Social Science Research Lab
American University
Washington, D.C.
http://www.american.edu/ssrl
202.885.3862

SAS Basic

1 Introduction

SAS (Statistical Analysis System) is based on the use of an enhanced text editor for program and data input. Commands are entered and amended using the SAS editor and are stored in memory. The resulting job can then be saved and/or submitted for processing.

2 Default SAS Windows

The default image is shown in **Figure 1**:

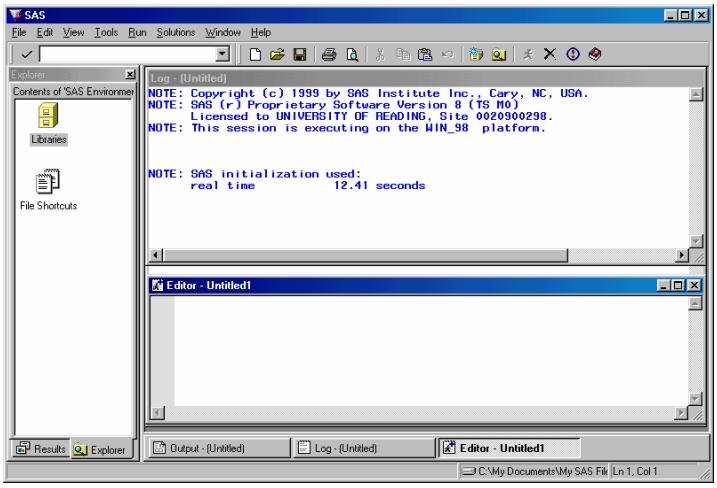

Figure 1

Initially the program and data are entered into the **Editor – Untitled1** window. A log of the statements submitted will appear in the **Log - (Untitled)** window, whilst any results created from the running of your program will appear in the **Output – (Untitled)** window. As you generate more and more outputs (both correct and incorrect) a summary list of the procedures called will appear in the **Results** window. The **Explorer** window allows you to set up access to previously defined datasets.

3 Notations

A SAS program consists of a sequence of one or more steps and each step comprises several SAS statements. There are two kinds of steps:

1. The data step is used to create and manipulate temporary and permanent SAS data sets and begins with the word **DATA**.
2. The procedure step is used for analysing or processing SAS data sets. Each procedure begins with the word **PROC** and ends with the **RUN;** statement or another **PROC** or **DATA** statement.

4 Data Entry

The data used to illustrate a simple use of the SAS package consist of three variables, a **FARM** number, the number of **TREES** on that farm, and the yield of **APPLES** (in unspecified units) from each of the farms. The data are used in the first few exercises to illustrate data entry and modification, saving and retrieving programs, descriptive statistics and X-Y scatter plots.

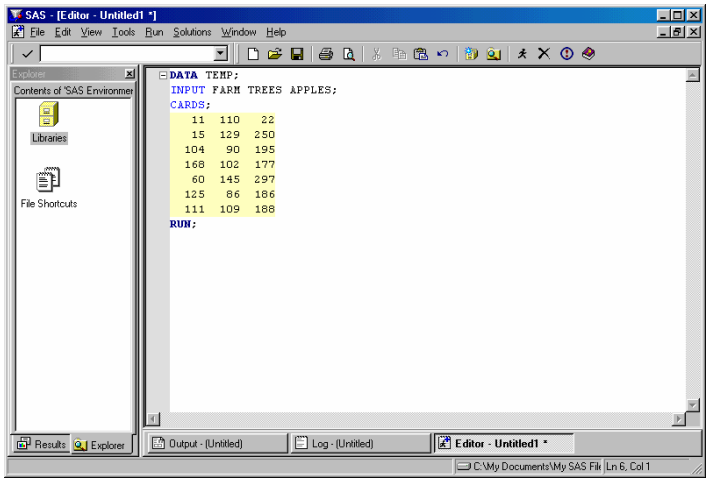

Figure 2

5 Saving Your Work

Although entering the statements in Exercise 1 did not take too long, it is always wise to save any work you have done. Whenever using a PC, saving your work should be a frequent operation.

Exercise 2

Click on the **Save** icon  on the menu bar (or use **File→Save As...** from the drop-down menu). A **Save As** dialogue box appears (as shown in Figure 3). If necessary, move to your home directory on Unix, which will be linked as the **N:** drive, then enter a relevant filename e.g. ***apples.sas*** . Click on **Save**.

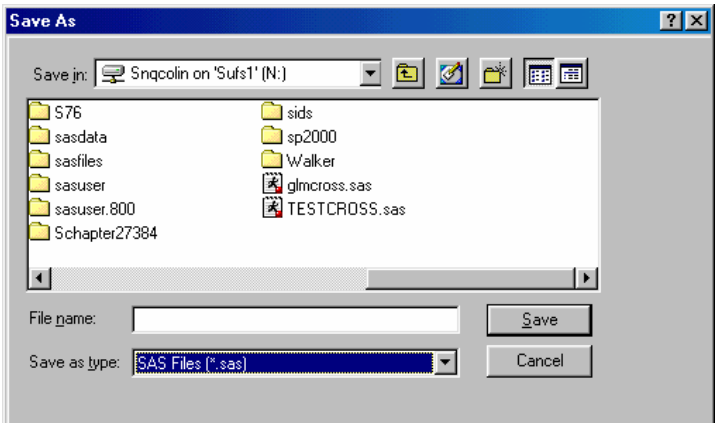

Figure 3

6 Submitting a Job

In order that SAS can process the commands that you have typed in, they have to be submitted.

Exercise 3

Click on the  button (**Submit**) on the main menu bar. Look at the **Log – (Untitled)** window (shown in Figure 4), and scan the contents looking for useful information (such as the number of observations, variables read in etc).
If you have followed the instructions correctly you will have created a SAS data set called **WORK.*TEMP*** with **7** observations and **3** variables.

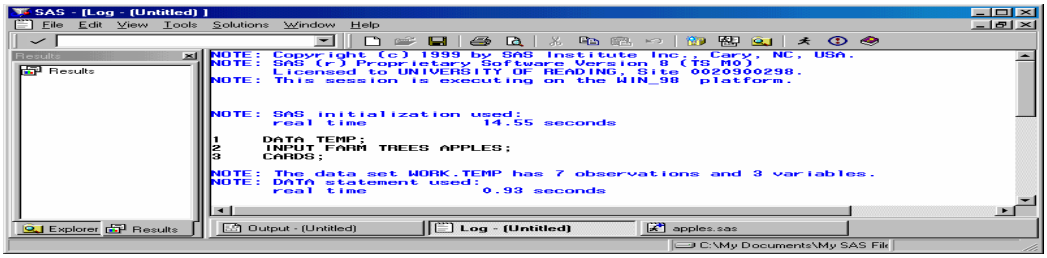

Figure 4

7 Correcting Errors

If there are error messages, they will appear in red in the **Log – (Untitled)** window, to make them obvious. The error message is displayed **after** the next line of programming as SAS attempts to process everything up to the next semicolon.

Exercise 4

To correct any errors that may have occurred, you need to return to the **Editor** window (now called ***apples.sas***) and correct any mistakes that you can see, re-save the program following the instructions in Exercise 2 and re-submit it. If you cannot see where you have made mistakes ask the lab assistance.

8 SAS Data set

SAS keeps the data on which it is working in a **data set**, similar to a spreadsheet of rows (observations) and columns (variables). A great amount of other information is, in fact, held as part of the data set but does not concern us here. You should always give a name to each data set created (such as ***TEMP*** in Exercise 1) and SAS will record it during the session as **WORK.*TEMP***. This allows you to access previously created datasets out of sequence. All temporary SAS data sets are deleted when you **Exit** from SAS. You can give it a two-part name (such as ***PERM.APPLES***) if you are saving it as a permanent data set. This data can then be referenced more easily in a later SAS session. Files that you have saved externally to the SAS system in Exercise 2 or Exercise 4 (i.e. on the **N:** drive) are retained until you send them to the **Recycle Bin**.

9 Listing the Data

There has been no output to consider up to now (there is nothing in the **Output – (Untitled)** window). To list the data onto the screen a procedure **PRINT** must be run.

Exercise 5

Move to the end of the previous commands in the ***apples.sas*** window and type

```
PROC PRINT DATA=TEMP;
RUN;
```

Highlight only these commands and submit them to run.

If you have been successful, your screen should now be similar to Figure 5. If nothing appears to have changed look at the **Log - (Untitled)** window and see if you can locate the error as described in Section 7. Correct the error in your program, if possible, as shown in **Exercise** and re-run. Ask the lab assistant for help if needed.

You may notice output has become lost from view on the right-hand side or from the bottom of the window. This can be rectified, by redefining the output window line width (**LS**) to be 75 characters and the output window depth (**PS**) to be 30 lines. Repeat this exercise by adding

```
OPTIONS LS=75 PS=30;
```

In front of the procedure call and resubmitting the three lines.

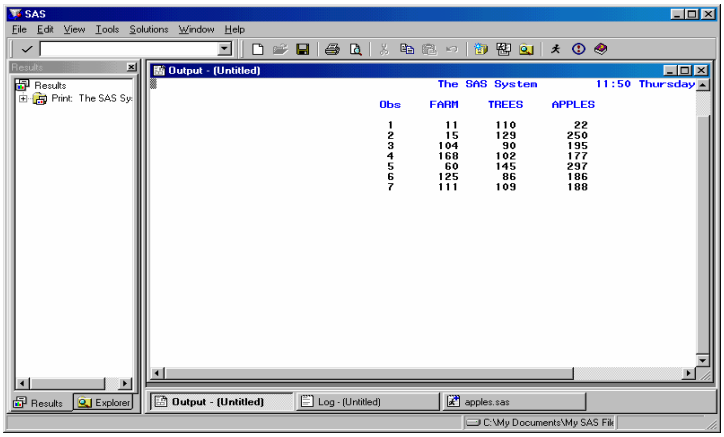

Figure 5

10 Checking Data

There are several techniques for helping you to check your data. Checking data means and plotting appropriate pairs of variables quickly give you an image of the spread of the data. Both of these checks can be made using SAS procedures as shown in here.

As before, move back to the ***apples.sas*** window and add the following lines at the end.

```
PROC MEANS DATA=TEMP;
RUN;
PROC PLOT DATA=TEMP;
```

PLOT *APPLES*TREES*;
RUN;
Highlight only these statements and submit the job. Look at the results that have appeared in the **Output - (Untitled)** window. There should be two new pages of output as shown in Figures 6 & 7 (use the scroll bar to see both of them).
The means of all numeric variables that exist in the data set ***TEMP*** are calculated, unless a **VAR** statement specifying the required numeric variables is used before **RUN;**. Return to the ***apples**** window and insert the appropriate statement as shown here.
PROC MEANS DATA=*TEMP*;
VAR *APPLES;*
RUN;
Submit just this procedure and check that the new output contains only the mean of the variable ***APPLES.***

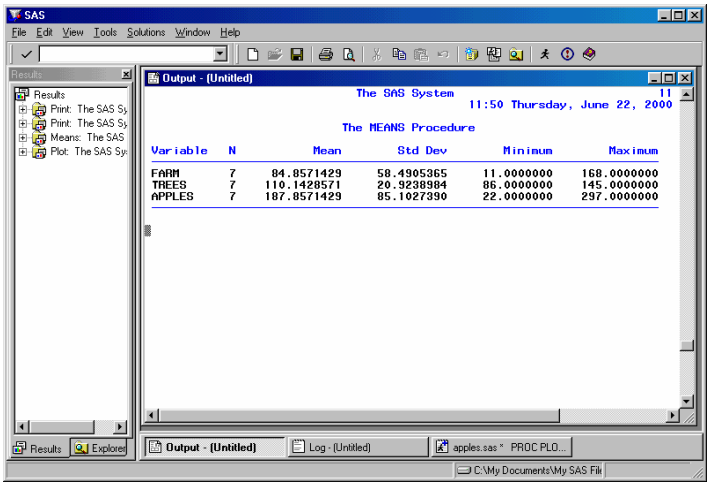

Figure 6

When regression analysis is being performed it is usual to want to save the predicted (or fitted) values and residuals and to plot them against each other to check that the residuals are randomly scattered about zero. An **OUTPUT** statement allows you to save these values while the **PLOT** statement can be used for simple plots. Try the following program and see if you can understand what is happening. Ask the tutor if you are not sure.
PROC REG DATA=*REGRESS*;
MODEL *Y=X*;
OUTPUT OUT=*RES* PREDICTED=*PREDY* RESIDUAL=*RESY*;
PLOT RESIDUAL.*PREDICTED. ;
RUN;
PROC PRINT DATA=*RES*;
RUN;

♦ THE END ♦