Class Exercise 10

This exercise is based upon Chapter 12 of the SAS Advanced Certification Prep Guide.

We will be storing SAS macros in several different ways. To distinguish between the different procedures, we will use multiple macros. You should have the code for the following macros available: REGSIMP, PRINTSUBSET, DEMBPLOT, and FISCAL. Note that a couple of these macros refer to specific data sets. Be sure the following data sets are loaded into the WORK library: HSB2, TODATE, and ALL. All the code for these macros and data sets can be found in the Chapter 11 code on the website.

To start, highlight and submit the REGSIMP macro. Now look at the WORK library and report what you see there.

Save FISCAL as a text file in your course directory. Use a %INCLUDE statement to compile the code and then test it:

filename fiscal "F:\STAT 541\fiscal.txt";

%INCLUDE fiscal;

%fiscal(FY)

How do you feel about storing macros as separate text files and then accessing them with %INCLUDE? %INCLUDE can also be used to compile long pieces of regular SAS code, and can thus simplify otherwise lengthy programs.

Now we will outline the steps to create a macro catalog in your course directory. I think my definition of a libref in class was redundant; rather than create a separate macro library, and then a macro catalog, you should use your main course directory as the library:

libname stat541 "F:\STAT 541";

run;

Now we’ll save PRINTSUBSET as a SOURCE file in a macro catalog in your main course directory—this takes multiple steps. First, open a new editor window by selecting New Program under File in SAS’s main menu bar—the new window should appear. Paste the code for PRINTSUBSET in the window. Now select Save As Object… under File, and then double-click on STAT541 under Libraries. At this point you need to create a new catalog—click on the New Catalog icon (the folder with a sparkly upper left corner and red ball in the lower right corner) and enter Macros as the catalog name (rules about catalog names are just as strict as rules for library names—they can only be 8 characters long, among other things).

At this point, you should double-click on the new catalog Macros, and you will then be prompted for Entry Name (the correct Entry Type—SOURCE—should be pre-selected. You should enter PRINTSUBSET as the Entry Name and then click Save. My version of SAS has been uncooperative with this procedure—the Save button is sometimes not highlighted. As a work around, you can right-click in the blank Macros window, select New.. and then select SOURCE Program. A text editor window will open; you can paste your macro code in this window and then close the window—the file will be saved automatically. The macro will be labeled Untitled, but you can right-click it to rename it PRINTSUBSET. Whew!

Though the above approach was tedious, this does seem like the most natural way to store macros. Repeat the procedure with DEMBPLOT so that two macros are now stored in the catalog. Graduate students should run PROC CATALOG to confirm that the macros are in the STAT541.Macros catalog.

To use these macros, let’s use the FILENAME shortcut, and the CATALOG option:

filename macdemo catalog "stat541.macros";

%include macdemo(printsubset) / source2;

%include macdemo(dembplot) / source2;

%put &syslast;

Look in the LOG to see that the source was loaded and that WORK.HSB2 is your most recent file (if not, re-run it). Now let’s run the macros to make sure they worked:

%printsubset(1,Female Students)

%dembplot(socst,group=race)

Finally, let’s store both the source code and compiled FISCAL macro (where is the source code stored?):

options mstored sasmstore=stat541;

%macro fiscal(report)/store source;

%if %upcase(&report)=FY %then %do;

title 'Fiscal Year Debits and Credits'; proc means data=all;

var debit credit;

run;

%end;

%else %do;

title 'To-Date Debits and Credits';

proc means data=todate;

var debit credit;

run;

%end;

%mend fiscal;

Run the following code to verify that the compiled code was saved, and then check the LOG to verify that the source code was saved.

%fiscal()

%copy fiscal/source;