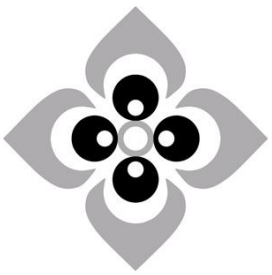


[Academic Script]

Introduction to SAS

Subject:	Business Economics
Course:	B. A. (Hons.), 5th Semester, Undergraduate
Paper No. & Title:	Paper – 502 Computational Techniques for Management
Unit No. & Title:	Unit – 3 Econometric Problem Solving
Lecture No. & Title:	Lecture – 2 Introduction to SAS

Academic Script

1. Introduction

SAS is a software suite that can mine, alter, manage and retrieve data from a variety of sources and perform statistical analysis on it. SAS provides a graphical point-and-click user interface for non-technical users and more advanced options through the SAS programming language. In order to use Statistical Analysis System, Data should be in an Excel table format or SAS format. SAS programs have a DATA step, which retrieves and manipulates data, usually creating a SAS data set, and a PROC step, which analyzes the data.

Having got the basic idea of what the software is let us move forward.

The topics that I will be covering in today's session are:

- Introduction to SAS through glancing at its website.
- Next we will be talking of Data files used in with the complicated extensions.
- Then we will go onto look at the libraries part where SAS files are stored.
- Next will be talking about output file where the output is saved.
- Next Log files i.e. files where commands are executed and if any error it is reported there.
- Further we will look into importing and exporting data using proc import & proc export commands.
- D
- We will also learn how to merge two files.

Now let us begin by looking at the website of SAS. You can simply google the name and get the website which is a very comprehensive site. It will give all details about the SAS program

that you are going to learn and also how to order the SAS program. Once you download the program on your computer you need to open the program and data file. Sometimes the program file is directly opened by double clicking on it and the other method is to right click the file and select, open with SAS.The Data files used in SAS have complicated extensions .sas7bdat, Excel files or CSV files that you can bring into the program.

Before starting to learn any features of SAS we need to make a note of few things.

- All the command starting with an asterisk * and ending with a semicolon will mean those lines will be ignored and not read by SAS.
- Also the second most important rule to be remembered is to end each procedure with a run statement.
- And lastly the way to execute the command is to highlight the lines and Click on submit or run and that would execute the program.

2. Creating a library where SAS data files are located

Now Let us start with creating a library in SAS. To create one you need to specify the library name you want to give besides the command libname, say lib1, then the location where the data files are located for eg. C drive, then the folder name let's say econometrics and lastly the file name, say Data. So the command to be entered would be:

```
* Creating a library where SAS data files are located;
libname lib1 'C:\Econometrics\Data';
run;
```

As I have already mentioned to execute any command you only have to highlight it and submit or run.

So once you do that you ll see where the files and data are located.

Using a sas file

As you can see I have already provided link for the data to be downloaded before you start the session. It contains data on Autos in three formats; SAS, EXCEL and CSV. The file name with SAS extension is intro_auto under the folder auto. So the command for the same will be:

```
* Using a sas file;
data auto;
set lib1.intro_auto;
run;
```

Note here it will look into the directory which we mentioned in previous command in C drive. After executing it will generate a data file in the work directory called auto(this is the name I gave). And as usual Run is how every procedure ends.

Importing a file

Now if your file is not a SAS file you don't need to worry if it is in either CSV or Excel format. For CSV format file name auto_csv we can import the CSV file into our SAS program through the command:

```
* Importing a csv file;
proc import out=auto_csv
datafile = "C:\Econometrics\Data\intro_auto.csv"
dbms=csv replace; getnames=yes; datarow=2;
run;
```

Similarly an Excel file can be imported through same proc import command with the only difference in the replacement where previously csv was mentioned and now Excel.

```
* Importing an Excel file;
```

```
proc import out=auto_excel
datafile = "C:\Econometrics\Data\intro_auto.xlsx"
dbms=excel replace; getnames=yes;
run;
```

Here the command used is Proc import with datafile mentioning the directory to be looked upon. If this is too complex to be remembered one can opt for the second method where you can directly click on File menu then under it import data and then select the data source format from the drop down menu which could be either excel or csv. And by clicking on Next Next... and finally ok you can get the commands and also save the records back into the program. Individually one needs to only change the directory that is the path has to be shown where the files are located.

Now that we are on Explorer window we can go one step above by clicking on the symbol for it on the bar. Double click on Work directory on the left pane and we can see all formats file have been brought inside the program.

The file in either three formats contains the following information on AUTOs

It talks about the makers of the model, the price, Mass per gallons, repairs needed, its weight and length and whether it is made in foreign or not shown by 1 and 0.

Make	Price	Mpg	Repairs	Weight	Length	Foreign
AMC	4099	22	3	2930	186	0
AMC	4749	17	3	3350	173	0
AMC	3799	22	3	2640	168	0
Audi	9690	17	5	2830	189	1
Audi	6295	23	3	2070	174	1
BMW	9735	25	4	2650	177	1
Buick	4816	20	3	3250	196	0
Buick	7827	15	4	4080	222	0
Buick	5788	18	3	3670	218	0
Buick	4453	26	3	2230	170	0
Buick	5189	20	3	3280	200	0
Buick	10372	16	3	3880	207	0
Buick	4082	19	3	3400	200	0
Cadillac	11385	14	3	4330	221	0
Cadillac	14500	14	2	3900	204	0
Cadillac	15906	21	3	4290	204	0
Chevrolet	3299	29	3	2110	163	0
Chevrolet	5705	16	4	3690	212	0
Chevrolet	4504	22	3	3180	193	0
Chevrolet	5104	22	2	3220	200	0
Chevrolet	3667	24	2	2750	179	0
Chevrolet	3955	19	3	3430	197	0
Datsun	6229	23	4	2370	170	1
Datsun	4589	35	5	2020	165	1
Datsun	5079	24	4	2280	170	1
Datsun	8129	21	4	2750	184	1

3. Print data in output window

Now proc print will print data in output window. One thing to be noted is it will be echoed in the log window once the code is executed. It shows every executed commands, number of observation read and also the most important part is it shows error, if any in red so as to distinguish easily. That's hence a good way to double check you program.

```
* Print data in output window;
proc print data=auto (obs=10);
run;
```

Here I have taken only 10 observation and the result that would be displayed would look like:

Obs	make	price	mpg	repairs	weight	length	foreign
1	AMC	4099	22	3	2930	186	0
2	AMC	4749	17	3	3350	173	0
3	AMC	3799	22	3	2640	168	0
4	Audi	9690	17	5	2830	189	1
5	Audi	6295	23	3	2070	174	1
6	BMW	9735	25	4	2650	177	1
7	Buick	4816	20	3	3250	196	0
8	Buick	7827	15	4	4080	222	0
9	Buick	5788	18	3	3670	218	0
10	Buick	4453	26	3	2230	170	0

Listing the variables

Next we can do is to list the variables. For that the command needed is:

```
* Listing the variables;
proc contents data=auto;
run;
```

This command when executed will show the output as shown on the screen.

The SAS System

The CONTENTS Procedure

Data Set Name	WORK.AUTO	Observations	26
Member Type	DATA	Variables	7
Engine	V9	Indexes	0
Created	Saturday, April 06, 2013 11:32:33 PM	Observation Length	64
Last Modified	Saturday, April 06, 2013 11:32:33 PM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type		Sorted	NO
Label			
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Engine/Host Information	Dependent
Data Set Page Size	8192
Number of Data Set Pages	1
First Data Page	1
Max Obs per Page	127
Obs in First Data Page	26
Number of Data Set Repairs	0
Filename	C:\...\auto.sas7bdat
Release Created	9.0301M0
Host Created	W32_7PR O

Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Format	Informat
7	foreign	Num	8	BEST12.	BEST32.
6	length	Num	8	BEST12.	BEST32.
1	make	Char	11	$11.	$11.
3	mpg	Num	8	BEST12.	BEST32.
2	price	Num	8	BEST12.	BEST32.
4	repairs	Num	8	BEST12.	BEST32.
5	weight	Num	8	BEST12.	BEST32.

Next is sorting the data. You can sort data through any of the variables. Here we have taken by make& price.

```
* Sorting the data;
proc sort data=auto;
by make price;
run;
```

To check whether the command is executed you can go in auto file and check it will be sorted as per the instruction given.

Descriptive statistics

Now you can also calculate descriptive statistics using SAS. For this the command to be used is Proc Means.

```
* Descriptive statistics;
proc means data=auto;
*class foreign;
run;
```

If you run this command the output seen would be:

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
Price	26	6651.73	3371.12	3299.00	15906.00
Mpg	26	20.9230769	4.7575042	14.00000	35.00000
Repairs	26	3.2692308	0.7775702	2.00000	5.00000
Weight	26	3099.23	695.0794089	2020.00	4330.00
Length	26	190.0769231	18.1701361	163.0000	222.0000
Foreign	26	0.2692308	0.4523443	0	1.0000

Detailed descriptive statistics

Next thing is detailed descriptive statistics using proc univariate with any variable like price or mass per gallon. In this you ll be able to have a lot more output here like the mean, the median, the mode and also all the percentas. It also gives knowledge about extreme observations i.e highest and the lowest to check the outliers too.

```
* Detailed descriptive statistics;
procunivariate data=auto;
var price mpg;
run;
```

The output would show:

The UNIVARIATE Procedure

Variable: price

Moments			
N	26	Sum Weights	26
Mean	6651.73077	Sum Observations	172945
Std Deviation	3371.11981	Variance	11364448.8
Skewness	1.470727	Kurtosis	1.5346717
Uncorrected SS	1434494797	Corrected SS	284111219
Coeff Variation	50.6803406	Std Error Mean	661.130988

Basic Statistical Measures			
Location		**Variability**	
Mean	6651.731	Std Deviation	3371
Median	5146.500	Variance	11364449
Mode		Range	12607
		Interquartile Range	3676

Tests for Location: Mu0=0

	Statistic			p Value
Student's	t	10.06114	Pr > |t|	<.0001
	M	13	Pr >= |M|	<.0001
Signed Rank	S	175.5	Pr >= |S|	<.0001

Quantiles (Definition 5)

Quantile		Estimate
100% Max		15906.0
99%		15906.0
95%		14500.0
90%		11385.0
75%	Q3	8129.0
50%	Median	5146.5
25%	Q1	4453.0
10%		3799.0
5%		3667.0
1%		3299.0
0% Min		3299.0

Merging two files

Lets us say we now wish to merge two files called auto and stats which have already been created in a new data file name auto2. The common variable make is used and the command given are:

```
* Merging two files - "make" is the common variable;
proc sort data=auto; by make; run;
proc sort data=stats; by make; run;
data auto2;
merge auto stats;
by make;
run;
```

Exporting a file

Lastly just as we talked about importing the various format files into SAS program now we will be talking about exporting them either as SAS file or CSV file or Excel file.

So the command for each of them would be:

```
* Exporting a SAS file as a sas file;
data lib1.auto1;
set auto1;
run;
* Exporting a SAS file as a csv file;
proc export data= auto1
outfile = "C:\Econometrics\Data\auto1.csv"
dbms=csv replace; putnames=yes;
run;
* Exporting a SAS file as an Excel file;
proc export data=auto1
outfile= "C:\Econometrics\Data\auto1.xls"
dbms=excel replace;
run;
```

And just like import data if you forget the command for this export too, you can directly go to the file menu and choose the option export data and select the format you wish to choose to save it as.

Frequency distribution for different variables

We can carry out frequency distribution analysis for various variables by using proc frequency. So the command for that for let us say variables make, foreign and repairs signifying the frequency of repairs needed for foreign auto made by particular company would be:

```
* Frequency distribution for different variables;
proc freq data=auto;
tables make foreign repairs;
run;
```

The output of this command would be three different tables each for make foreign and repairs showing normal frequency, percentage, cumulative frequency and cumulative percentage.

make	Frequency	Percent	Cumulative Frequency	Cumulative Percent
AMC	3	11.54	3	11.54
Audi	2	7.69	5	19.23
BMW	1	3.85	6	23.08
Buick	7	26.92	13	50.00
Cadillac	3	11.54	16	61.54
Chevrolet	6	23.08	22	84.62
Datsun	4	15.38	26	100.00

foreign	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	19	73.08	19	73.08
1	7	26.92	26	100.00

repairs	Frequency	Percent	Cumulative Frequency	Cumulative Percent
2	3	11.54	3	11.54
3	15	57.69	18	69.23
4	6	23.08	24	92.31
5	2	7.69	26	100.00

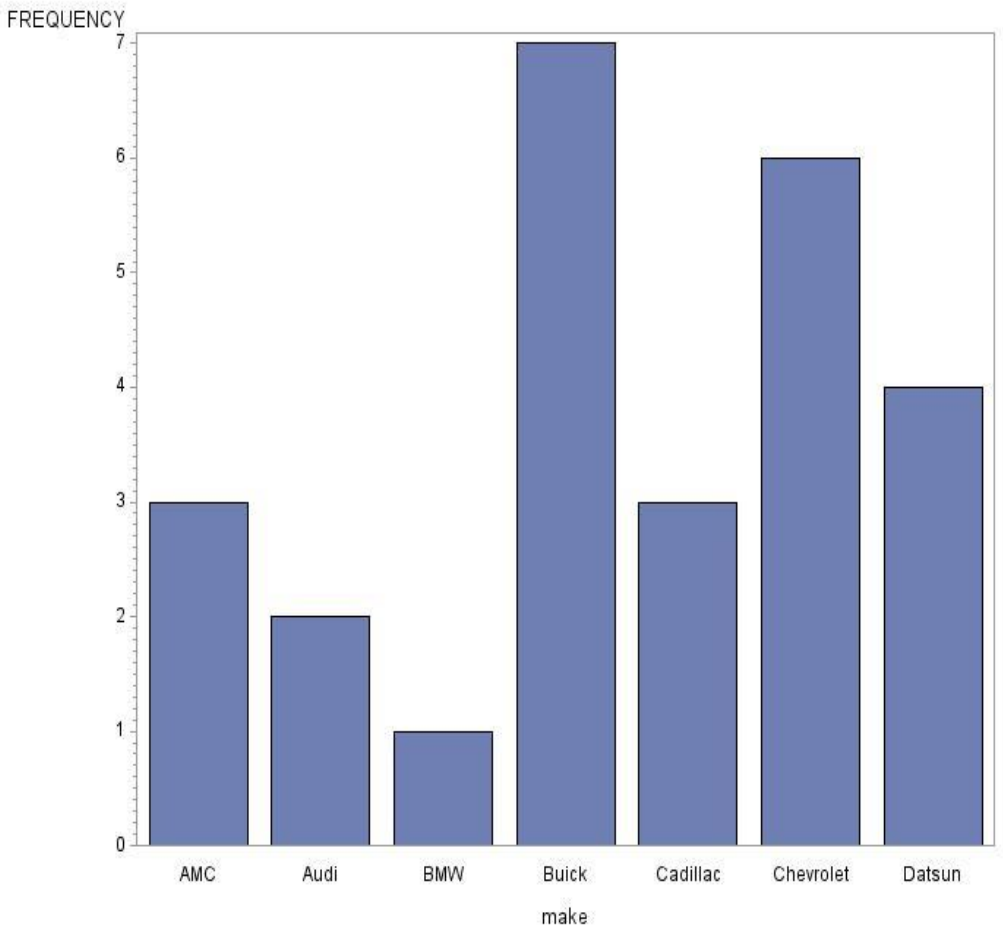
This is much more helpful when you have categorical data and mean wouldn't be appropriate to use.

4. Bar chart

To simplify the frequency analysis numerically we can also have bar chart graphical representation using Proc gchart in SAS.

For vertical bars Vbar command has to be mentioned and because it's a discrete variable that too will be mentioned. Hence the command would be:

```
* Bar chart;
procgchart data=auto;
vbar make/ discrete;
run;
```



5. Correlations

Next we will calculate correlations between various variables using Proc corr and mentioning the variable to be correlated.

Here we are building relation between the price of the auto with its mass per gallon, weight and length. So the command for the same would be:

```
* Correlations;
proccorr data=auto;
var price mpg weight length;
run;
```

The tables shown would be:

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
Price	26	6652	3371	172945	3299	15906
Mpg	26	20.92308	4.75750	544.00000	14.00000	35.00000
weight	26	3099	695.07941	80580	2020	4330
length	26	190.07692	18.17014	4942	163.00000	222.00000

Pearson Correlation Coefficients, N = 26

Prob> |r| under H0: Rho=0

	price	mpg	weight	length
price	1.00000	-0.43846	0.55607	0.43604
		0.0251	0.0032	0.0260

Pearson Correlation Coefficients, N = 26

Prob> |r| under H0: Rho=0

	price	mpg	weight	length
Mpg	-0.43846	1.00000	-0.80816	-0.76805
	0.0251		<.0001	<.0001
Weight	0.55607	-0.80816	1.00000	0.90654
	0.0032	<.0001		<.0001
Length	0.43604	-0.76805	0.90654	1.00000
	0.0260	<.0001	<.0001	

Here as we can see in the tables correlation with oneself will yield 1. Here you can see negative correlation between price and mass per gallon and a positive correlation between price and weight.

We can also correlate the data by group. For that you need to sort the data first and then use by statement. So here we will first sort the data in auto file by foreign category i.e only those auto made in foreign would be considered and then internal correlation would be built between the price of each unit with its mass per gallon, weight and length. Hence the command would be:

```
* Correlations by group - need to sort first and then use "by" statement;
proc sort data=auto; by foreign; run;
proccorr data=auto;
var price mpg weight length;
by foreign;
run;
```

The output would be showing the group correlation as shown on the screen:

The CORR Procedure

Foreign=0

4 Variables: price mpg weight length

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
Price	19	6484	3768	123199	3299	15906
Mpg	19	19.78947	4.03566	376.00000	14.00000	29.00000
Weight	19	3348	627.17691	63610	2110	4330
Length	19	195.42105	17.96390	3713	163.00000	222.00000

Pearson Correlation Coefficients, N = 19

Prob> |r| under H0: Rho=0

	price	mpg	weight	length
price	1.00000	-0.52852 0.0200	0.74972 0.0002	0.52504 0.0210
mpg	-0.52852 0.0200	1.00000	-0.86236 <.0001	-0.77040 0.0001
weight	0.74972 0.0002	-0.86236 <.0001	1.00000	0.87771 <.0001
length	0.52504 0.0210	-0.77040 0.0001	0.87771 <.0001	1.00000

The CORR Procedure

Foreign=1

4 Variables: price mpg weight length

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
Price	7	7107	2102	49746	4589	9735
Mpg	7	24.00000	5.50757	168.0000	17.00000	35.00000
Weight	7	2424	325.15930	16970	2020	2830
Length	7	175.5714	8.46280	1229	165.0000	189.00000

Pearson Correlation Coefficients, N = 7

Prob> |r| under H0: Rho=0

	price	mpg	weight	length
price	1.00000	-0.64108	0.88279	0.85397
		0.1207	0.0085	0.0144
mpg	-0.64108	1.00000	-0.71010	-0.81171
	0.1207		0.0738	0.0266
weight	0.88279	-0.71010	1.00000	0.87537
	0.0085	0.0738		0.0098
Length	0.85397	-0.81171	0.87537	1.00000
	0.0144	0.0266	0.0098	

Here as we can see on the screen that two different tables of output would be created. The correlation is calculated for two groups. The first one with foreign equals to zero and second one with foreign equals to 1

6. Summary

So friends let me summarize this session. Today we talked about the basic introduction to one of the most popularly used Software SAS in the big Corporates which can mine, alter, manage and retrieve data from a variety of sources and perform statistical analysis on it. We looked upon basic things to be performed on the software from creating library, exporting & importing data files to merging files and doing looking into descriptive statistics.

I hope it would have provided you with useful insights on Basics of SAS and will help in further understanding the software in detail. Thank You.