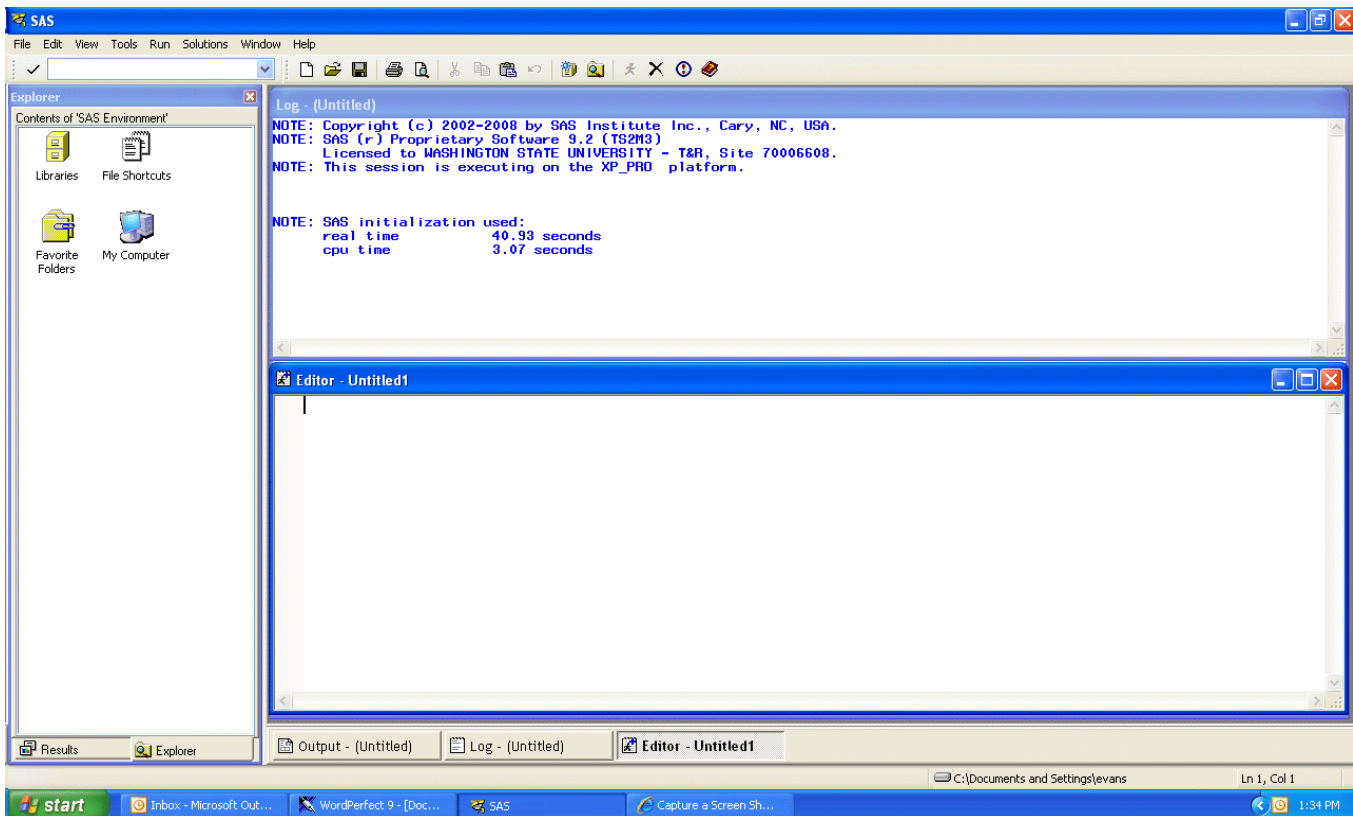


The SAS interface is shown in the following screen shot:

There are several items of importance shown in the screen shot. First there are the usual main menu items, such as File, Edit, etc. I seldom use anything but the File and Edit menu items. In the File menu there are the typical Open, Close, Save, Print, etc. items, while in the Edit menu there is Copy, Paste, Replace, etc. For the most part these are all I use. The Help menu may or may not be fully functional. In the past this would be the approach to opening the SAS online documentation. On my current PC I must open the online documentation by going to the START icon, going to All Programs, finding SAS in the list of programs and among the SAS software items is the Documentation.

On the Command bar (second bar from the top) are a series of icons, the most important of which are the Submit, Clear All and Break (4th, 3rd and 2nd from the last items). If you place the cursor over these icons you will get a brief description of each. The submit button submits a set of SAS code that is located in Editor window. All programs will originate from the Editor window. The Clear All button simply clears all text in the Editor and the Break button will stop SAS processing a set of code. This is very useful if you have made a coding mistake that results in an infinite loop or when some SAS procedures are running too long due to model complexity.

There are typically three windows of importance to SAS programs (only two of the important one are shown in the screen shot. I typically close the Explorer window in the left panel. The two of importance that are shown are the Editor, where all SAS code goes and the Log, which reports on operation of the SAS. All warnings and errors will show up in the Log. For this reason it is a good idea to look at the Log for problems. The other window of importance is the Output window. This is where all of the results from the submitted program will reside. You can bring any of the three windows into focus by clicking on the appropriate button at the bottom of the screen (e.g., Output, Log and Editor).



A Short Course on SAS Commands

Before any data can be analyzed, you must make the data available to SAS. This is performed in a DATA step.

Data Input

SAS manipulates SAS data files, whether it is to compute the statistics for an analysis of variance or to add two numbers together. A SAS data file persists only while the session is running (from the time you load SAS until you terminate the program). Therefore, to perform any analyses using SAS you must first create a SAS data file. There are several SAS statements required to create a SAS data file. These include the *DATA*, *INPUT*, *INFILE* and *CARDS* statements. Some of these statements have several *options* available. *All SAS statements end with a semicolon.*

There are essentially two methods of creation: either in the program stream or from an external data file.

The general form of a data step with the data entered in the program stream is:

```
DATA filename;  
INPUT variable_1 variable_2 ... variable_n;  
SAS statement_1;  
SAS statement_2;  
.  
.  
.  
SAS statement_n;  
CARDS;  
data row 1  
data row 2  
.  
.  
.  
data row n  
;  
RUN;
```

The general form of a data step with the data entered from an external file is:

```
DATA filename;  
INFILE "path and filename" options;  
INPUT variable_1 variable_2 ... variable_n;  
SAS statement_1;  
SAS statement_2;  
.  
.  
.  
SAS statement_n;  
RUN;
```

For all programs developed over the next several pages only the first approach will be used.

The SAS statements in the DATA step:

DATA filename;	Initializes the data step and assigns the “filename” to the SAS data file.
INFILE “path and file name” options;	Data will be read from an external file . The path and file name are specified in quotes. Two main options are of use here: dlm = “character” and firstobs = n. The “character” in the dlm option is typically a comma since this is the separator between columns of a comma separated value file (MS Excel data type). The firstobs option indicates from which row of the data file to start reading. Here n is an integer.
INPUT variable_1, variable_2, ..., variable_n;	Variables variable_1, variable_2, ..., variable_n are to be created using in-stream data entry. There are several options available for the INPUT statement, but only a couple will be discussed. Data may be numeric or character. The default data type is numeric. Character data must be followed by a \$. If data are missing, use a period. The @@ character at the end of an INPUT statement allows for data to be read in multiple complete sets from the same data line.
CARDS; or DATALINES;	Indicates that the data are to be entered in-stream and directly follow the CARDS statement.

SAS Statements in the data step can be used to manipulate data or create new variables. These SAS Statements appear in the data step following the INPUT statement. These SAS statements can be arithmetic, logical or functional in nature.

Arithmetic Operators

operator	function	example
**	raise to a power	X = Y**Z;
*	multiplication	X = Y*Z;
/	division	X = Y/Z;
+	addition	X = Y + Z;
-	subtraction	X = Y - Z;

Comparison Operators

operator	function	example
= or EQ	equal to	if Y = Z then X = 2;
^= or NE	not equal to	if Y ^= Z then X = 2;
> or GT	greater than	if Y > Z then X = 2;
< or LT	less than	if Y < Z then X = 2;
>= or GE	greater than or equal to	if Y >= Z then X = 2;
<= or LE	less than or equal to	if Y <= Z then X = 2;

Logical Operators

operator	function	example
& or AND	intersection of two logicals	if $Y = Z$ AND $X = 2$ then $W = X$;
or OR	union of two logicals	if $Y = Z$ OR $X = 2$ then $W = X$;
^ or NOT	complement	if NOT ($Y = 2$) then $W = X$;

SAS Functions

SAS function can be applied to variables or constants. For example, suppose you wanted to create a new variable X that was the natural logarithm of the variable Y. The functional form would be:

```
X = LOG(Y);
```

Here, LOG is the function for the natural logarithm and Y is the argument. The argument can be either a defined variable or a constant.

A Survey of SAS Functions

function	definition
ABS	Absolute value
SQRT	Square root
ROUND	Rounds to the nearest roundoff unit
LOG	Natural logarithm
LOG10	Logarithm base 10
ARCOS	Arc-cosine
ARSIN	Arc-sine
ATAN	Arc-tangent
COS	Cosine
SIN	Sine
TAN	Tangent
POISSON	Poisson probability distribution function (cumulative probability)
PROBBNML	Binomial probability distribution function (cumulative probability)
PROBCHI	Chi-square probability distribution function (cumulative probability)
PROBF	F probability distribution function (cumulative probability)
PROBNORM	Standard normal probability distribution function (cumulative probability)
PROBT	T probability distribution function (cumulative probability)
CINV	Inverse of the chi-square distribution function (quantile)
FINV	Inverse of the F distribution function (quantile)
PROBIT	Inverse of the normal distribution function (quantile)
TINV	Inverse of the T distribution function (quantile)

SAS Procedures

The effectiveness of SAS as a statistical software package lies in the union of the ability to manipulate data in the data step and then send the data to a procedure for further analysis. SAS procedures are essentially functions. However, unlike the functions just described, SAS procedures are far more complex and produce an array of output. All SAS procedure call begin with the key work PROC. Some of the basic statistical procedures supplied by SAS include PROC UNIVARIATE, PROC GLM, PROC REG, PROC PLOT PROC ANOVA PROC MIXED and PROC LOGISTIC. There are many more SAS PROCs, too many to discuss here. The procedures of interest to the analysis of variance include:

PROC GLM	general linear models procedure for analysis of variance and regression
PROC ANOVA	analysis of variance procedure
PROC MIXED	mixed model analysis of variance procedure
PROC UNIVARIATE	univariate statistics and tests for normality
PROC PLOT	X, Y scatter plot procedure
PROC SORT	sort data in ascending or descending order
PROC TTEST	perform both an independent two sample t-test and a paired t-test
PROC REG	regression analysis procedure

The use of arithmetic operators, comparison operators, logical operators, SAS functions and SAS procedures are discussed in detail within the online documentation. You should explore these pages for your own benefit. The arithmetic operators, comparison operators, logical operators, SAS functions can be found in the Base SAS pages, while the procedures documentation can be found in the SAS/STAT pages (near the bottom of the list).

Now that you have some understanding of the lay-out of SAS, it is time to create a SAS data set and use a procedure.

Example 1 - SAS Data File Creation Using In-Stream Data Entry:

In the following example, a SAS data set named *new* will be created with variables X, Y and Z. X will be numeric, while Y and Z will be character variables. SAS reads data line by line, reading from left to right. The data set “new” is printed (results will appear in the Output window) using the Print procedure. To invoke a procedure you simply use the term “proc” followed by the procedure name. Each procedure will have many possible options or statements. For the Print procedure the only option used is specification of the data set to use, “new” in this case.

```
data new;
input X Y $ Z $;
cards;
10 Kim Forester
15 Jim Green
25 Peter Zanis
;
Proc Print data = new;
run;
```

The “Proc Print data = new;” and “run;” statements are not needed for the data step, they are needed to see the results of the data step.

Entering the example SAS code into the SAS Editor window and clicking on the Submit button produces the following screen shot:

