Introduction to SAS Programming

Hoofar Pourzand

Email: rcc@rcc.psu.edu

Friday July 26th, Computer Building, 223

Overview

The Workshop consists of two separate parts. Part I will review basic data and proc commands in SAS in interactive mode on Hammer and as well as batch mode execution and parallel execution on LionX machines.
Part II will be for you to review what you've learned during the workshop. Please feel free to contact RCC with any questions you have regarding using our services.
Finally, we wish all of you good coding with SAS and we hope that you benefit from this hands-on workshop.

Important Notes:

a) Please use Exceed on Demand to connect to Hammer. You can download ExceedonDemand from here: http://rcc.its.psu.edu/user_guides/remote_display/exceed_on_demand/

b) Please make sure you've an activated account on Hammer.

c) Don't forget to bring a laptop with PSU wifi set up. You can set up your wifi from here: http://wireless.psu.edu/using.html

d) The workshop will be held at the Computer Building, 2nd Floor, Conference Room (Room 223)

Part I- Hands-on: Use Survey Complete Data

In this part, we will analyze the complete dataset survey which is available here.
As a general word of advice: Don't open your data files in CSV format (like *complete.csv*) with Excel and re-save it. Excel sometimes corrupt a couple of the variables depending on how they are provided, etc. It's safer to use notepad or wordpad to open such files for a visual review.

Steps:

1. [10 minutes] Read the complete dataset into SAS. I provide the SAS code to read the dataset below, so you can copy this code into SAS program editor and use it to read your dataset. You need to create a SAS library to save this dataset into a permanent SAS file, give a two-level SAS name to the dataset. Then, you need to read the SAS code and run it. Then, you need to check the log window, produce HTML output for the first 20 observations using SAS ODS, and then use PROC CONTENTS to check your dataset. Based on the output of PROC CONTENTS, we want to answer the following questions:
   How many observations in the dataset? 1004
   How many variables in the dataset? 119

   NOTE: (1) When you use the following SAS code, you need to change the disk directory of the dataset in the INFILE statement.
   (2) In the INPUT statement of the provided SAS code, '**Variable : $ 12.**' specifies that the variable is a character variable with length=12.
   (3) If you find something in the SAS code you didn't understand or wasn't explained in the workshop, please Google it or check SAS help and try to learn it by yourself.

```
/* read the data */
libname rccsas '\\winhpc\winfs\home\hup128\Downloads';
data rccsas.complete;
  infile '\\winhpc\winfs\home\hup128\Downloads\complete.csv'
              firstobs=2 delimiter=',' lrecl=2056 dsd;
      Input Gender $ Race : $10. HomeTwn : $12. Live : $10. Greek $
Parents $ GPA SkipCl
              TextSpd SitArea $ HrsStudy FornTeach $ HrsCompu PayEduc :
$20. GetEduc $
              HrsWork : $8. HubFood : $20. Relimport $ ReadBibl : $15.
RelPref : $12. God $
              Weight Height IdlWt IdlHt ViewWt : $12. Handed $ LongFing
$ EyeSight : $15.
              SelfAttr : $12. ShoeSize LooksImp MakeFrnds : $20.
DateRace $ InstMsgr
              PeopKiss : $20. DatePerly $ AgeDiff : $25. FutSpous :
$20. HowComm : $20.
              DateSpnd LongRels UseFaceBk $ AgeMarry CDsOwn Parties
CellPhn $ MinTalk
              FavMusic : $12. EntrPref : $12. CDsBurnd OwnIPod : $12.
MP3Playr : $25.
```

```
              MovyGenr : $10. Book_Movy $ SpendTVA : $12. HrsExer
SmokeCig $ HrsSleep
Abortion $ ExerPref : $14. WtTry $ FastDrvn SameSex : $20. IceTerm $
CitedOff
DrvnInfl $  BevPref : $10. DaysAlco SmokedMj $ Ecstasy $ LegalMj $
DrnksUsl $
SpndAlco : $14.  DrvnDrnk DrnkIntox DrnkPref $ Shroomed $ LiteBeer :
$20.
Cocaine $ MjBrowny $  MostBeer RolldMj $ Age1stDrk BushJob : $12.
AttkIraq $
PolAffln : $16. JoepaQuit $ TVSport : $14. HSSports $ Virgin $ SexNoRel
$
SexB4Mar $ CheatdSO $ CheatTel $  SexPartns DatesB4Sex AgeLostIt
SexPerWeek : $10. SameSex_1 $ UseProtn : $10. HrsPorn : $10. MinDatesB4
: $20.
OKSameSex $ SpeedDating : $12. SexUnsure $  PreMarSex $;

      weightKG = weight / 2.2;
      heightM  = height * 0.0254;
      bmi = weightKG / (heightM * heightM);


      if (PeopKiss = "n/r") then PeopKiss=' ';

      /*  Recode DatesB4Sex because it has too many categories  */
      length DatesBeforeSex $ 12;
      if ( DatesB4Sex <= 5 ) then DatesBeforeSex = put(DatesB4Sex, 3.
);
      if ((DatesB4Sex >  5) & (DatesB4Sex <= 10)) then
DatesBeforeSex='5 to 10';
      if ((DatesB4Sex > 10) & (DatesB4Sex <= 20)) then
DatesBeforeSex='11 to 20';
      if ((DatesB4Sex > 20) & (DatesB4Sex <= 50)) then
DatesBeforeSex='21 to 50';
      if ( DatesB4Sex > 50 ) then DatesBeforeSex='More Than 50';

      /*  Recode Religious Preference  */
      length ReligiousPref $ 12;
      /*  Code Everyone who isn't Catholic or Protestant
            (who did answer the question) to Other  */
      if ( ~missing(RelPref) ) then ReligiousPref = "Other";
      /*  Keep the Catholic & Protestant Groups  */
      if ( RelPref = "Catholic"   ) then ReligiousPref = "Catholic";
      if ( RelPref = "Protestant" ) then ReligiousPref = "Protestant";

      /*  Recode Frequency of Sexual Activity to Yes/No Indicator  */
      if ( ~missing(SexPerWeek)      ) then SexuallyActive = "Yes";
      if ( SexPerWeek = "Not active" ) then SexuallyActive = "No";

      /*  Recode MinDatesB4 to reduce number of Categories  */
      length MinDatesBeforeSex $ 12;
      if ( MinDatesB4 = "First"   ) then MinDatesBeforeSex = "1 to 3";
      if ( MinDatesB4 = "Second"  ) then MinDatesBeforeSex = "1 to 3";
      if ( MinDatesB4 = "Third"   ) then MinDatesBeforeSex = "1 to 3";
      if ( MinDatesB4 = "Fourth"  ) then MinDatesBeforeSex = "4 to 6";
      if ( MinDatesB4 = "Fifth"   ) then MinDatesBeforeSex = "4 to 6";
      if ( MinDatesB4 = "Sixth"   ) then MinDatesBeforeSex = "4 to 6";
```

```
      if ( MinDatesB4 = "Seventh" )            then MinDatesBeforeSex =
"7 to 9";
      if ( MinDatesB4 = "8 (on eighth date)" ) then MinDatesBeforeSex =
"7 to 9";
      if ( MinDatesB4 = "9 (on ninth date)" )  then MinDatesBeforeSex =
"7 to 9";
      if ( MinDatesB4 = "Tenth or later" )      then MinDatesBeforeSex
= "10 or later";
      if ( MinDatesB4 = "Not sexually active" ) then MinDatesBeforeSex
= "Not Active";

      /*  Coded Min Dates Before Sex  */
      if ( MinDatesB4 = "First"   ) then MinDatesB4Code = 1;
      if ( MinDatesB4 = "Second"  ) then MinDatesB4Code = 1;
      if ( MinDatesB4 = "Third"   ) then MinDatesB4Code = 1;
      if ( MinDatesB4 = "Fourth"  ) then MinDatesB4Code = 2;
      if ( MinDatesB4 = "Fifth"   ) then MinDatesB4Code = 2;
      if ( MinDatesB4 = "Sixth"   ) then MinDatesB4Code = 2;
      if ( MinDatesB4 = "Seventh" )            then MinDatesB4Code = 3;
      if ( MinDatesB4 = "8 (on eighth date)" ) then MinDatesB4Code = 3;
      if ( MinDatesB4 = "9 (on ninth date)"  ) then MinDatesB4Code = 3;
      if ( MinDatesB4 = "Tenth or later"      ) then MinDatesB4Code =
4;
      if ( MinDatesB4 = "Not sexually active" ) then MinDatesB4Code =
5;

      /*  Recode AgeLostIt to Exclude Virgins  */
      AgeLostVirginity = AgeLostIt;
      if ( AgeLostIt = 0 ) then AgeLostVirginity=.;
      if ( AgeLostIt = 3 ) then AgeLostVirginity=.;
      if ( AgeLostIt = 6 ) then AgeLostVirginity=.;
      if ( AgeLostIt = 8 ) then AgeLostVirginity=.;

      /*  Recode Use Protection to Exclude Non-Active Individuals  */
      useProtection = useProtn;
      if ( useProtn = "Not active" ) then useProtection=' ';

      /*  Recode Number of Sexual Partners Variable  */
      nSexualPartners = SexPartns;
      if ( nSexualPartners = 1027 ) then nSexualPartners =.;

      /*  Recode Live to get Frat or Not Frat  */
      if ( ~missing(live) ) then liveAtFrat = "No ";
      if ( live = "Frat"  ) then liveAtFrat = "Yes";
run;
/* produce html report using ods*/
ods                    html                    file                    =
'\\winhpc\winfs\home\hup128\Downloads\partone_complete.html';
proc print data = rccsas.complete (obs=20);
run;
ods html close; /* close the ods */
ods listing;
run;

proc contents data= rccsas.complete; /* check the data set*/
run;
```

The log file here:

```
NOTE: The infile '\\winhpc\winfs\home\hup128\Downloads\complete.csv' is:
      Filename=\\winhpc\winfs\home\hup128\Downloads\complete.csv,
      RECFM=V,LRECL=2056,File Size (bytes)=583664,
      Last Modified=19Jul2013:14:33:06,
      Create Time=19Jul2013:14:33:06

NOTE: 1004 records were read from the infile
      '\\winhpc\winfs\home\hup128\Downloads\complete.csv'.
      The minimum record length was 526.
      The maximum record length was 630.
NOTE: Missing values were generated as a result of performing an operation on missing
values.
      Each place is given by: (Number of times) at (Line):(Column).
      7 at 320:23   5 at 321:23   5 at 322:20   5 at 322:31
NOTE: The data set RCCSAS.COMPLETE has 1004 observations and 119 variables.
NOTE: DATA statement used (Total process time):
      real time           0.17 seconds
      cpu time            0.06 seconds


395
396  ods html file = '\\winhpc\winfs\home\hup128\Downloads\partone_complete.html';
NOTE: Writing HTML Body file: \\winhpc\winfs\home\hup128\Downloads\partone_complete.html
397  proc print data = rccsas.complete (obs=20);
398  run;

NOTE: There were 20 observations read from the data set RCCSAS.COMPLETE.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           0.15 seconds
      cpu time            0.14 seconds


399  ods html close; /* close the ods */
400  ods listing;
401  run;
402
403  proc contents data= rccsas.complete;
404  run;

NOTE: PROCEDURE CONTENTS used (Total process time):
      real time           0.01 seconds
      cpu time            0.01 seconds
```

***

The attributes coming off from the proc content:

```
The SAS System          15:39 Friday, July 19, 2013   1

                                The CONTENTS Procedure

      Data Set Name        RCCSAS.COMPLETE                    Observations
1004
      Member Type          DATA                               Variables             119
```

Engine	V9	Indexes	0
Created	Friday, July 19, 2013 04:30:16 PM	Observation Length	1192
Last Modified	Friday, July 19, 2013 04:30:16 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_64		
Encoding	wlatin1 Western (Windows)		

Engine/Host Dependent Information

Data Set Page Size	16384
Number of Data Set Pages	78
First Data Page	1
Max Obs per Page	13
Obs in First Data Page	3
Number of Data Set Repairs	0
Filename	\\winhpc\winfs\home\hup128\Downloads\complete.sas7bdat
Release Created	9.0301M0
Host Created	X64_ES08R2

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
60	Abortion	Char	8
84	Age1stDrk	Num	8
38	AgeDiff	Char	25
98	AgeLostIt	Num	8
116	AgeLostVirginity	Num	8
44	AgeMarry	Num	8
86	AttkIraq	Char	8
68	BevPref	Char	10
55	Book_Movy	Char	8
85	BushJob	Char	12
51	CDsBurnd	Num	8
45	CDsOwn	Num	8
47	CellPhn	Char	8
95	CheatTel	Char	8
94	CheatdSO	Char	8
66	CitedOff	Num	8
80	Cocaine	Char	8
37	DatePerly	Char	8
34	DateRace	Char	8
41	DateSpnd	Num	8
97	DatesB4Sex	Num	8
111	DatesBeforeSex	Char	12

The SAS System 15:39 Friday, July 19, 2013 2

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
69	DaysAlco	Num	8
76	DrnkIntox	Num	8
77	DrnkPref	Char	8
73	DrnksUsl	Char	8
75	DrvnDrnk	Num	8
67	DrvnInfl	Char	8
71	Ecstasy	Char	8
50	EntrPref	Char	12
61	ExerPref	Char	14
29	EyeSight	Char	15
63	FastDrvn	Num	8
49	FavMusic	Char	12
12	FornTeach	Char	8
39	FutSpous	Char	20
7	GPA	Num	8
1	Gender	Char	8
15	GetEduc	Char	8
21	God	Char	8
5	Greek	Char	8
90	HSSports	Char	8
27	Handed	Char	8
23	Height	Num	8
3	HomeTwn	Char	12
40	HowComm	Char	20
13	HrsCompu	Num	8
57	HrsExer	Num	8
102	HrsPorn	Char	10
59	HrsSleep	Num	8
11	HrsStudy	Num	8
16	HrsWork	Char	8
17	HubFood	Char	20
65	IceTerm	Char	8
25	IdlHt	Num	8
24	IdlWt	Num	8
35	InstMsgr	Num	8
88	JoepaQuit	Char	8
72	LegalMj	Char	8
79	LiteBeer	Char	20
4	Live	Char	10
28	LongFing	Char	8
42	LongRels	Num	8
32	LooksImp	Num	8
53	MP3Playr	Char	25
33	MakeFrnds	Char	20
103	MinDatesB4	Char	20
115	MinDatesB4Code	Num	8
114	MinDatesBeforeSex	Char	12

The SAS System 15:39 Friday, July 19, 2013 3

The CONTENTS Procedure

Alphabetic List of Variables and Attributes

#	Variable	Type	Len
48	MinTalk	Num	8
81	MjBrowny	Char	8
82	MostBeer	Num	8
54	MovyGenr	Char	10
104	OKSameSex	Char	8
52	OwnIPod	Char	12
6	Parents	Char	8
46	Parties	Num	8
14	PayEduc	Char	20
36	PeopKiss	Char	20
87	PolAffln	Char	16
107	PreMarSex	Char	8
2	Race	Char	10
19	ReadBibl	Char	15
20	RelPref	Char	12
112	ReligiousPref	Char	12
18	Relimport	Char	8
83	RolldMj	Char	8
64	SameSex	Char	20
100	SameSex_1	Char	8
30	SelfAttr	Char	12
93	SexB4Mar	Char	8
92	SexNoRel	Char	8
96	SexPartns	Num	8
99	SexPerWeek	Char	10
106	SexUnsure	Char	8
113	SexuallyActive	Char	3
31	ShoeSize	Num	8
78	Shroomed	Char	8
10	SitArea	Char	8
8	SkipCl	Num	8
58	SmokeCig	Char	8
70	SmokedMj	Char	8
105	SpeedDating	Char	12
56	SpendTVA	Char	12
74	SpndAlco	Char	14
89	TVSport	Char	14
9	TextSpd	Num	8
43	UseFaceBk	Char	8
101	UseProtn	Char	10
26	ViewWt	Char	12
91	Virgin	Char	8
22	Weight	Num	8
62	WtTry	Char	8
110	bmi	Num	8
109	heightM	Num	8
119	liveAtFrat	Char	3

The CONTENTS Procedure

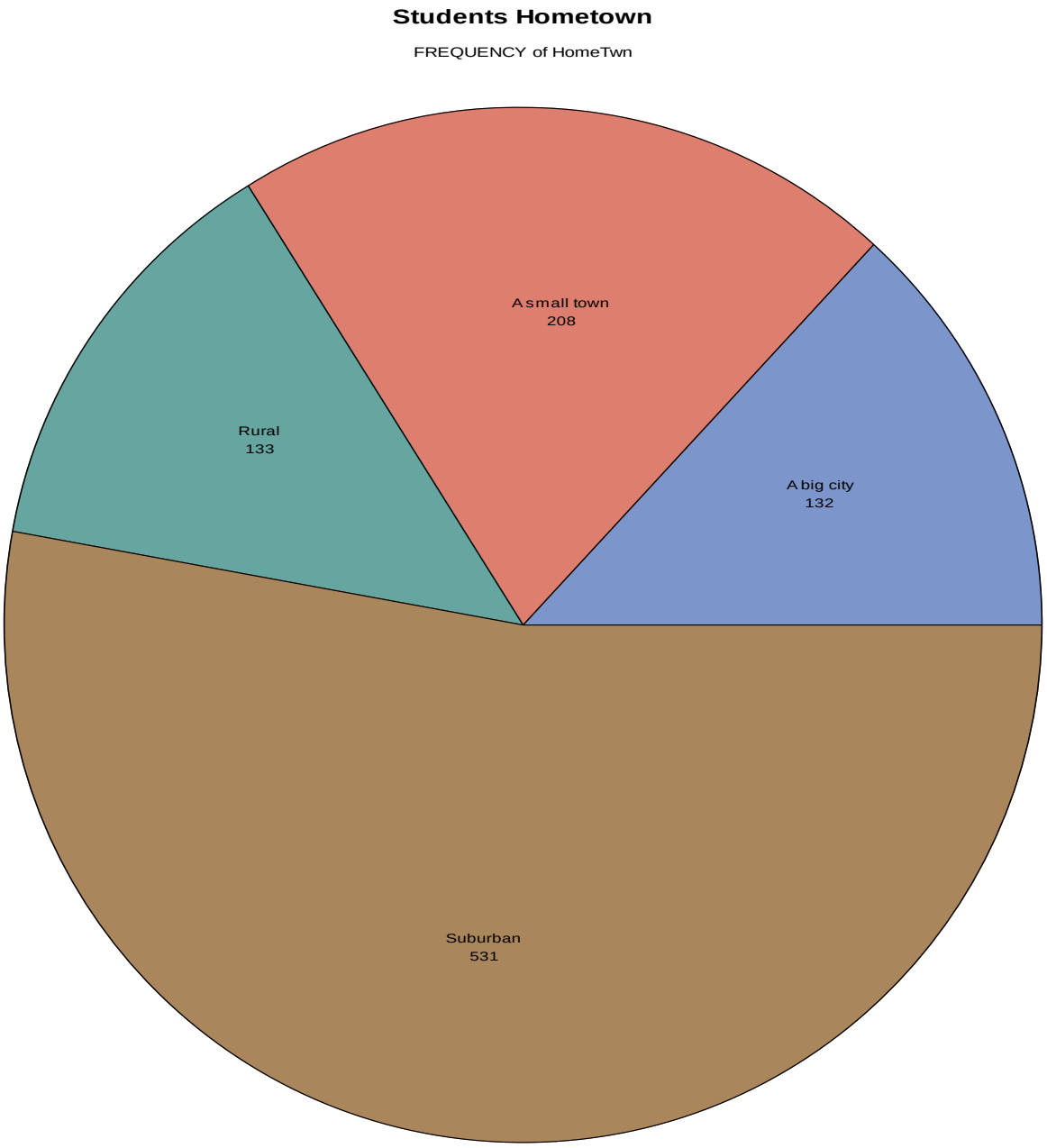
Alphabetic List of Variables and Attributes

#	Variable	Type	Len
118	nSexualPartners	Num	8
117	useProtection	Char	10
108	weightKG	Num	8

2. [20 minutes] In this exercise, we are interested in the features of demographic information. You need to use PROC GCHART to produce each chart and draw some conclusions.

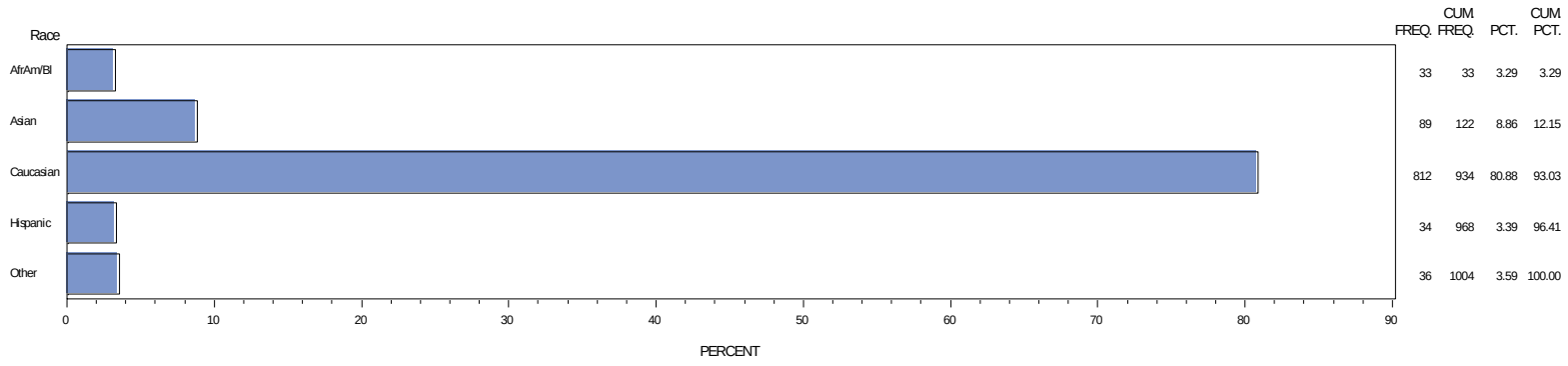
   i. [5 minutes] Let's produce a pie chart for students hometown (*hometwn*), in which you need to specify the value inside each slice of the pie.

```
proc gchart data= rccsas.complete;
pie hometwn /  value= inside;
title 'Students Hometown';
run;
```



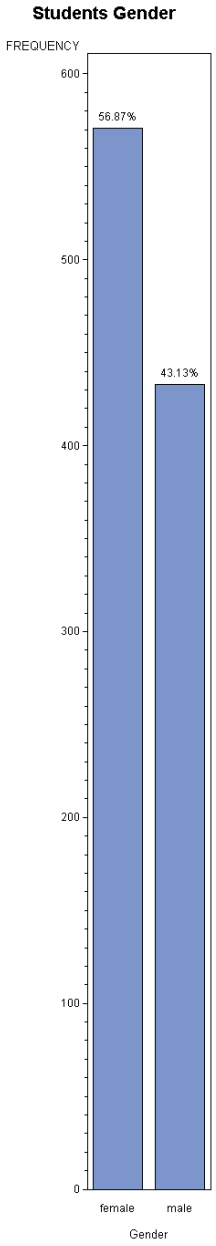
ii. [5 minutes] Let's produce a horizontal bar chart for students race (*race*).

```
proc gchart data= rccsas.complete;
hbar race / type = pct  inside=mean;/*plus extra info on the chart*/
title 'Students Race';

run;
```



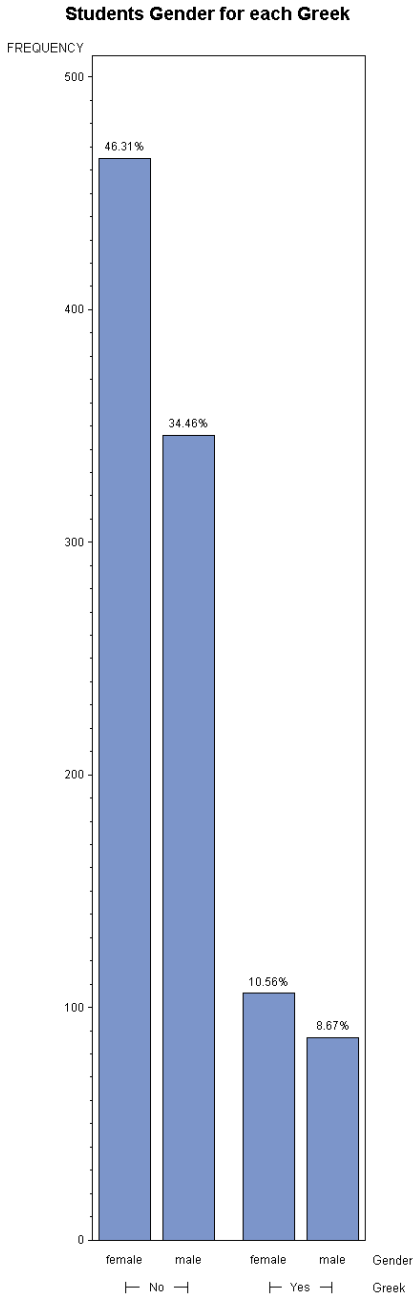
iii. [5 minutes] Let's produce a vertical bar chart for students gender (*gender*), in which you need to specify the percentage above each bar.

```
proc gchart data= rccsas.complete;
vbar gender /  outside=pct ;
title 'Students Gender';
run;
```



iv. [5 minutes] Let's produce a vertical bar chart for students gender (*gender*) for each level of the variable *greek*, in which you need to specify the percentage above each bar. (More Rope: use the Group=greek option. Let's Google it!)

```
proc gchart data= rccsas.complete;
vbar gender / Group= greek   outside=pct ;
title 'Students Gender for each Greek';
run;
```



3. [15 minutes] In this exercise, we are interested in testing the relationship between "Which following finger on your left hand is longer: your ring finger or index finger? " (variable name: *longfing*) and the variable (*gender*), between *longfing* and the shoe size (*shoesize*).
   i. [5 minutes] Let's produce the two-way frequency table for the variable *longfing* and *gender*. In the frequency table, you need to suppress the column percentage and row percentage. Perform a chi-square test to test the relationship between *longfing* and *gender* and draw your conclusion.

```
proc freq data = rccsas.complete;
tables longfing*gender /chisq nocol norow; /* suppress col & row*/
run;
```

15:39 Friday, July 19, 2013

The FREQ Procedure

Table of LongFing by Gender

LongFing Gender

Frequency Percent	female	male	Total
index	226 22.51	97 9.66	323 32.17
ring	254 25.30	283 28.19	537 53.49
same	91 9.06	53 5.28	144 14.34
Total	571 56.87	433 43.13	1004 100.00

Statistics for Table of LongFing by Gender

Statistic	DF	Value	Prob
Chi-Square	2	44.9960	<.0001
Likelihood Ratio Chi-Square	2	45.6715	<.0001
Mantel-Haenszel Chi-Square	1	10.3173	0.0013
Phi Coefficient		0.2117	
Contingency Coefficient		0.2071	
Cramer's V		0.2117	

Sample Size = 1004

The P value of <0.0001 is significant and therefore we can reject the Null Hypothesis and there for there is a dependency between longfing mean and the gender.

ii. [5 minutes] The variable *longfing* has three levels: ring, index, same. In this exercise, we are only interested in students who have different lengths of index and ring fingers. Thus, you need to create a new dataset which only contains students who have different lengths of index and ring fingers. In this new dataset, you only need to keep the following variables: *shoesize*, *longfing*, *gender*. Let's print out the first 20 observations of the new dataset to check it. [More rope: Let's use where statement or If-THEN-Delete statement]

```
data rccsas.complete_p3(keep=shoesize longfing gender);
set rccsas.complete;
where longfing <> 'same';

run;
ods html file='\\winhpc\winfs\home\hup128\Downloads\print_prob3.htm';
proc print data=rccsas.complete_p3 (obs=20);
run;
ods html close;
```

Obs	Gender	LongFing	ShoeSize
1	male	ring	6
2	male	ring	6
3	female	index	3
4	male	ring	6
5	male	ring	6
6	female	index	5
7	female	ring	3
8	female	index	4
9	male	ring	8
10	female	index	4
11	female	ring	4
12	male	ring	9
13	male	ring	6
14	female	index	3
15	female	index	4
16	female	index	3
17	female	index	3
18	male	index	9
19	male	ring	6
20	female	index	3

iii. [5 minutes] Let's perform a two-sample t-Test to test for a difference (two-sided hypothesis) in shoe size of those with longer index fingers vs. those with longer ring fingers.

```
   /* paired t-test (two-sided) */
data rccsas.complete_p4;
set rccsas.complete_p3;
length shoesizering 8;
length shoesizeindex 8;


if  (longfing = 'ring')  then shoesizering = shoesize ;
if (longfing = 'index') then shoesizeindex = shoesize ;
run;


proc ttest data=rccsas.complete_p4;
      paired shoesizering*shoesizeindex;
      title 'two sample paird t-test for shoesize:index vs ring';
   run;
```

4. [45 minutes] In this exercise, we are interested in the relationship between students GPA (*GPA*) and hours of study per week (*HrsStudy*).

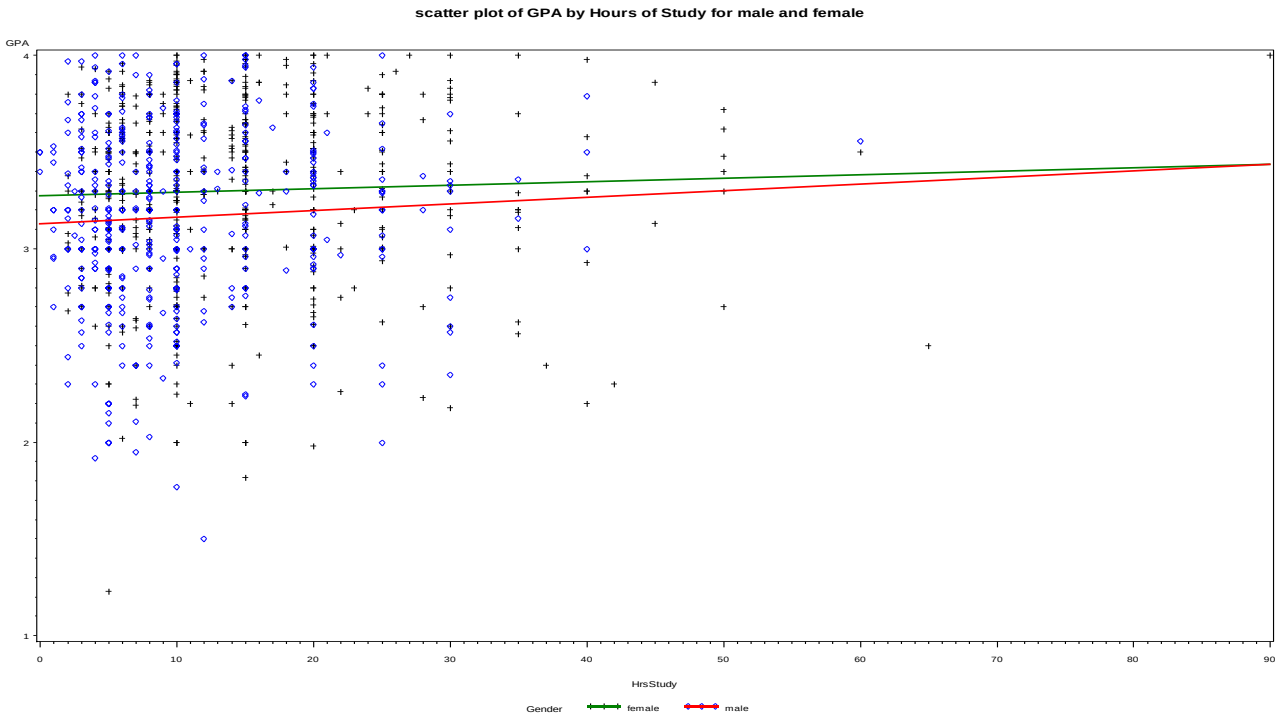
   i. [5 minutes] Let's compute the summary statistics for the variables *GPA* and *HrsStudy*, including the number of non-missing values, mean, standard deviation, median, minimum and maximum of both variables. Produce HTML output.

```
/*Note: check for the number of non missing values*/
ods html file='\\winhpc\winfs\home\hup128\Downloads\print_4_means.htm';
proc means data=rccsas.complete n  mean std median min max ; /*mean,
std, median, min & max */

var GPA HrsStudy;
run;
ods html close;
```

```
two sample paird t-test for shoesize:index vs ring                    8
                                                          13:19 Monday, July
22, 2013

                                The MEANS Procedure

Variable       N            Mean         Std Dev          Median         Minimum
Maximum
------------------------------------------------------------------------------
GPA          995       3.2392171       0.4626342       3.3000000       1.2300000
4.0000000
HrsStudy     999      12.4664665       9.3739997      10.0000000               0
90.0000000
------------------------------------------------------------------------------
```

ii. [7 minutes] Let's generate two separate scatter plots of *GPA* vs *HsrStudy* on the same graph for male and female students. Use the symboln statement to specify that the type of point is plus and color of point is black for female students; the type of point is diamond and color of the point is blue for male students. Add a green regression line for female students and add red regression line for male students in each separate scatter plot with the width=3. We also want the regression equation *GPA~HsrStudy* for female and male students, respectively .

```
/*4_ii: Plot a scatter plot of GPA vs HsrStudy.
Use the symboln statement as above.*/
proc format;
      value Genderfmt 0='male' 1='female';
run;
proc gplot data=rccsas.complete;
      plot GPA*HrsStudy=Gender;
      symbol1 value=plus i=r width=3 ci=green cv=black; /*for female*/
      symbol2 value=diamond i=r width=3 ci=red cv=blue; /*for male*/
      title 'scatter plot of GPA by Hours of Study for male and
female';    /*the type of the point is dot*/;
      format Gender Genderfmt.;
       run;
```



From the log window:

```
NOTE: Regression equation :  GPA(Gender:female) =  3.2724 + 0.001783*HrsStudy.
NOTE: Regression equation :  GPA(Gender:male) =  3.127944 + 0.003416*HrsStudy
```

iii. [7 minutes] Let's create a new dataset, in which you need to create a new binary variable called *shr:* if *HrsStudy>10*, then *shr='>10 hrs'*; otherwise, *shr='<=10 hrs'*. In this new dataset, you only need to keep the following variables: *GPA, HrsStudy, Gender, Shr*. Print the first 20 observations to check your new data.

```
data rccsas.complete_p5 (keep=GPA HrsStudy Gender Shr);
set rccsas.complete;
if HrsStudy > 10 then shr='>10hrs';
if HrsStudy <= 10 then shr='<=10hrs';
run;

proc print data= rccsas.complete_p5 (obs=20);
run;
```

Obs	Gender	GPA	Hrs Study	shr
1	male	2.90	12	>10hrs
2	male	3.20	10	<=10hr
3	female	3.40	10	<=10hr
4	female	3.50	9	<=10hr
5	male	3.47	15	>10hrs
6	male	3.47	15	>10hrs
7	female	3.41	12	>10hrs
8	female	3.89	15	>10hrs
9	female	3.25	10	<=10hr
10	male	3.37	20	>10hrs
11	female	4.00	20	>10hrs
12	female	3.50	8	<=10hr
13	male	3.40	0	<=10hr
14	male	3.12	10	<=10hr
15	female	3.10	20	>10hrs
16	female	3.74	6	<=10hr
17	female	2.86	12	>10hrs
18	female	4.00	16	>10hrs
19	male	3.30	3	<=10hr
20	male	2.50	6	<=10hr

iv. [5 minutes] Let's produce the two-way frequency table for the variable *shr* and *gender*, and perform chi-square test to test the relationship between *shr* and *gender* and draw your conclusion. Do the female students work harder than the male students?

```
/*problem iv:*/
PROC FREQ DATA = rccsas.complete_p5 ;
TABLES Gender*shr/chisq;
RUN;
```

The FREQ Procedure

Table of Gender by shr

Gender	shr		
Frequency Percent Row Pct Col Pct	<=10hr	>10hrs	Total
female	292 29.08 51.14 49.16	279 27.79 48.86 68.05	571 56.87
male	302 30.08 69.75 50.84	131 13.05 30.25 31.95	433 43.13
Total	594 59.16	410 40.84	1004 100.00

Statistics for Table of Gender by shr

Statistic	DF	Value	Prob
Chi-Square	1	35.2914	<.0001
Likelihood Ratio Chi-Square	1	35.7897	<.0001
Continuity Adj. Chi-Square	1	34.5254	<.0001
Mantel-Haenszel Chi-Square	1	35.2562	<.0001
Phi Coefficient		-0.1875	
Contingency Coefficient		0.1843	
Cramer's V		-0.1875	

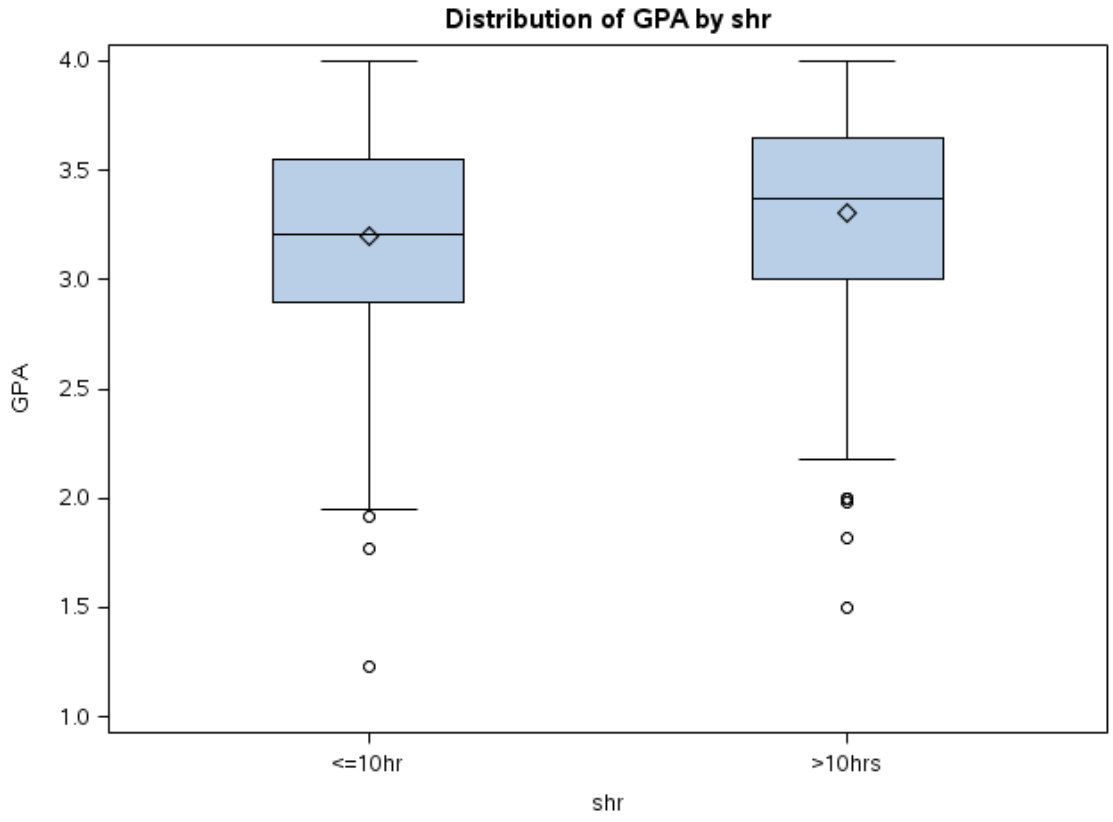
Fisher's Exact Test

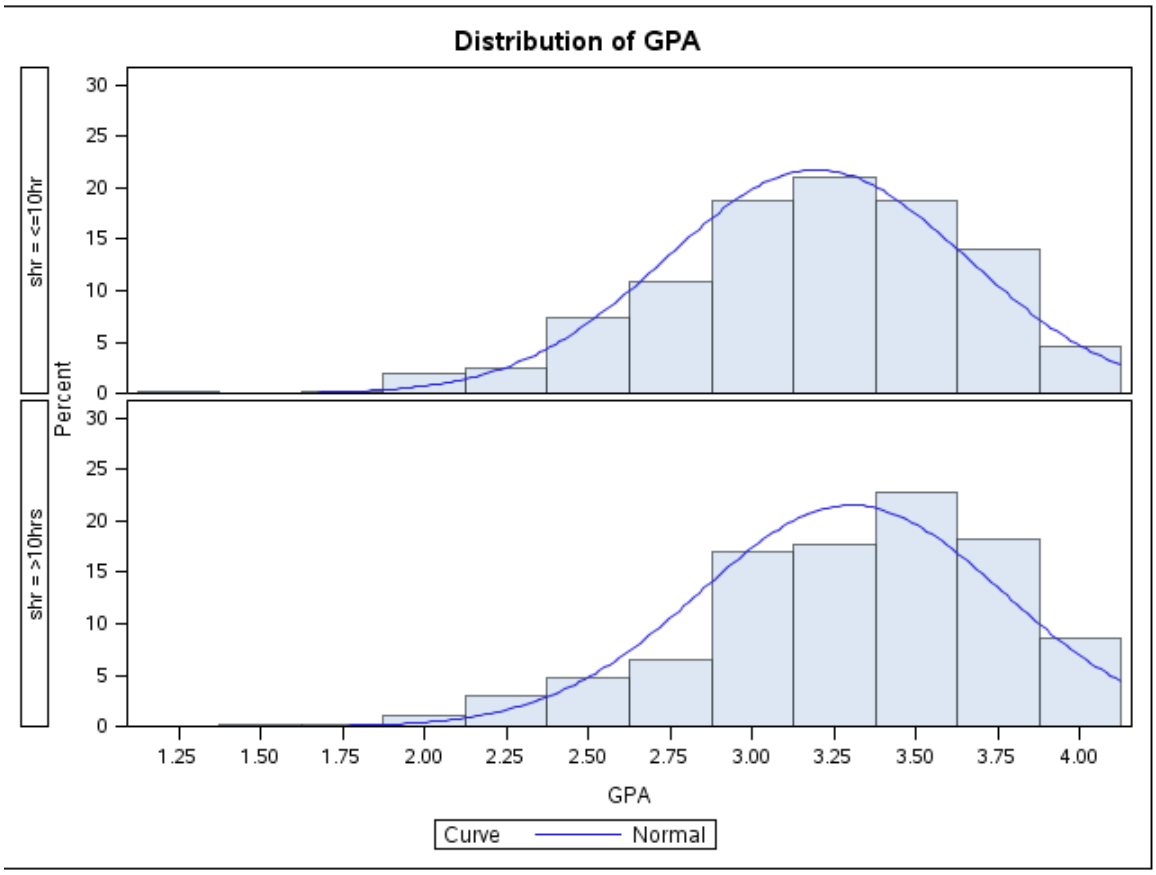
Cell (1,1) Frequency (F)	292
Left-sided Pr <= F	1.656E-09
Right-sided Pr >= F	1.0000

```
Table Probability (P)    9.199E-10
Two-sided Pr <= P        3.234E-09

        Sample Size = 1004
```

v. [6 minutes] Let's compute the summary statistics and histogram with the fitted normal curve of GPA for two groups defined by the variable *shr*. Plot the boxplot of GPA for two groups. Draw some conclusion based on the outputs.

```
PROC UNIVARIATE DATA = rccsas.complete_p5 NOPRINT ;
class shr;
histogram GPA / normal;
title "Descriptive Statistics";
RUN;


proc sort data =rccsas.complete_p5;
by shr;
run;
/*Boxplot for a subsets of data*/
proc boxplot data=rccsas.complete_p5;
plot GPA*shr/
boxstyle =schematic boxwidth=10;
run;
```




Distribution of GPA
shr = <=10hr
shr = >10hrs
Percent
30
25
20
15
10
5
0
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
3.25
3.50
3.75
4.00
GPA
Curve
Normal

vi. [5 minutes] Let's perform the two-sample t-Test to compare the means of GPA between these two groups defined by *shr*, and draw your conclusion. We always want to understand the output from any package. What is SAS suggesting regarding hours of study effect on GPA?

```
/* two-sample(independent) t-test (two-sided) */
proc ttest data=rccsas.complete_p5;
      class shr;
      var GPA;
      title 'two sample t-test for GPA';
run;
```

```
two sample t-test for GPA      21:00 Monday, July 22, 2013  35

                                The TTEST Procedure

                                  Variable:  GPA

          shr              N        Mean     Std Dev     Std Err     Minimum     Maximum

          <=10hr         590      3.1959      0.4579      0.0189      1.2300      4.0000
          >10hrs         405      3.3023      0.4628      0.0230      1.5000      4.0000
          Diff (1-2)            -0.1064      0.4599      0.0297

  shr           Method               Mean       95% CL Mean        Std Dev      95% CL
Std Dev

  <=10hr                           3.1959      3.1589   3.2329      0.4579      0.4332
0.4857
  >10hrs                           3.3023      3.2571   3.3475      0.4628      0.4329
0.4970
  Diff (1-2)    Pooled            -0.1064     -0.1646  -0.0481      0.4599      0.4405
0.4811
  Diff (1-2)    Satterthwaite     -0.1064     -0.1647  -0.0480

                  Method           Variances        DF    t Value    Pr > |t|

                  Pooled           Equal           993      -3.58      0.0004
                  Satterthwaite    Unequal      862.39      -3.58      0.0004

                                Equality of Variances

                       Method      Num DF    Den DF    F Value    Pr > F

                       Folded F       404       589       1.02    0.8136
```

Based on the results, the difference between the mean of GPA is not significant for the two studyhour group.

vii. [5 minutes] Let's perform the Wilcoxon Rank-Sum Test to compare the means of GPA between two groups defined by the variable *shr*, and draw your conclusion. Is the result consistent with what you concluded before?

```
/*4_vii*/
proc npar1way data=rccsas.complete wilcoxon ;
class shr;
var GPA;
title 'wilcoxon rank-sum test';
exact;
run;
```

```
wilcoxon rank-sum test       21:00 Monday, July 22, 2013  36

                              The NPAR1WAY Procedure

                    Wilcoxon Scores (Rank Sums) for Variable GPA
                             Classified by Variable shr

                                  Sum of      Expected       Std Dev          Mean
            shr          N        Scores      Under H0      Under H0         Score
            ------------------------------------------------------------------------
            >10hrs     405     218764.50      201690.0    4451.76328     540.159259
            <=10hr     590     276745.50      293820.0    4451.76328     469.060169

                         Average scores were used for ties.


                             Wilcoxon Two-Sample Test

                         Statistic             218764.5000

                         Normal Approximation
                         Z                          3.8353
                         One-Sided Pr >  Z          <.0001
                         Two-Sided Pr > |Z|         0.0001

                         t Approximation
                         One-Sided Pr >  Z          <.0001
                         Two-Sided Pr > |Z|         0.0001

                       Z includes a continuity correction of 0.5.


                              Kruskal-Wallis Test

                         Chi-Square                14.7107
                         DF                              1
                         Pr > Chi-Square            0.0001
```

[**LionX Job Submit]** Let's ask for one processors on one node for 30 minutes on LionXF machine. LionX machines are designed for heavy computations but here as a test bench example, we ask for the resources above to avoid further queue time. We want to submit a sample SAS script in batch mode and retrieve the results when the computation was finished in the directory of our interest.

```
# This is a sample PBS script. It will request 1 processor on 1 node
# for 30 minutes.
#
#   Request 1 processors on 1 node
#
#PBS -l nodes=1:ppn=1
#
#   Request 30 minutes of walltime
#
#PBS -l walltime=00:30:00
#
#   Request 1 gigabyte of memory per process
#
#PBS -l pmem=1gb
#
#   Request that regular output and terminal output go to the same file
#
#PBS -j oe
#
#   The following is the body of the script. By default,
#   PBS scripts execute in your home directory, not the
#   directory from which they were submitted. The following
#   line places you in the directory from which the job
#   was submitted.
#
cd $PBS_O_WORKDIR
#
#load module and then run SAS
module load sas
sas sas_batch.sas -nodate -linesize 90
#   The output file is only created if your program runs (has no
errors) and is named filename.lst.
```

Part II: Take-Home: US Regular Gas Price Trend

In this part, you'll work on a different dataset: Weekly U.S. Regular All Formulations Retail Gasoline Prices from 08/20/1990 to 07/11/2011, downloaded from U.S. Energy Information Administration.

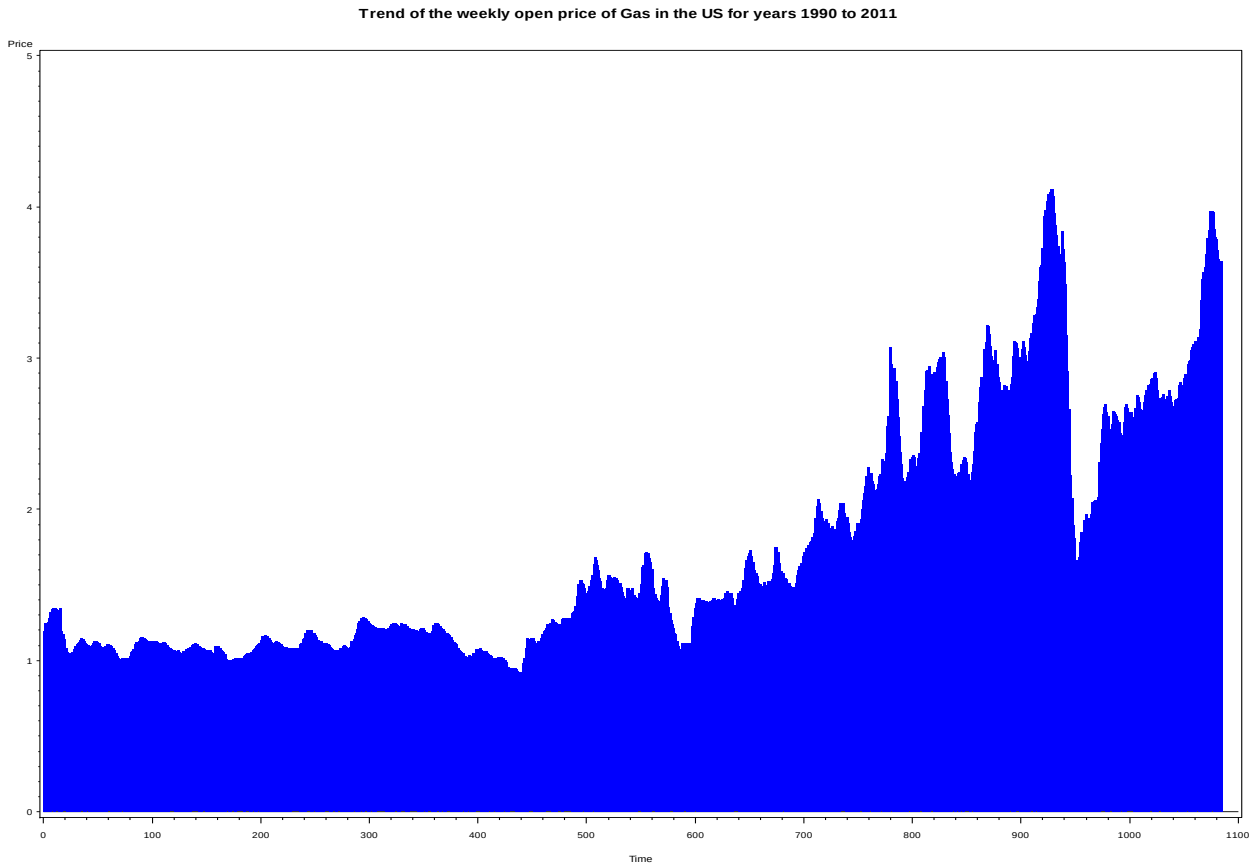
The working dataset is a .csv file called *gas.csv*, which you can download from the public link below. The variable *date* in the original dataset is t transformed into the variable *Time*, which presents the weeks where we consider 08/20/1990 as the first week. Therefore, in the *gas.csv* dataset, there are two variables Price (Weekly U.S. regular gas price) and Time.

(The original dataset link: http://www.eia.gov/dnav/pet/hist_xls/EMM_EPMR_PTE_NUS_DPGw.xls)

More Exercises:

5. [10 minutes] You need to read the data into SAS and plot the trend of the weekly U.S. regular gas price for the last 20 years. In detail, use PROC GPLOT to plot the scatter plot of Price over Time, specify that the type of point is none, the type of interpolation is needle, and the color is blue.

```
/*Part II*/

/* exercise five: weekly US regular Gas price from 1990 to 2011 in csv
format */
data usagas;
infile '\\winhpc\winfs\home\hup128\Downloads\gas.csv' delimiter
=',' firstobs=2; /* read the data into SAS */
input Price Time; /* Price (weekly price) and Time. */
run;
/*sort the data by time before plotting.) */
proc sort data= usagas;
by Time;
run;
/* plot the trend of the weekly open price of gas. */
proc gplot data=usagas;
plot Price*Time;
title 'Trend of the weekly open price of Gas in the US for years 1990
to 2011';

/* type of point is none, the type of
interpolation is needle, color of lines is blue. */
symbol value = none i=needle cv=red ci = blue width =3;
run;
```


Trend of the weekly open price of Gas in the US for years 1990 to 2011
Price
5
4
3
2
1
0
0
100
200
300
400
500
600
700
800
900
1000
1100
Time

6. [5 minutes] I designed the following SAS code to enhance the information and readability of the trend plot. The plot is given here! Please note the difference between the new plot and your plot in exercise 5. In this SAS code, there are some commands we didn't review in the workshop. Referring to the numbers in the code comments, write a brief description of each:

(1) int(argument) returns the integer value by truncation to avoid any floating point results. So here the price is divided by 0.5 and then truncated and the output in integers is saved in price_range.

(2) Axis<1...99><options>; so here the first axis assignment is performed (axis1) with the scale option of the values of 0 to 4.5 and 0.5 steps- that is like 0, 0.5,1.0,1.5,…,4.5 on the vertical here as we see. With the text option of Price($/gal) as its label.

(3) AUTOVREF option on the PLOT statement of PROC GPLOT is used to draw reference lines at all major tickmarks/grids.

(4) nolegend option after the '/' allows the user to suppress the legend.

(5) vaxis is the vertical axis option setting Axis1 as the vertical axis. Also, haxis is the horizontal axis option which sets the axis2 as the horizontal axis in the plot statement.

[More rope: Let's use Google to search for it, for example, search "SAS support PROC GPLOT + option name"; or you can use SAS help to check the syntax and options of PROC GPLOT, this always works! Happy SAS Coding!]

```
/** create a new data set called gas2, in which I added two new variables**/
data gas2;
  set gas;
  year=1990+36/52+Time/52;  /* decode week back to year */
  price_range=int(Price/.50); /* (1) */
run;
proc print data=gas2;
run;

goption reset=symbol;
symbol1 value=none interpol=needle color=cxFFFFCC;
symbol2 value=none interpol=needle color=cxFFEDA0;
symbol3 value=none interpol=needle color=cxFED976;
symbol4 value=none interpol=needle color=cxFEB24C;
symbol5 value=none interpol=needle color=cxFD8D3C;
symbol6 value=none interpol=needle color=cxFC4E2A;
symbol7 value=none interpol=needle color=cxE31A1C;
symbol8 value=none interpol=needle color=cxBD0026;
symbol9 value=none interpol=needle color=cx800026;
axis1 c=blue label=('Price($/gal)') order=(0 to 4.5 by .5)
offset=(0,0); /* (2) */
axis2 c=blue label=('Year')  value=(angle=90) order=(1991 to 2012 by 1)
offset=(0,0);


proc gplot data=gas2;
  plot Price*Year=price_range /
  autovref                    /*      (3)      */
```

```
  nolegend                          /*      (4)      */
  vaxis=axis1  haxis=axis2;         /*      (5)      */
  title 'US Regular Gasoline Price (avg weekly price per gallon)';
run;
```