Mike Zdeb (msz03@albany.edu)

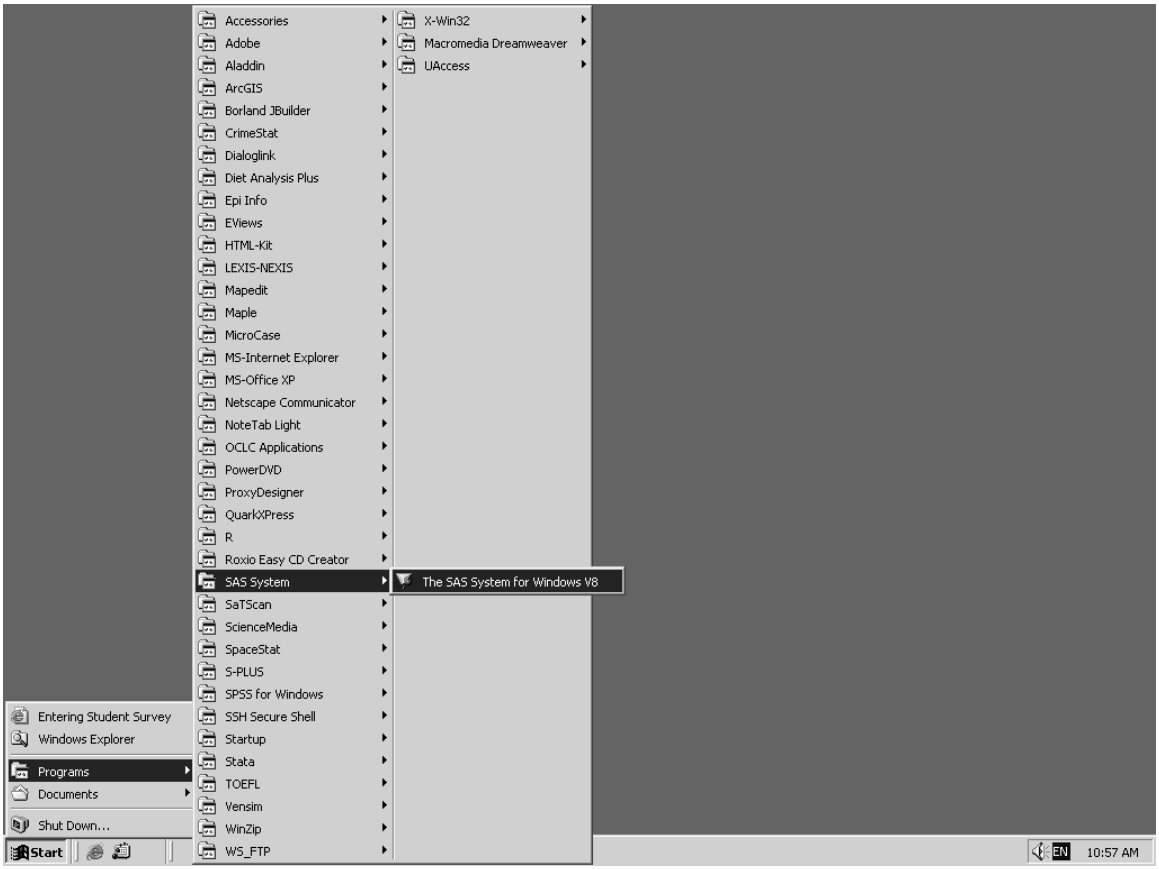
If you have successfully written computer programs before, learning SAS should be quite easy. If you have never written computer programs, it still might be easy, but chances are programming might require you to be just a bit more 'exact' in the way you express ideas than you do in your other speaking or writing. You can think of SAS as a foreign language with a very exact syntax. You won't be trying to make another person understand your commands, rather you'll be trying to make a computer understand something you want to do with your data (organize it, combine it with other data, write reports, run analyses, create graphics, post it on the web). The course EPI514 teaches you how to use SAS to organize, manage, and present data. The advanced SAS course (currently one of the several EPI697 courses) expands on the topics covered in EPI514 and also introduces some other powerful data management, organization, and presentation tools available in SAS. Neither course teaches data analysis. That's left to the BIOSTAT and EPI courses you'll take.

Start SAS by clicking on the SAS icon (look under programs after clicking on the Windows START tab). You will be using SAS in what is called DISPLAY MANAGER MODE. You should see a window that looks like the display at the bottom of this page. There are several parts to this default SAS display:

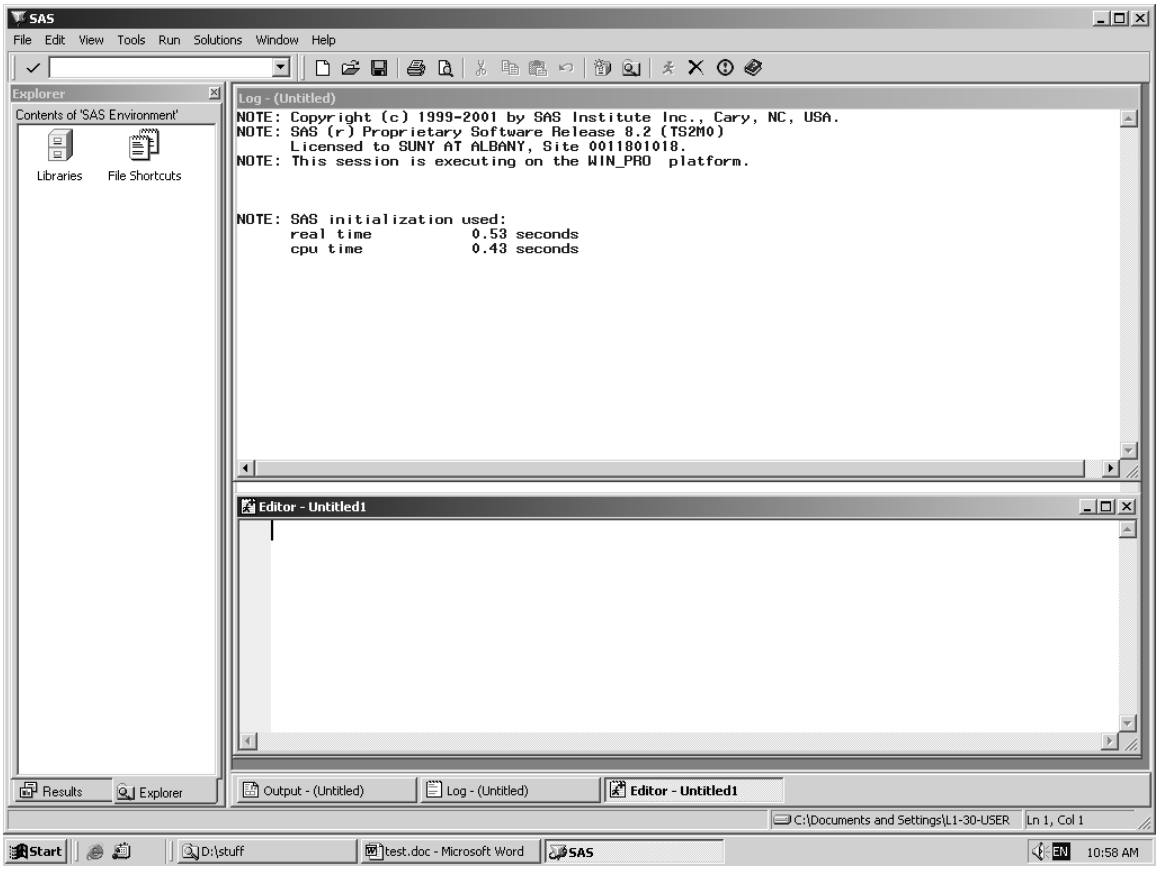
EDITOR - This is the window where you can write SAS programs or insert SAS code that you have written previously. It is also the window in which you SUBMIT SAS code for execution.

LOG - This is the window where you see notes, warnings, and error messages after you submit SAS code for execution.

EXPLORER - This window gives you easy access to SAS data libraries. NOTE: I rarely (if ever) use this window and do not even mention it during my SAS classes.



In addition to the various windows, you'll also see: a TOOLBAR starting with FILE and ending with HELP; a small window in the upper left that can be used to enter commands (enter the word KEYS, click on the CHECKMARK, and you'll see that various KEYS can also be used to enter commands - click on the X on the upper right of the KEYS window or press F3 to close the window); another TOOLBAR to the right of the command window that displays only ICONS, no text (the function of each icon will appear as you drag the mouse cursor over the various icons).

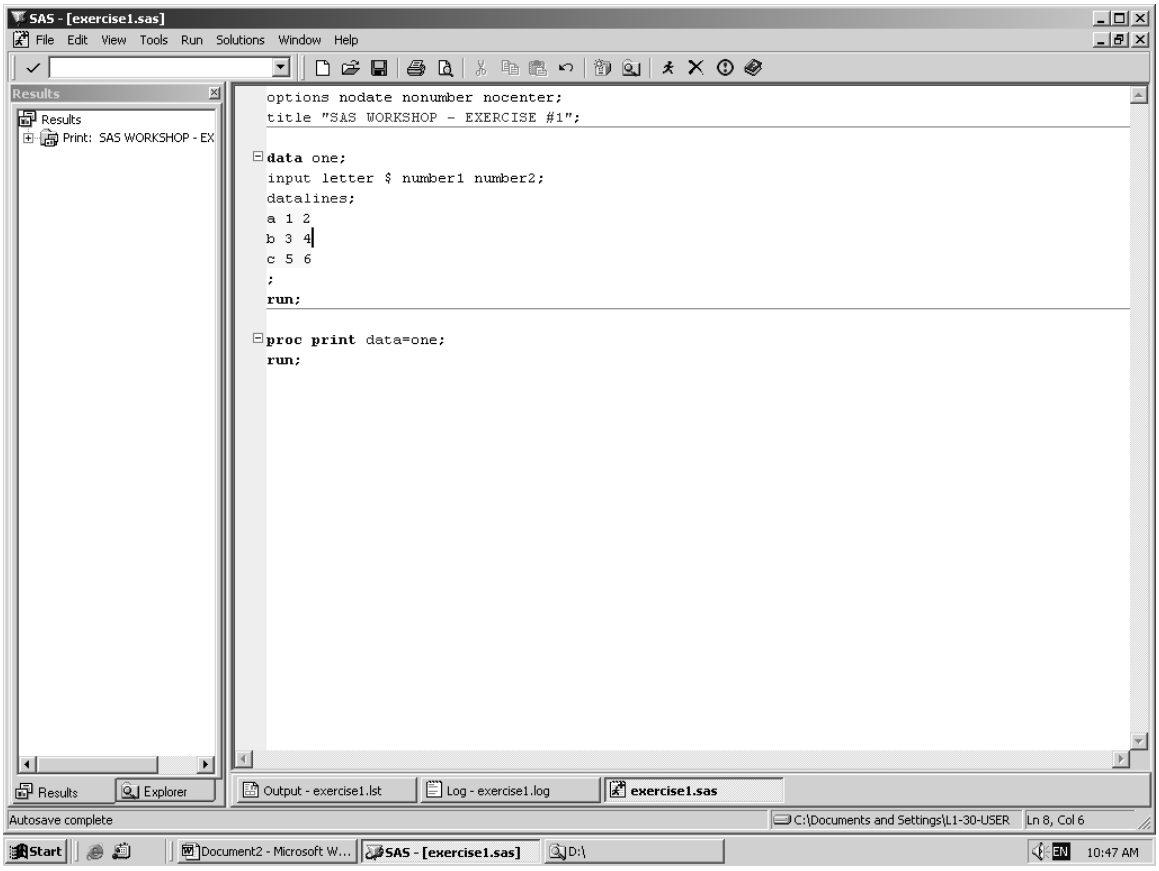


Exercise #1

Maximize the size of the EDITOR window by clicking on the box on the right of the blue bar above the editor window. Enter the following...

```
options nodate nonumber nocenter;
title "SAS WORKSHOP - EXERCISE #1";

data one;
input letter $ number1 number2;
datalines;
a 1 2
b 3 4
c 5 6
;
run;
```

```
proc print data=one;
run;
```

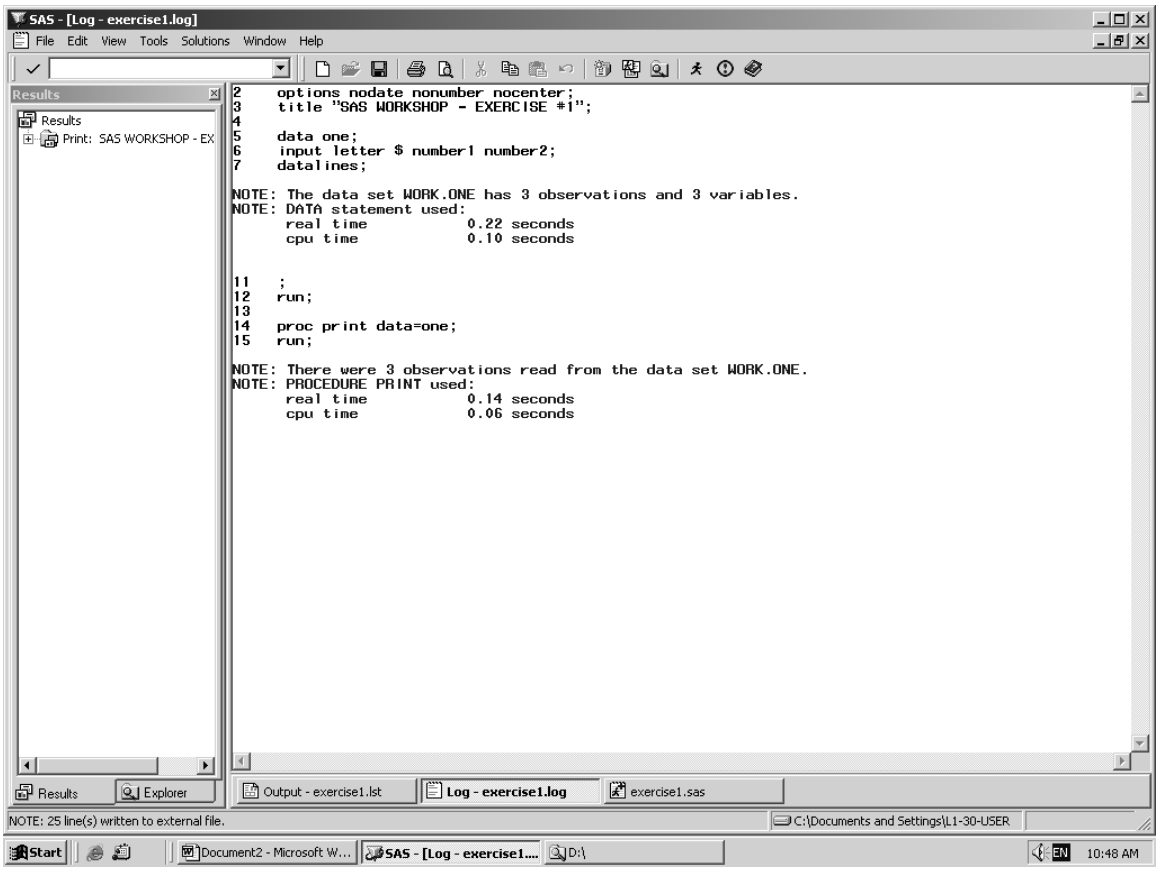


After you have entered all the SAS code, either...

press F8, or
click on the toolbar icon that looks like a running person, or
click n the word RUN at the top of the screen and select the option SUBMIT

You'll discover that there are a number of different ways to perform common tasks in the SAS display manager.

Running a SAS job should always produce a LOG file (click on the LOG tab on the bottom of the SAS screen). The SAS log is where you will see messages from SAS about your program — notes, warnings, and error messages. If you typed the SAS code correctly (spelled all the key words correctly and didn't leave out any semi-colons), you should see a SAS log that looks as shown on the next page. It should only contain NOTES (no WARNINGS, no ERRORS). The notes tell you that you have converted your data (in the datalines file) into a SAS data set...data in a proprietary format that SAS can 'understand'. If you had chosen to enter your data into an Excel spreadsheet, you would have saved it in an Excel worksheet (a file with a .XLS extension that contains data in a proprietary format that Excel can' understand'). Your SAS data set has 3 observations (it would have had 3 rows in the spreadsheet) and 3 variables (it would have had 3 columns in the spreadsheet).



In the DATA step, there is a DATA statement. In that statement you gave a name to the SAS data set you created, ONE. If you had entered the data in Excel, you could have saved it with a name ONE.XLS and you would have saved it (hopefully) in some location on you computer where you could find the file when you wanted to use the data again. It's not important to know where the SAS data set is located (hint: it's on the hard drive of the computer) and what the SAS equivalent is to the .XLS extension (hinbt: its' SAS7BDAT).

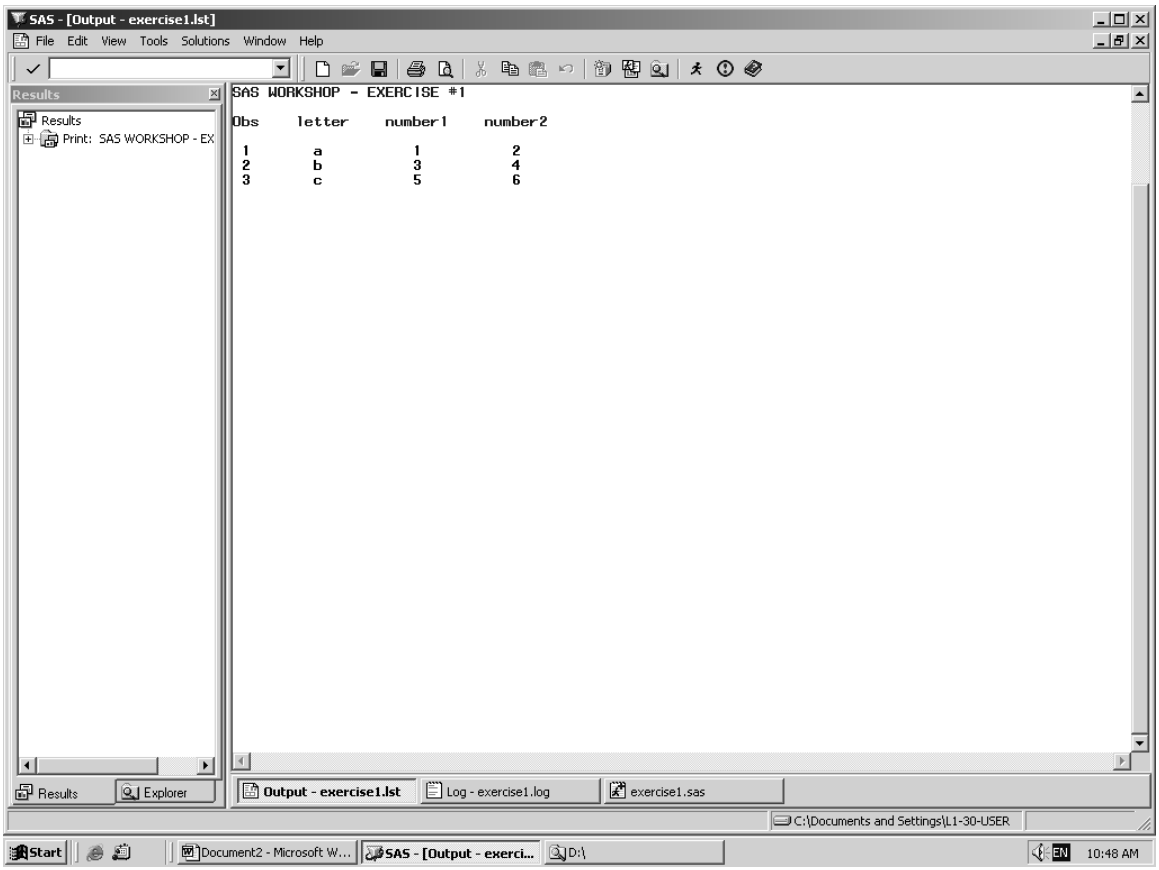
In addition to the SAS LOG, you should also have SAS OUTPUT — produced by the SAS procedure (or PROC) PRINT (click on the OUTPUT tab on the bottom ofthe SAS screen). There are numerous SAS procedures that are use to analyze and present data. The PRINT procedure is an easy way to display the values of variables in a SAS data set. The OUTPUT window should look as shown on the next page. Normally...

there is a date on each page
there is a page number on each page
the output is centered

You changed the above defaults by including the line...

```
options nodate nonumber nocenter;
```

as part of your SAS job.

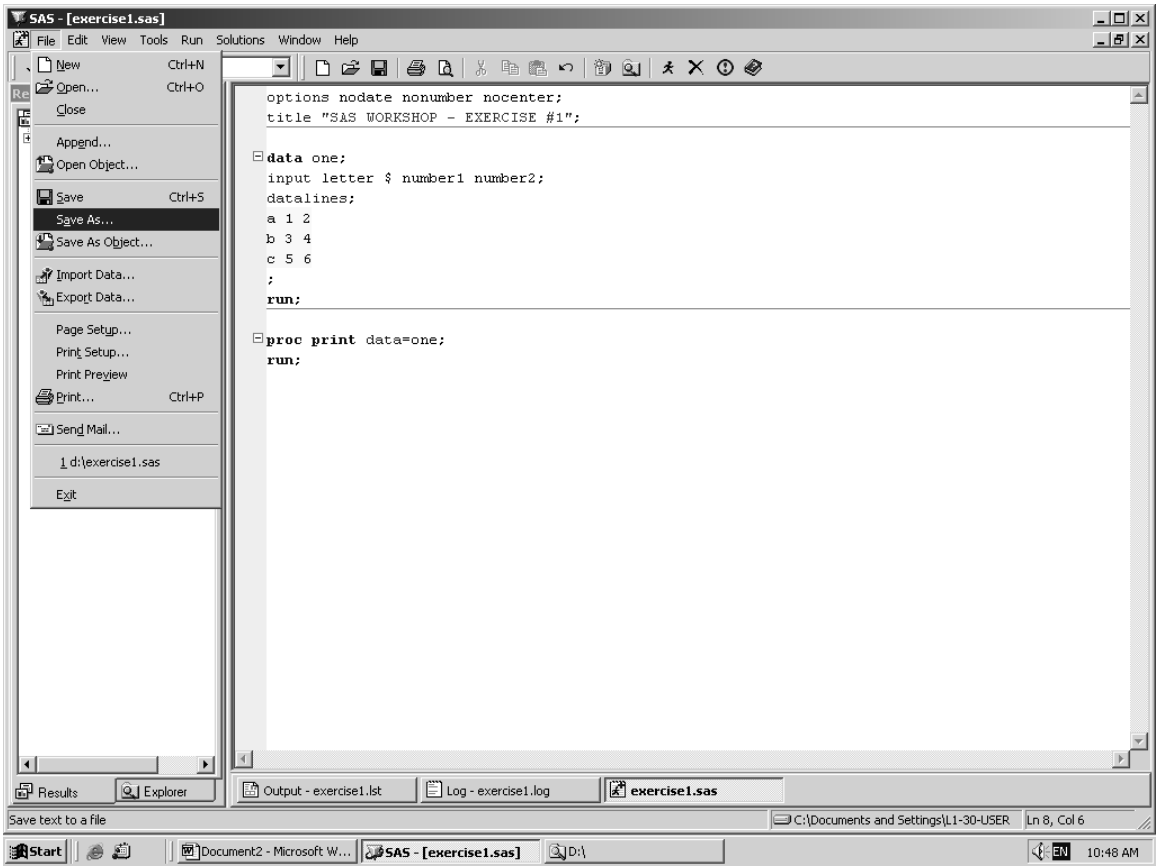


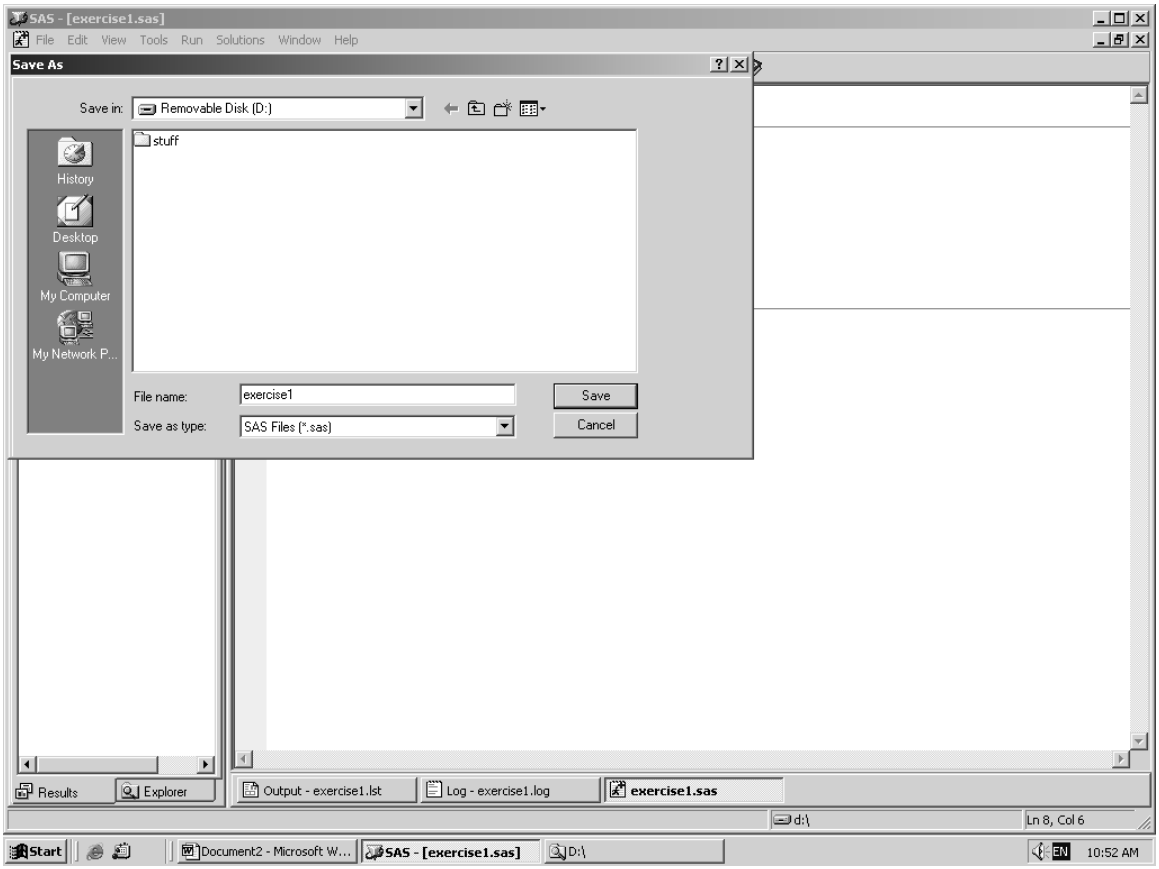
Your output has no date or page number, and it's left-justified. There is a title at the top of the page (also left-justified) resulting from the title statement...

```
title "SAS WORKSHOP - EXERCISE #1";
```

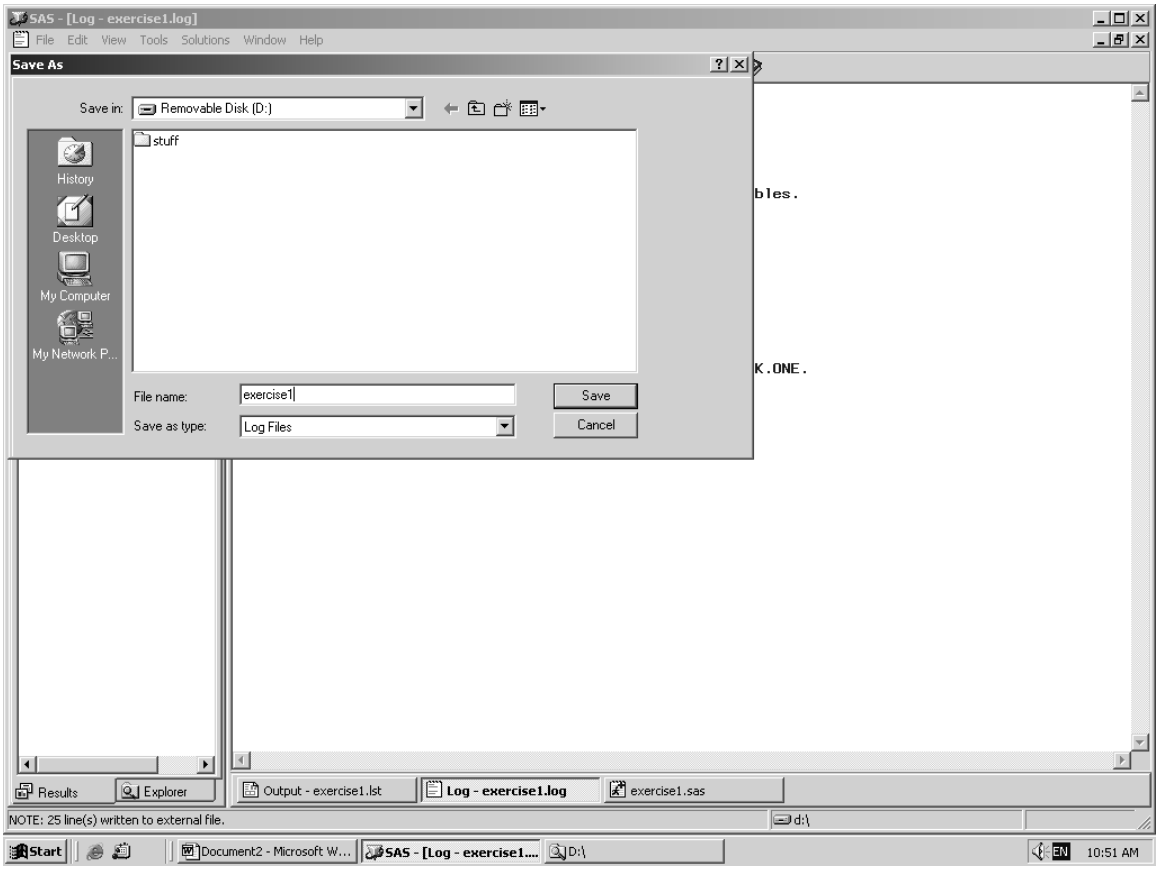
PROC PRINT adds an observation number on each line and uses the variable names as the column headers.

You can save the contents of any of the SAS windows. If you have written a long SAS job, correcting errors as you go along, you'll probably want to save your work so you can use it again during another SAS session. Place the cursor in any place within the editor window and then choose FILE / SAVE AS (as shown on the next page). Notice that the next window you see assumes that the file extension given to this file will be .SAS. It's good practice NOT to change the file extension. Save your SAS code in files with a .SAS extension.


SAS - [exercise1.sas]
File Edit View Tools Run Solutions Window Help
New Ctrl+N
Open... Ctrl+O
Close
Append...
Open Object...
Save Ctrl+S
Save As...
Save As Object...
Import Data...
Export Data...
Page Setup...
Print Setup...
Print Preview
Print... Ctrl+P
Send Mail...
1 d:\exercise1.sas
Exit
options nodate nonumber nocenter;
title "SAS WORKSHOP - EXERCISE #1";
data one;
input letter $ number1 number2;
datalines;
a 1 2
b 3 4
c 5 6
;
run;
proc print data=one;
run;
Results Explorer
Output - exercise1.lst
Log - exercise1.log
exercise1.sas
Save text to a file
C:\Documents and Settings\L1-30-USER
Ln 8, Col 6
Start
Document2 - Microsoft W...
SAS - [exercise1.sas]
D:\
10:48 AM


SAS - [exercise1.sas]
File Edit View Tools Run Solutions Window Help
Save As
Save in: Removable Disk (D:)
stuff
History
Desktop
My Computer
My Network P...
File name: exercise1
Save as type: SAS Files (*.sas)
Save
Cancel
Results Explorer
Output - exercise1.lst
Log - exercise1.log
exercise1.sas
d:\
Ln 8, Col 6
Start
Document2 - Microsoft W...
SAS - [exercise1.sas]
D:\
10:52 AM

There are occasions when you might also want to save the LOG file that resulted from running your SAS job. One reason might be that you have encountered ERRORS that you do not understand and can't correct. In that case, you might want to attach the LOG to an e-mail and send it to someone who knows a bit more about SAS than you do. Place the cursor in any place in the LOG window and choose FILE/SAVE AS. You'll see a window that assumes you want to save the file with the extension .LOG.



Can you figure out how to save the contents of the OUTPUT window?

Clear out all of the windows. You can press CTRL-E while the cursor is in any window to clear out the window (or choose EDIT/CLEAR). See if you can bring your SAS code back into the EDITOR window by using FILE/OPEN (or the appropriate ICON on the toolbar).

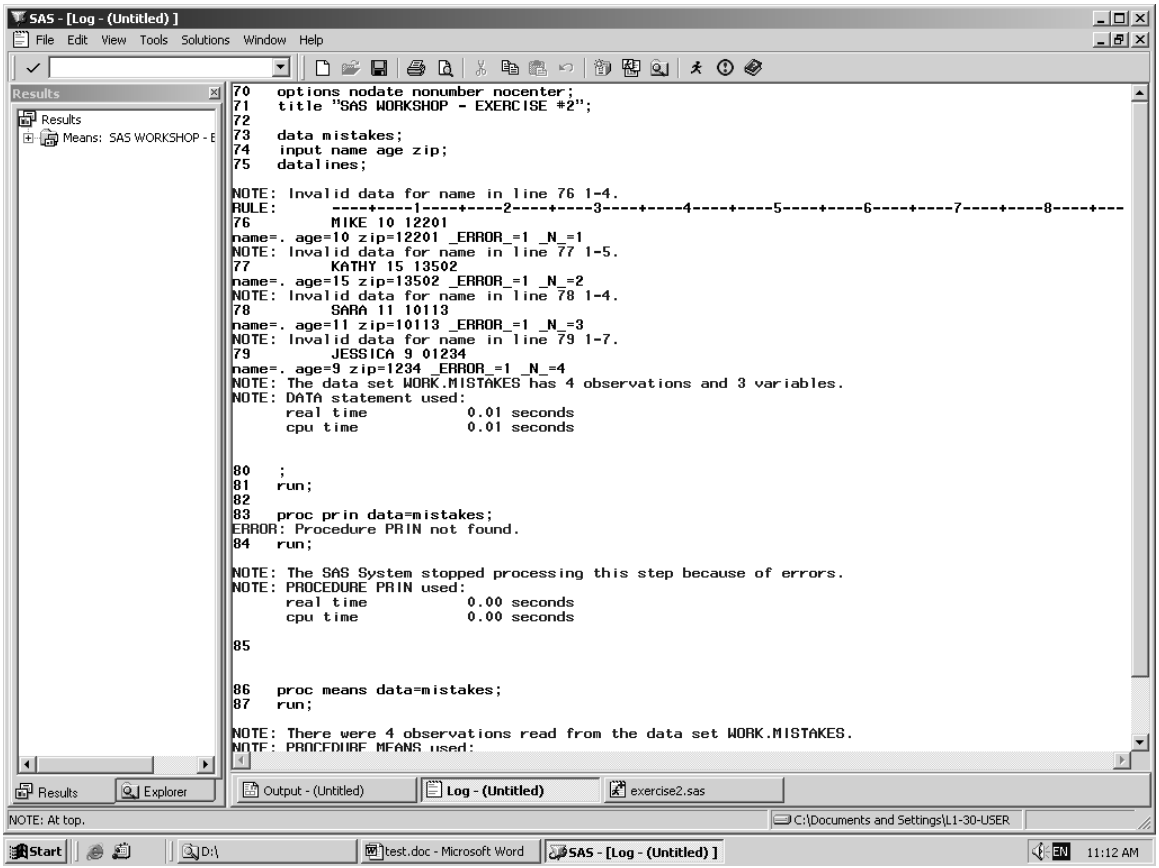
Exercise #2

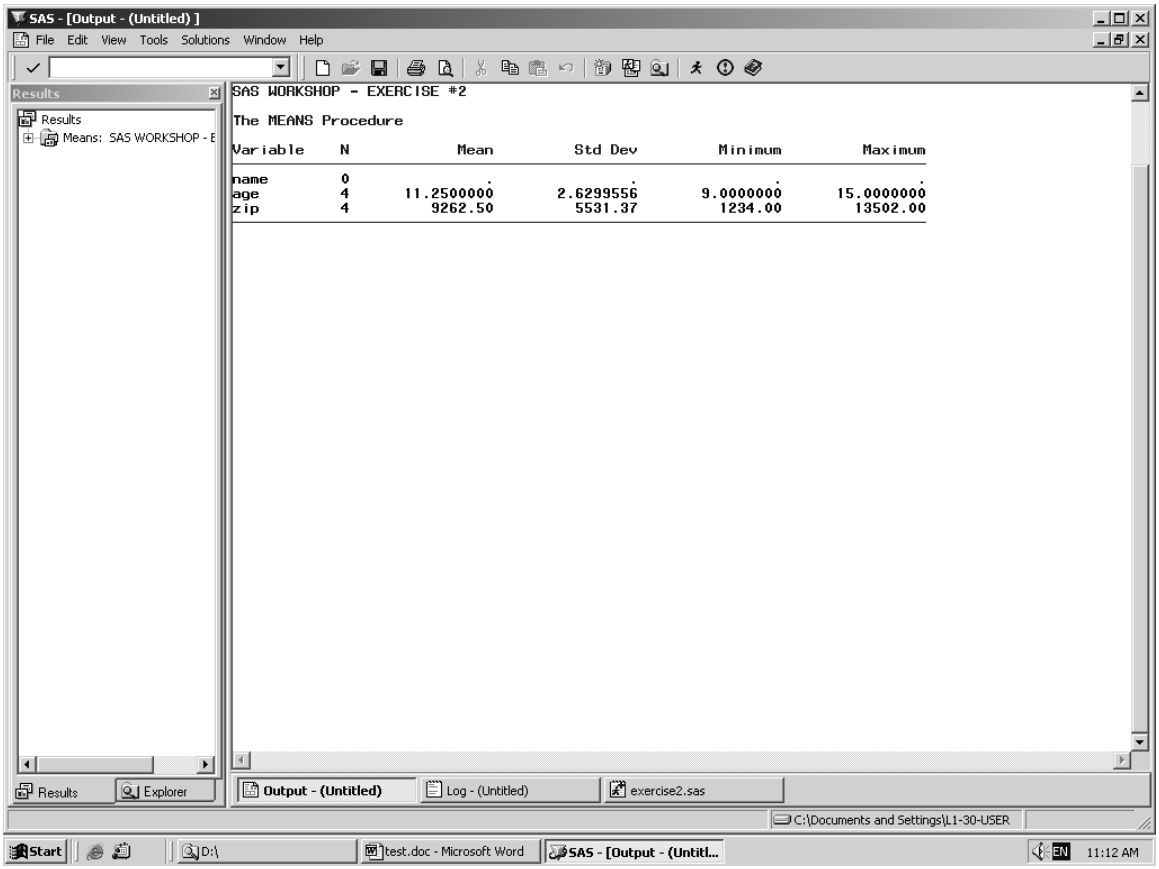
If SAS is still running, EXIT SAS, then start it again. Enter the following in the EDITOR window...

```
options nocenter formdlim = '-';;
title "SAS WORKSHOP - EXERCISE #2";

data mistakes;
input name age zip;
datalines;
MIKE 10 12201
KATHY 15 13502
SARA 11 10113
JESSICA 9 01234
;
run;

proc prin data=mistakes;
run;

proc means data=mistakes;
run;
```

This job contains errors that will produce ERRORS (not just NOTES) in the LOG. Notice in the LOG (next page) that there are errors in the data step. In exercise #1, the INPUT statement included a $ after the name of the first variable to tell SAS to possibly expect non-numeric data as values for the variable LETTER.

In this exercise, there are names in the data and you have tried to read the values without using the $. Unless you tell SAS otherwise, all variables are assumed to be NUMERIC. Notice that the LOG says that you have invalid data for the variable NAME. Next, you have misspelled PRINT (your first RED ERROR message). So, there is no output from PROC PRINT. Finally, the output from PROC MEANS shows no values for the variable NAME, real data for the variable AGE, and statistics computed on the ZIP code. By default, PROC MEANS analyzes all NUMERIC variables in a data set (in this case, that includes NAME and ZIP).


SAS - [Log - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Means: SAS WORKSHOP - E
70 options nodate nonumber nocenter;
71 title "SAS WORKSHOP - EXERCISE #2";
72
73 data mistakes;
74 input name age zip;
75 datalines;
NOTE: Invalid data for name in line 76 1-4.
RULE: ----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----8----+---
76 MIKE 10 12201
name=. age=10 zip=12201 _ERROR_=1 _N_=1
NOTE: Invalid data for name in line 77 1-5.
77 KATHY 15 13502
name=. age=15 zip=13502 _ERROR_=1 _N_=2
NOTE: Invalid data for name in line 78 1-4.
78 SARA 11 10113
name=. age=11 zip=10113 _ERROR_=1 _N_=3
NOTE: Invalid data for name in line 79 1-7.
79 JESSICA 9 01234
name=. age=9 zip=1234 _ERROR_=1 _N_=4
NOTE: The data set WORK.MISTAKES has 4 observations and 3 variables.
NOTE: DATA statement used:
real time 0.01 seconds
cpu time 0.01 seconds
80 ;
81 run;
82
83 proc prin data=mistakes;
ERROR: Procedure PRIN not found.
84 run;
NOTE: The SAS System stopped processing this step because of errors.
NOTE: PROCEDURE PRIN used:
real time 0.00 seconds
cpu time 0.00 seconds
85
86 proc means data=mistakes;
87 run;
NOTE: There were 4 observations read from the data set WORK.MISTAKES.
Results Explorer
Output - (Untitled) Log - (Untitled) exercise2.sas
NOTE: At top.
C:\Documents and Settings\L1-30-USER
Start D:\ test.doc - Microsoft Word SAS - [Log - (Untitled)] 11:12 AM


SAS - [Output - (Untitled)]
File Edit View Tools Solutions Window Help
Results
Means: SAS WORKSHOP - E
SAS WORKSHOP - EXERCISE #2
The MEANS Procedure
Variable N Mean Std Dev Minimum Maximum
name 0 . . . .
age 4 11.2500000 2.6299556 9.0000000 15.0000000
zip 4 9262.50 5531.37 1234.00 13502.00
Results Explorer
Output - (Untitled) Log - (Untitled) exercise2.sas
C:\Documents and Settings\L1-30-USER
Start D:\ test.doc - Microsoft Word SAS - [Output - (Untitl... 11:12 AM

You can correct the SAS code an rerun the job...

```
options nocenter formdlim='-';
title "SAS WORKSHOP - EXERCISE #2";
data mistakes;
input name $ age zip $;
datalines;
MIKE 10 12201
KATHY 15 13502
SARA 11 10113
JESSICA 9 01234
;
run;

proc print data=mistakes;
run;

proc means data=mistakes;
run;
```

If you rerun the job, you'll notice that rather than having the output on separate pages (the default, you will see a line of dashes between the various procedure outputs. This is a result of using the FORMDLIM option (a good way to save paper if you print the output).