/
proc univariate data=hs0 NORMALTEST;
   var read;
QQPLOT read/NORMAL(MU=EST SIGMA=EST COLOR=RED L=1);
run;

/* and use proc "capability" too */
PROC CAPABILITY DATA=hs0 NORMAL;
VAR read;
QQPLOT read /NORMAL(MU=EST SIGMA=EST COLOR=RED L=1);
RUN;
```

How to read/write Excel files in SAS?

Reading an Excel file into SAS

Suppose that you have an Excel spreadsheet called **auto.xlsx**. The data for this spreadsheet are shown below.

```
MAKE           MPG  WEIGHT PRICE
AMC Concord    22   2930   4099
AMC Pacer      17   3350   4749
AMC Spirit     22   2640   3799
Buick Century  20   3250   4816
Buick Electra  15   4080   7827
```

Using the Import Wizard is an easy way to import data into SAS. The Import Wizard can be found on the drop down **file** menu. Although the Import Wizard is easy it can be time consuming if used repeatedly. The very last screen of the Import Wizard gives you the option to save the statements SAS uses to import the data so that they can be used again. The following is an example that uses common options and also shows that the file was imported correctly.

```
PROC IMPORT OUT= WORK.auto1 DATAFILE= "T:\sas_data\auto.xlsx" DBMS=xlsx
REPLACE;
SHEET="auto";
GETNAMES=YES;
RUN;
```

- The **out=** option in the **proc import** tells SAS what the name should be for the newly-created SAS data file and where to store the data set once it is imported.
- Next the **datafile=** option tells SAS where to find the file we want to import.
- The **dbms=** option is used to identify the type of file being imported.
- The **replace** option will overwrite an existing file.
- To specify which sheet SAS should import use the **sheet="sheetname"** statement. The default is for SAS to read the first sheet. Note that sheet names can only be 31 characters long.
- The **getnames=yes** is the default setting and SAS will automatically use the first row of data as variable names. If the first row of your sheet does not contain variable names use the **getnames=no**.

Writing Excel files out from SAS

It is very easy to write out an Excel file using **proc export** in SAS.

Here is a sample program that writes out SAS data called mydata to an Excel file called **mydata.xlsx** into the directory "T:\sas_data".

```
proc export data=mydata outfile='T:\sas_data\mydata.xlsx' dbms = xlsx
replace;
run;
```