Lab 1: STAT 571B

Regular SAS edition: watch the first movie at: **http://www.ats.ucla.edu/stat/sas/notes/**
University Edition: watch: https://www.youtube.com/watch?v=pE5awNW53z8

http://support.sas.com/software/products/university-edition/faq/data_access_import.htm

http://www.sas.com/apps/webnet/video-sharing.html?player=brightcove&width=640&height=360&autoStart=true&playerID=1873162645001&playerKey=AQ~~,AAABs_kuvqE~,9q03viSCCi8Qu-ec7KH7e-bapzBTKVDB&videoPlayer=4262543790001&emptyPage=false

Download an example dataset "hs1.csv" from: D2L.
And save it into the shared folder "/folders/myshortcuts/MySAS_folder/" for SAS studio use.
"D:\sas_data\"

1) Import data

```
/* import data in regular SAS*/
data hs1;
infile 'D:\sas_data\hs1.csv' delimiter=',' dsd n=2 line=Linept;
  length prgtype $10;
  input gender id race ses schtyp prgtype $ read write math
science socst ;
run;
```

```
/* or import data in SAS Studio*/
FILENAME REFFILE "/folders/myshortcuts/MySAS_folder/hs1.csv" TERMSTR=CR;

PROC IMPORT DATAFILE=REFFILE
        DBMS=CSV
        OUT=hs1;
        GETNAMES=YES;
RUN;
```

```
/* take a look at part of data*/
proc print data=hs1 (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=hs1;
  dot prgtype / response=write stat=mean
            limitstat=stddev numstd=1;
run;

/* draw scatter plot with ellipse */
proc sgplot data=hs1;
  scatter x=write y=read;
  ellipse x=write y=read;
run;

/* draw vertical or horizontal boxplot*/
proc sgplot data=hs1;
  vbox write / category=prgtype;
run;

proc sgplot data=hs1;
  hbox write / category=prgtype;
run;

/* draw histogram with a normal density curve, and a kernel
density curve */
proc sgplot data=hs1;
  histogram write;
  density write;
  density write / type=kernel;
run;

/* draw bar chart*/
proc sgplot data=hs1;
  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=hs1;
   var read write;
run;

/* check normality for the variable read using test and Q-Q
plot*/
proc univariate data=hs1 NORMALTEST;
   var read;
QQPLOT read/NORMAL(MU=EST SIGMA=EST COLOR=RED L=1);
run;
```

4) T-test

```
/* one sample T- test, conf. interval is included in the results
*/
```

```
proc ttest data=hs1 H0=50;
  var write;
run;

/* change the alpha level */
proc ttest data=hs1 H0=50 alpha=0.1;
  var write;
run;

/* two -sample t-test, confidence interval for the difference of
two group means are included in the result */
proc ttest data=hs1 alpha=0.1;
  class gender;
  var write;
run;


/* paired t-test*/
proc ttest data=hs1;
  paired write*read;
run;

/* another way to get confidence interval for the mean of one
variable */
proc univariate data=hs1 cibasic(alpha=0.05);
  var write;
run;
```

---

5) 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= "D:\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;
```

6) SAS code for sample size/power calculation in two-sample comparisons

— Procedure : power

— The TWOSAMPLEMEANS statement performs power and sample size analyses for pooled and unpooled tests, equivalence tests, and confidence interval precision involving two independent samples.

[1]. Two-Sample *t* Test Assuming Equal Variances

You can use the NPERGROUP= option in a balanced design and express effects in terms of the mean difference, as in the following statements. Default values for the DIST=, SIDES=, NULLDIFF=, and ALPHA= options specify a two-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=diff
      meandiff = 7
      stddev = 12
      npergroup = 50
```

```
      power = .;
run;
```

You can also specify an unbalanced design by using the NTOTAL= and GROUPWEIGHTS= options and express effects in terms of individual group means:

```
proc power;
   twosamplemeans test=diff
      groupmeans = 8 | 15
      stddev = 4
      groupweights = (2 3)
      ntotal = .
      power = 0.9;
run;
```

Another way to specify the sample sizes is with the GROUPNS= option:

```
proc power;
   twosamplemeans test=diff
      groupmeans = 8 | 15
      stddev = 4
      groupns = (25 40)
      power = .;
run;
```

[2]. Two-Sample Satterthwaite *t* Test Assuming Unequal Variances

The following statements demonstrate a power computation for the two-sample Satterthwaite ***t*** test allowing unequal variances. Default values for the DIST=, SIDES=,NULLDIFF=, and ALPHA= options specify a two-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=diff_satt
      meandiff = 3
      groupstddevs = 5 | 8
      groupweights = (1 2)
      ntotal = 60
      power = .;
run;
```

Exercise:

Pilot study result:

```
data one;
input y type @@;
```

```
datalines;
65     1      64     2
81     1      71     2
57     1      83     2
66     1      59     2
82     1      65     2
82     1      56     2
67     1      69     2
59     1      74     2
75     1      82     2
70     1      79     2
run;

proc ttest data=one;
class type;
var y;
run;

 proc power;
      twosamplemeans test=diff
         meandiff = 2
         stddev = 9.3155
         npergroup = 10
         power = .;
   run;

    proc power;
      twosamplemeans test=diff
         meandiff = 1
         stddev = 9.3155
         npergroup = .
         power = 0.9;
   run;
```

The SAS System

The TTEST Procedure

type	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
1		70.4000	63.7729	77.0271	9.2640	6.3721	16.9125
2		70.2000	63.4995	76.9005	9.3666	6.4427	17.0998
Diff (1-2)	Pooled	0.2000	-8.5524	8.9524	9.3155	7.0389	13.7759
Diff (1-2)	Satterthwaite	0.2000	-8.5525	8.9525			

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F

Equality of Variances				
Method	**Num DF**	**Den DF**	**F Value**	**Pr > F**
Folded F	9	9	1.02	0.9744