Introduction to SAS

Updated by: Tina Tian

Center for Social Science Computation & Research
110 Savery Hall
University of Washington
Seattle WA 98195

(206)543-8110
April 2013

http://julius.csscr.washington.edu/pdf/sas.pdf

Introduction to SAS for Windows Version 9

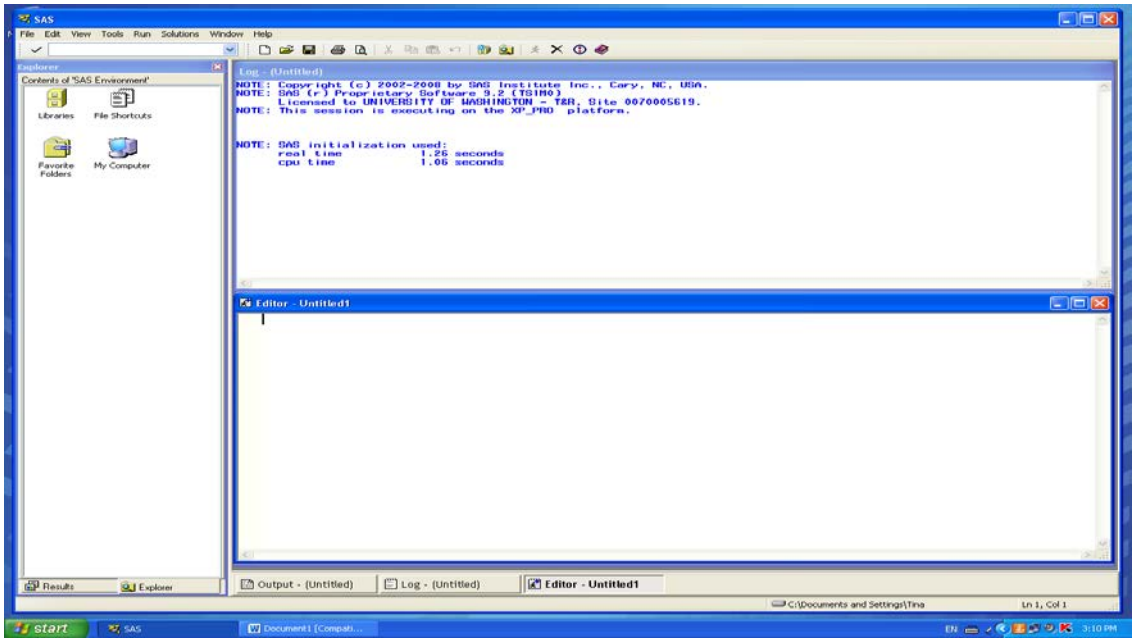
This document exposes beginners to basic features of SAS Version 9 on the Windows platform. Part One of this handout emphasizes command-driven rather than mouse-driven execution. Part Two reviews running SAS using built-in routines.

Part One

Although the command-driven version appears to have a higher learning curve, it does offer several advantages. First, operations are trackable because syntax files keep all the commands that have been executed. If one uses mostly mouse-click to do an analysis, one can easily lose the connections between output and commands, especially after a long period of time. Second, the syntax is great for repetitious work. You just change a few variable names or use different data and essentially run the same syntax. It also leads you to some great features of SAS, such as, arrays and SAS macros

SAS environment

To open SAS, click on the SAS icon on the desktop or go to:
Start > Programs > The SAS System > The SAS System for Windows V9



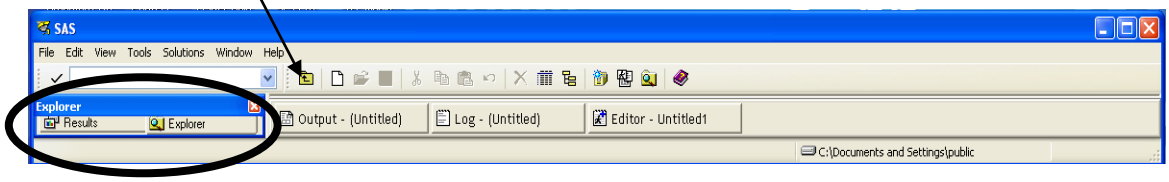
SAS Program Structure

SAS will open three windows, **Explorer, Log,** and **Editor.**

Explorer.

Explorer is for displaying data sets and **Results** is, naturally, for displaying result titles (detailed results are displayed in output windows). Both Explorer and Results have tree structures so you can click on items to show details. If you want to go up a level (a little tricky for first time users), find the first icon in the standard menu bar with up arrow in a folder (It says 'Up One Level' if you put your mouse over it)

Up one level



Log.

The **log window** is where SAS tells you what it is doing. It echoes your instructions and displays errors, warnings, and informational messages.

Editor.

Editor is where you write and execute the commands. The SAS Enhanced Editor has nice features such as color coding and separation lines. The commands you type will not be executed until you submit them to SAS.

After you submit your statements, a new window will become active. The **Output** window displays the statistical output (text, table or chart) you requested. Only one window is active at a time. You can move around the windows by using the Window menu, View menu or the task bar at the bottom.

SAS statement styles

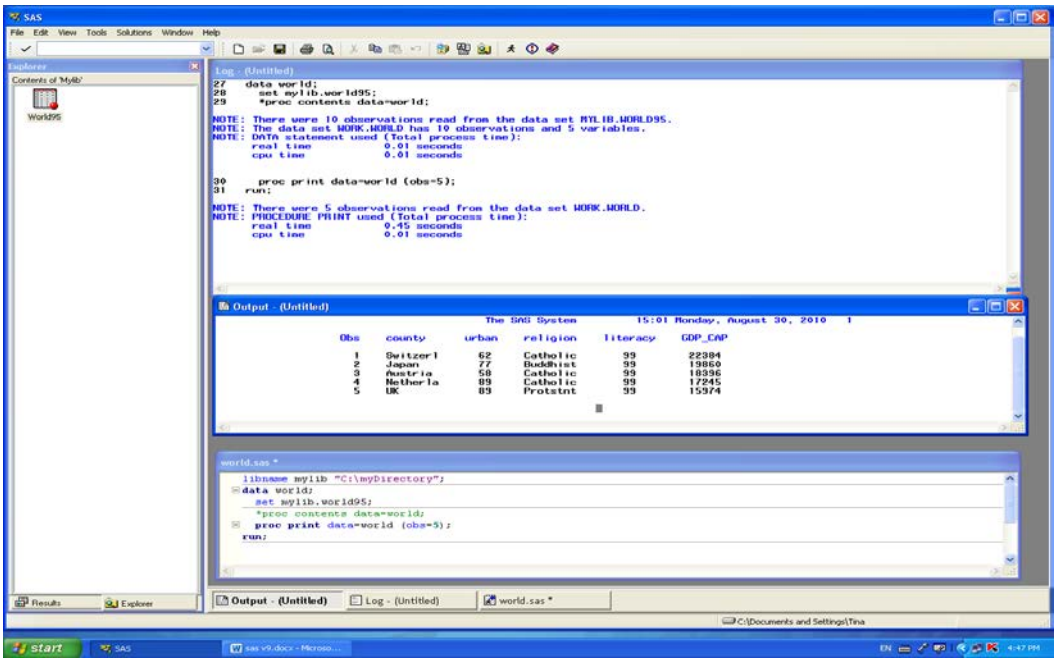
A SAS program consists of a number of **statements**. A statement is a line of text ended with a semi-colon. Typically, one statement takes one line but you can choose to put several statements in one line. A SAS statement is not case sensitive; data1 is the same as Data1. Often you want to put comments in SAS to facilitate understanding of statements.

That can be done by using asterisks: *this is one type of comment* or slashes and asterisks: /*this is another type of comment*/.

SAS statements can be grouped into two broad categories called steps. They are **data steps** and **proc steps.** Data steps deal with data manipulation and transformation, such as entering data, sorting and merging data sets, and recoding variables. In comparison, proc steps work on SAS data sets to produce desired statistical results, such as creating a statistical summary of variables, making charts and running regression analysis. There are twenty or so popular SAS procedures.

The **program editor** is the place where you work most of the time (see the figure below). You can

type new statements in the program editor. There are several ways that you can run the program. You can click on **Run** on the menu bar and choose **Submit**. Or you can click on the icon with a person running. Or you can press F3 or F8. Use your mouse to highlight the SAS statements you want to run, otherwise SAS will run all the statements every time.



Input Data

The method of data input in SAS depends on the types of data. Different statements are available for reading in raw data online or externally, importing data from SPSS or Excel, converting between Version 6 and Version 9, and randomly generating data.

Use the **cards (datalines**) statement to create data from scratch within the SAS editor. The following statements read internal data into a SAS session:

These are data steps:

```
data grade; /*grade is SAS data set name*/
input name$ grade;/*name is character variable so use $8*/
cards;/*cards (or datalines) specify inline data*/
Ben 3.6 /*following are actual records*/
Mike 2.8
Susan 3.1
Jessica 4.0
run; /*signifies the end of data step and asks SAS to process it*/
```

Use the **infile** statement to read raw data from an external text file.

```
data grade2;
infile 'C:\myDirectory\grade.dat'; /*directory and file name*/
```

```
input name$ grade;
run;
```

Use the **libname SPSS** statement to read in SPSS data. Save your SPSS data in portable format (.por instead of.sav) since SAS cannot read a.sav file. In SPSS, click on Save As and choose SPSS portable (.por). **libname** specifies mylib as the SPSS library engine and ties it to the physical location of grade.por

```
libname mylib SPSS 'C:\myDirectory\grade.por';
data grade3;
set mylib.grade;
run;
```

Use **proc import** to read in an Excel file. Provide a name for your SAS data by using the command out=dataName;

```
proc import datafile='C:\myDirectory\grade.xls' out=grade4 replace;
 run;
```

To convert SAS Version 6 files to SAS Version 9 files using engines:

```
libname mylib v6 'C:\myDirectory';
data grade5; /*new version 8 data8/
set mylib.grade; /*version 6 data*/
run;
```

Randomly generate test data by **ranuni**, which means randomly generated uniform data.

```
data grade;
set mylib.grade;
ran_grade=round(ranuni(4)*4);
run;
```

Data Manipulation

Now you have data in the SAS program. First look at the data to make sure that the data input is correct. Then you can add new variables, sort the data, recode data, or merge data sets. For the rest of Part One, we will use randomly sampled data from the world95 data set in SPSS. The data has twenty cases and several variables, such as country name (character variable), religion (character), urbanization (numeric), literacy, and GDP per capita.

SAS Procedures for getting general information

There are several SAS procedures to get general information. The procedure **proc contents** displays information about file location and time, data engine, and a complete list of variables in the data set. It is very useful for large data sets since it is difficult to browse in the data window.

```
libname mylib 'C:\myDirectory'; /*specify the location of world95 data*/
data world; /*new data in default WORK directory*/
set mylib.world95; /* old data name*/
```

```
proc contents data=world; /*proc contents much act some data*/
run;
```

If you want to make the data permanent, add a library name in front of the new data name, like this:

```
libname mylib 'C:\myDirectory';
data mylib.new_world; *data new world is created in my library*
set mylib.world95;
run;
```

SAS does allow you to open the data in the table format that you are probably familiar with in SPSS or Excel. Simply double click mylib (or work if you didn't specify 'library' in the data step). Double-click on the data name; a view table should open up for you to browse.

If you want to see the first few cases of the data, you can use the procedure **proc print** to do it:

```
libname mylib C:\myDirectory';
data world;
set mylib.world95;
proc print data=world (obs=5);
run;
```

Procedures for data manipulation

Say you want to sort your world data by gdp_cap (GDP per capita). Invoke **proc sort** by asking SAS to put the sorted data in a new data set. To sort gdp_cap in descending order, use these statements:

```
proc sort data=world out=world_sorted; *out creates new data*
by descending gdp_cap; *ascending is the default*
proc print data=world_sorted; *(check the new data whether it worked*
run;
```

The output window displays following results.

The SAS System output:

Obs	COUNTRY	URBAN	RELIGION	LITERACY	GDP_CAP
1	Switzerland	62	Catholic	99	22384
2	Japan	77	Buddhist	99	19860
3	Austria	58	Catholic	99	18396
4	Netherlands	89	Catholic	99	17245
5	UK	89	Protstnt	99	15974
6	U.Arab Em.	81	Muslim	68	14193
7	Spain	78	Catholic	95	13047
8	Lithuania	69	Catholic	99	6710
9	Estonia	72	Protstnt	99	6000
10	Croatia	51	Catholic	97	5487
11	Georgia	56	Orthodox	99	4500

12	Bosnia	36	Muslim	86	3098
13	Ukraine	67	Orthodox	97	2340
14	Somalia	24	Muslim	24	2126
15	Thailand	22	Buddhist	93	1800
16	Colombia	70	Catholic	87	1538
17	Morocco	46	Muslim	50	1062
18	Egypt	44	Muslim	48	748
19	Zambia	42	Protstnt	73	573
20	Ethiopia	12	Muslim	24	122

For recoding, you can use the **conditional statement** in the data step to change variables or create a new variable. For example, to recode the variable 'religion' into categorical variables:

```
Data world1;
Set world;
Relcat=.; /*create a new variable*/
If religion = 'Buddhist' then relcat = 1;
If religion = 'Catholic' then relcat = 2;
If religion = 'Muslim' then relcat = 3;
If religion = 'Orthodox' then relcat = 4;
If religion = 'Protstnt' then relcat = 5;
Run;
```

This way you create a new variable relcat (religion in categorical format) in the new data world1.

Statistical Analysis

Here are a few SAS procedures for doing simple statistics, such as descriptive statistics and linear regression. **Procedure means** is a convenient method to display descriptive statistics of numeric variables. In this example, we can have basic statistics for variables 'gdp_cap,' 'urbanization' and 'literacy.'

```
proc means data=world1;
var gdp_cap urban literacy;
run;
```

For more sophisticated information regarding distribution and graphic output, you can use

procedure univariate.

```
proc univariate data=world1;
var gdp_cap urban literacy;
run;
```

You can use **procedure freq** (frequency) to get basic information on categorical variables.

```
proc freq data=world1;
tables country religion; /*no comma between these variables*/
run;
```

or doing cross tabulations of categorical variables;

```
proc freq data=world1;
tables country*religion; /*star specifies crosstab of two variables*/
run;
```

The last common method is **regression**. You can use **procedure reg** (regression) to get the statistics. Here 'gdp_cap' is the dependent variable and 'urban' and 'literacy' are the independent variables. The output window will automatically give you all the basic statistics and results.

```
proc reg data=world1; /*reg refers to linear regression model*/
model gdp_cap = urban literacy; /* y = x1 b1 + x2 b2 + e */
run;
```

The SAS System output:

The REG Procedure
Model: MODEL1
Dependent Variable: GDP_CAP

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	483090738	241545369	6.87	0.0065
Error	17	597980486	35175323		
Corrected Total	19	1081071225			

Root MSE	5930.87875	R-Square	0.4469
Dependent Mean	7860.15000	Adj R-Sq	0.3818
Coeff Var	75.45503		

Parameter Estimates

Variable	DF	Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-6587.49806	4647.71340	-1.42	0.1744
URBAN	1	190.99865	80.36733	2.38	0.0295
LITERACY	1	42.99848	70.54843	0.61	0.5503

The model shows that only urbanization is statistically significant in explaining the differences of GDP per capita across the twenty countries.

SAS output can be saved as SAS list files or in Rich Text format (.rtf). The later can be easily incorporated into reports.

Help Resources

UCLA has a superb SAS help site. You can either browse by subjects or by using the search engine. I found it extremely helpful in finding almost any topic at hand which I don't know. The best part, particularly for beginners, is that you can find simple examples for statistical methods and SAS procedures. Highly recommended. The site more than compensates the unfriendliness of SAS documentation. Here is the link: http://www.ats.ucla.edu/stat/sas/

Another good resource is the SAS user conference proceedings. Every year, SAS convenes an international user conference. There are tutorials on various specific topics at different levels of difficulty. This is the place to find smart ways to run SAS programs.
Here is the link for the SUIG proceedings
http://support.sas.com/events/sasglobalforum/previous/online.html

A good book one for beginners is My Little SAS Book, written by Lora D. Delwiche and Susan J. Slaughter. It can be purchased online or in bookstore at a reasonable price.

Part Two

This section describes how to use the SAS system in the interactive and batch mode. There are two different routines within the interactive mode; display manager and SAS/ASSIST.

Two Interactive Modes

These two modes use the Windows graphical interface. To run SAS for Windows at CSSCR, double-click the SAS icon to launch SAS. You will get the SAS **program editor, log** and **output windows.**

In the **program editor window**, you can create or edit SAS codes. The **log window** displays state ments submitted to SAS from the program editor window with any notes or error messages. The **output window** shows the output from your SAS codes. These features are covered in Part One of this document

The following sections will use typical SAS statement files to show you how to apply these two different modes to get the same output.

Here is what you'll do in the exercise:

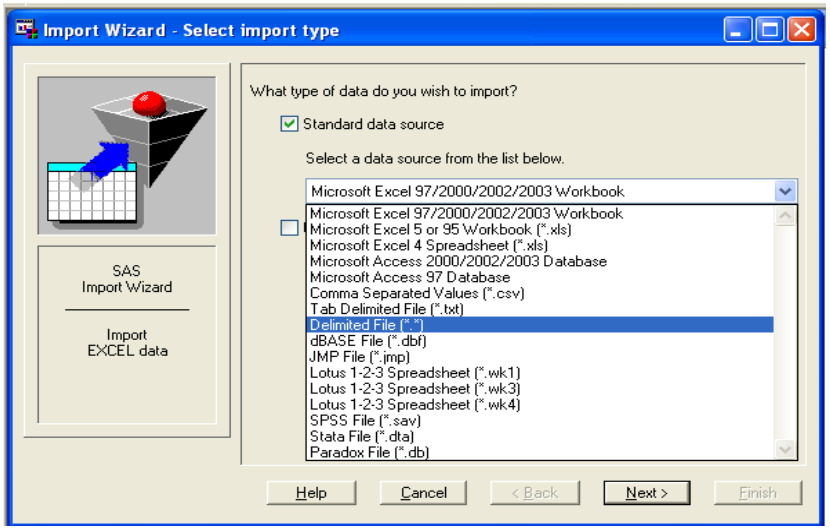
1. Import an ASCII data file, sample.dat that is on floppy disk. That file has six time series, M1, NBR, R, IP, and TIME for sample period Jan. 59-Sept. 95.
2. Draw a plot for the IP (industrial production index) to TIME, (time trend).
3. Simple time series regression of IP on M1.

4. (optional) Forecasting of P (consumer price index, 1987=100) using ARIMA (1,1,0).

First interactive mode – Display Manager Mode Using Menus

Task 1. Import a File

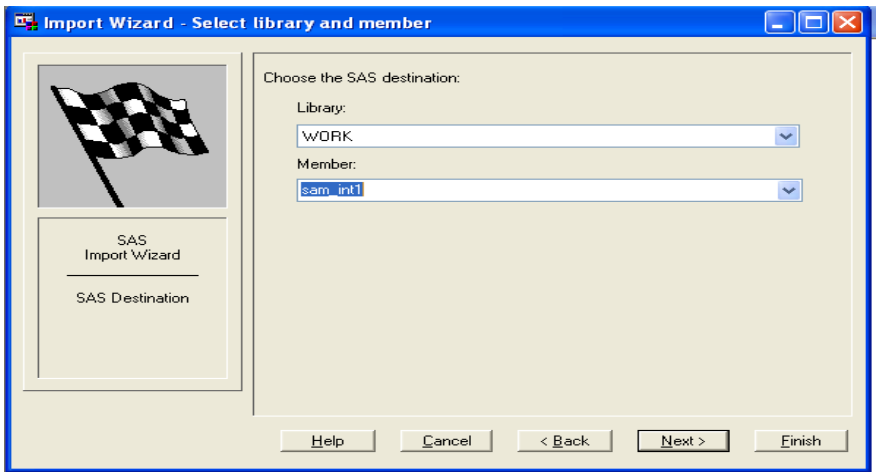
Go to the File menu and select Import. You will see the **Import Wizard** dialog box; choose the file format Delimited file.



Click on the **Next>** button to locate the file that you will import into SAS.

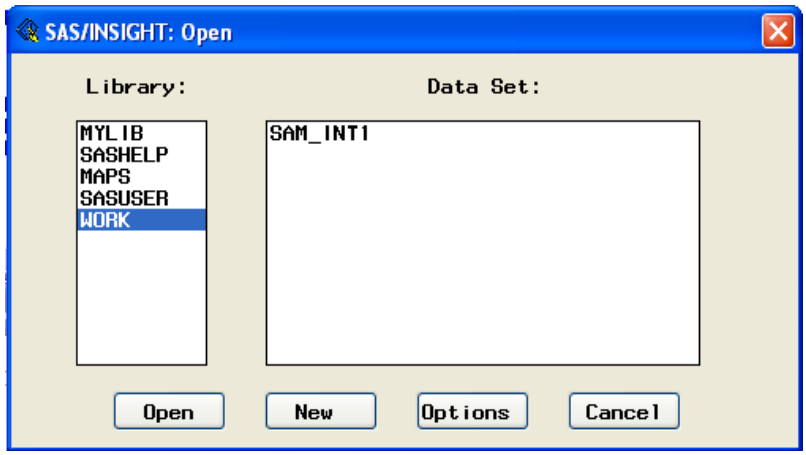
Because you know that the file 'sample.dat' is on disk in drive A, type in the box: a:\sample.dat.

Click the Next> button to get the Library and Member dialog box. When you use the SAS temporary data format, the library name is WORK and the MEMBER is the data set name. Name the data set 'sam_int1'. The 'sam_int1' data set will become a temporary file in the library WORK by clicking the Finish button.

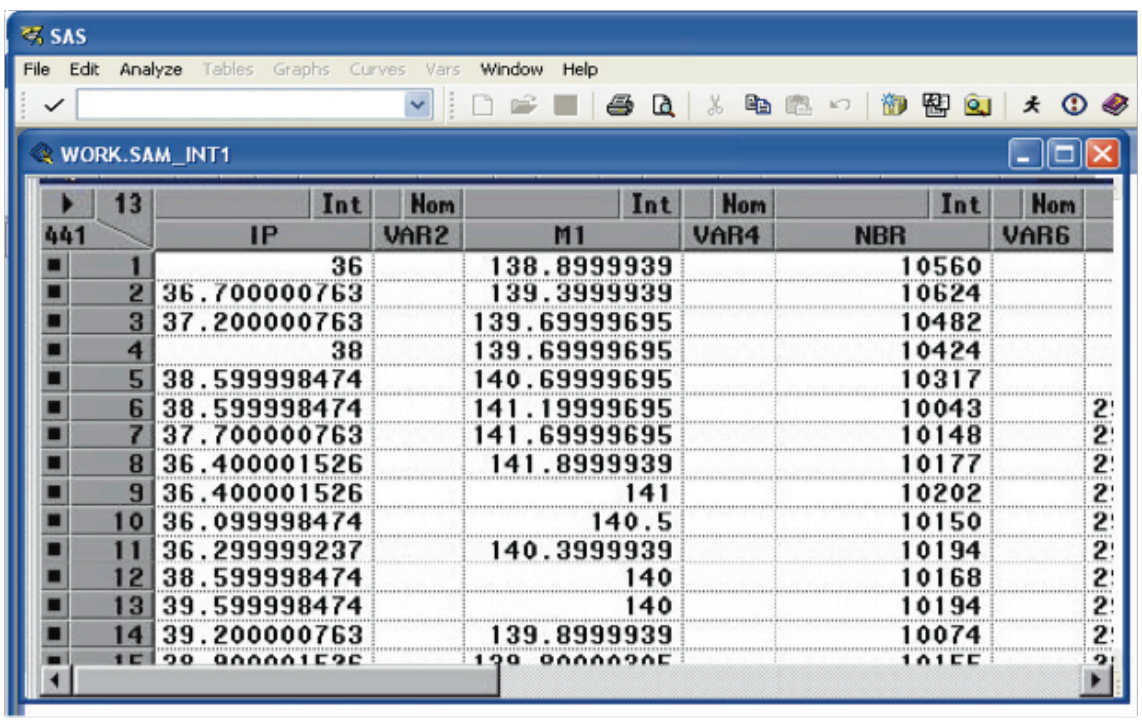


Task 2. Time Plot

To draw a **time plot** with the variables IP and TIME, go to Solutions on the menu bar, select Analysis and Interactive Data Analysis. SAS will automatically launch the SAS/INSIGHT software.

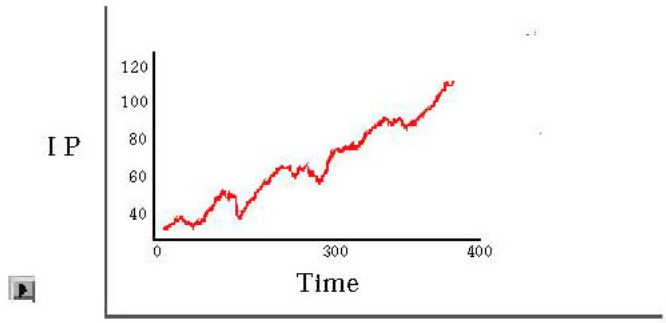


The dialog box above is where you open a data set to use SAS/INSIGHT. Our data set is 'work.sam_int1.' If you click the Open button, you will see the following window.



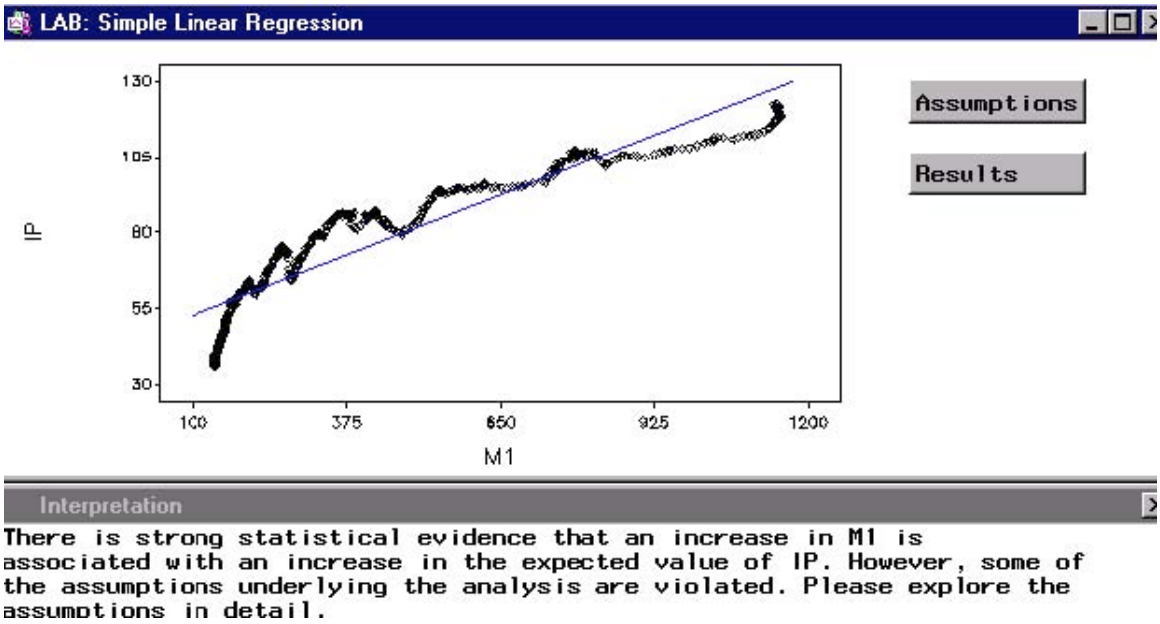
13	Int	Nom	Int	Nom	Int	Nom
441	IP	VAR2	M1	VAR4	NBR	VAR6
1	36		138.8999939		10560	
2	36.700000763		139.3999939		10624	
3	37.200000763		139.69999695		10482	
4	38		139.69999695		10424	
5	38.599998474		140.69999695		10317	
6	38.599998474		141.19999695		10043	
7	37.700000763		141.69999695		10148	
8	36.400001526		141.8999939		10177	
9	36.400001526		141		10202	
10	36.099998474		140.5		10150	
11	36.299999237		140.3999939		10194	
12	38.599998474		140		10168	
13	39.599998474		140		10194	
14	39.200000763		139.8999939		10074	

The active window is now work.sam_int1. Go to Analyze and select Line Plot (Y X). At the **Line Plot** dialog box, select IP in the variable list box and click the Y button. In the same way, click TIME and the X button. When you click on OK, you will have an IP-TIME plot like this.



Task 3. Regression Analysis

Simple regression can be done using the option **Guided Data Analysis**. At the menu, click Locals and select the simple linear regression option as Analysis in Change menu. SAS/LAB opens the Dependent and Predictor Variable dialog box. Select IP and M1 as your dependent and predictor variable respectively. Again, if you click on the Analyze button, you will have the following output.



The results from SAS/LAB contain an Interpretation of your analysis. There are more options in the **Assumption** and **Results** buttons.

Second interactive mode –SAS/ASSIST software

SAS/ASSIST is an environment for using the SAS System to perform everyday activities such as retrieving and storing data, manipulating and analyzing data writing reports and creating graphic displays. SAS/ASSIST automatically writes the necessary statements.

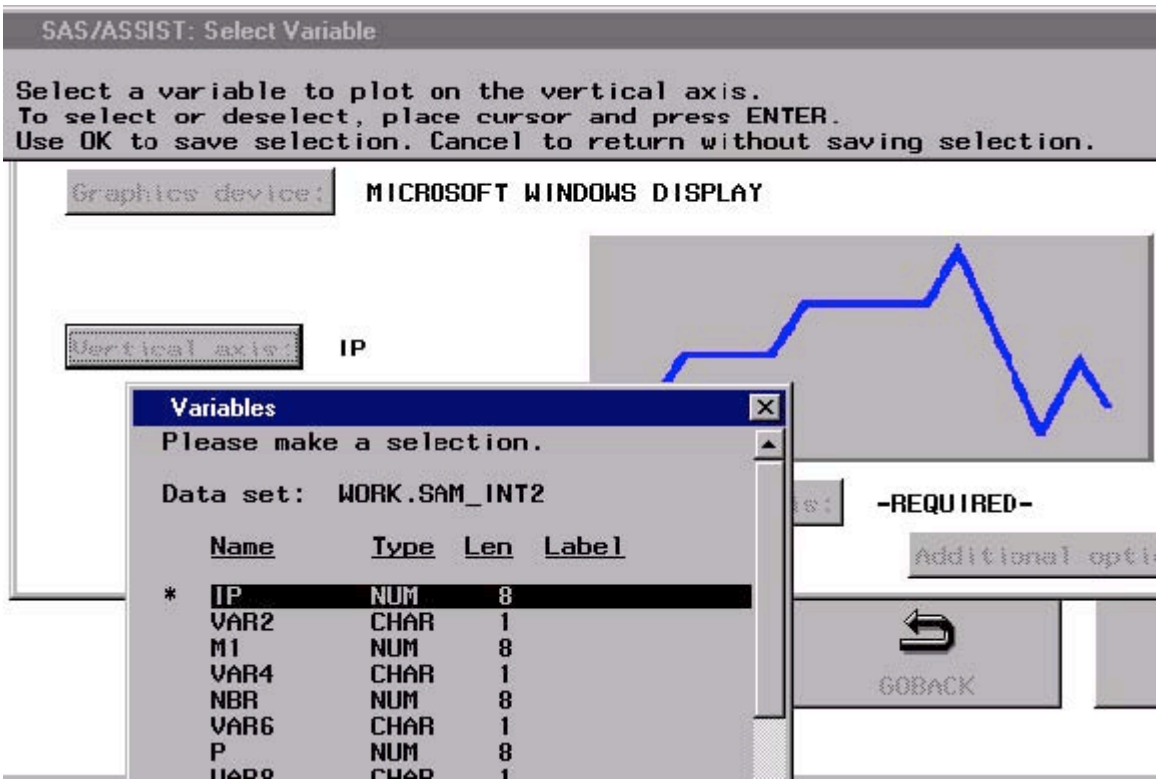
You can launch SAS/ASSIST by clicking the SAS/ASSIST button on the tool bar or go to Solutions and select ASSIST.

Task 1.

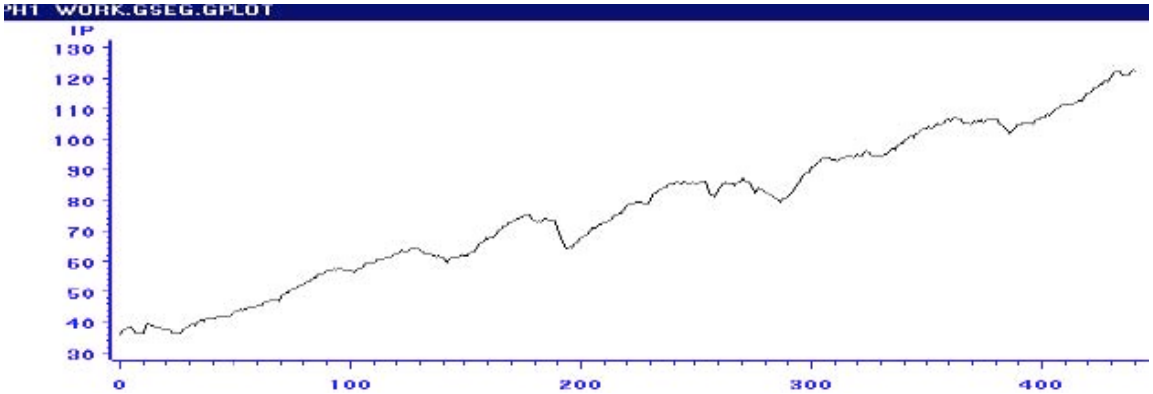
In the SAS/ASSIST windows, you need to click Date Mgmt. button to open the **Data Management Menu** window. To import an ASCII file from your floppy drive, click on the Create/Import button in that window. You will see the Create/Import menu. The easy way to import an ASCII file is to use the option Import data (beta). This selection brings you to the **Import Wizard**. Save the ASCII file as a SAS temporary data set 'work.sam_int2' If you are not sure of this procedure, go Task 1 of the previous section. You can check the contents of the 'sam_int2' data file by clicking the Edit/Browse button.

Task 2.

To create a **line graph** with SAS/ASSIST, click the Graphics button on the SAS/ASSIST: WorkPlace window. Select the High resolution option to get the High resolution graphics menu. For a line graph, click on the Plot button and simple X and Y plot… to see the simple x and y plot dialog box. You need to specify each variable which is marked -required- by clicking the buttons. The following figure shows how to select the Y-axis variable IP in the work.sam_int2 data file.



After you finish selecting the variables, click Local on the menu bar and select Run to plot your IP and TIME variables. Below is the output of this procedure.



Task 3.

To get regression results, go back to the SAS/ASSIST: WorkPlace window and click the Data analysis button. It brings you to the Data Analysis menu and you will see lots of buttons. Just select the Regression button and Linear regression. You need to specify the information in the Regression analysis dialog box by clicking on each. After you finish, go to Local on the menu and select Run. Then you see your regression output.

OUTPUT - (Untitled)

01:20 Friday, September 1

L1
ariable: IP

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Prob>F
Model	1	215589.07682	215589.07682	2782.600	0.0001
Error	439	34012.64854	77.47756		
C Total	440	249601.72535			

Root MSE	8.80213	R-square	0.8637
Dep Mean	76.96994	Adj R-sq	0.8634
C.V.	11.43581		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	T for H0: Parameter=0	Prob > |T|
INTERCEP	1	45.035372	0.73633125	61.162	0.0001
M1	1	0.072511	0.00137461	52.750	0.0001

Conclusion

This exercise focuses on running SAS in the interactive mode. We covered: importing an external file, plotting, and simple regression in SAS. Because many SAS analysis tools share the same procedures, following this exercise step by step will give you the foundation for conducting your own SAS sessions. Good Luck!!!!!!!