PHPM 672/677 Assign#6: Macros
Due date: Submit on E-Campus by 11:59pm Monday 4/11
Midpoint LAB due date: on E-Campus by 11:59pm Monday 4/4

Submission. Submit on E-campus by 11:59pm the day before the class they are due.

1. Lab (2 points)
   - Submit as midpoint check in one week
2. Assignment (6 points)
   - Commented code (lnameN.sas; where N indicates the assignment number)
   - Output from your code (lnameN.log & (lnameN.lst or lnameN.html))
   - Answers to questions: Readme.txt

Late Assignments. Each student will be allowed one late assignment, due 7 days from the due date. NO other late assignments or make up will be accepted.

Plagiarism: If you consult any outside sources when doing your work, you are expect to further document these sources. Give credit where credit is due. Plagiarism will not be tolerated.

All handed in homework should state at the top any assistance with debugging and programming, as well as citations of any program segments copied from a website.

Guideline for assignment grading (Total of 8)

- Assignment (Total 4)
  - 1: Submitted code that does not run.
  - 2: Mostly running but incorrect.
  - 3: Correct and meets requirements (i.e use of arrays and loops)
  - 4: Correct & Elegant. Comments.
- Answers to questions on the assignment (Total 2)
- Midpoint Check Lab (Total 2) – Remember to submit in ONE WEEK.

Objective

By the end of this assignment, you should be able to

- Read and write SAS macro variables
- Read, use, and modify SAS macro functions

Assignment 6: Macro variables and functions

Setting Up: Download program files from the class website (config_a6.sas & assign6.sas) into the appropriate folder. The required input data file is under data menu for assignment 2 (ipstatw.zip).

Blue highlights are edits you have to make to the program. Yellow highlights are items you have to put in the readme file.

1. There should be two sas code (config_a6.sas & assign6.sas).
2. You should also have downloaded the required input data and put it in the correct folder.
3. Open assign6.sas and edit the code so that it can properly locate the config_a6.sas code.
4. Open config.sas and edit the code so that the libname is referencing the correct folder.
5. Also, set `options` as needed.
6. READ the sas code from top to bottom and try to understand what is happening. This is from lab 2 & 3, with less lines of code (so less number of tasks done) so it should be familiar to you.
7. Now *resolve*("find and replace") all macro variables and functions used
   a. In the readme file: First, copy and paste all lines of code that are using either a macro variable or functions. (Hint: there are three)

b. In the readme file: For each line, how is the line of code resolved? That is, each line of code, type in the regular SAS program that is generated by the macro used.

8. Now we are going to see if we can read other similar macros provided: `%concatI, %concatM, %concatMC.` This is only a mental exercise; do not make any changes to the program for this section. Just type things into the readme file. Also, some of the macros might not make logical sense to apply to the parameters. But just be the computer, and resolve the code as given.
   a. In the readme file: First, copy and paste the one line (that I provided) using `%concatQ`
   b. In the readme file: how is this line of code resolved? That is, type in the regular SAS program that is generated by the macro used.
   c. In the readme file: Now change `%concatQ(dc, 20101, 20124)` with `%concatI(dc, 20101, 20124, 2)`. How is this line of code resolved? That is, type in the regular SAS program that is generated by the macro used.
   d. In the readme file: What if `%concatM(dc, 201001, 201204)` was used instead? How is this line of code resolved? That is, type in the regular SAS program that is generated by the macro used.
   e. In the readme file: What if `%concatMC(dc, 201001, 201204)` was used instead? How is this line of code resolved? That is, type in the regular SAS program that is generated by the macro used.
   f. In the readme file: Explain succinctly in plain English what each of the `%concatX` macro functions does. There are a total of 4.
9. In the program: Find three places where you can use the macro `%concatQ`, and edit the code to use it. That is, delete the regular SAS code, and replace it with code that uses the macro
10. Run your code and make sure that everything is running properly before you move to the next step.
11. Locate P8.1 & P8.2 in the comments (in the label statement).
    a. P8.1: label the 12 discharge variables in the same format as the example for dc20101 using regular sas code.
    b. Save and run your code to make sure that everything is running properly before you move to the next step. If you have bugs, iteratively debug your regular SAS code until it is correct.
    c. Now, copy and paste the lines of code you wrote to label the 12 variables into the readme file.
    d. Now you will write the required macros to replace the sas code you just wrote by modifying one of the macros functions I provided. First, copy the macro `%concatQ`, and name it to be `%labelQ` in the config file. Now modify the macro so that you can use it label the 12 variables. Then, use it in the program.
12. Edit the program to use a different set of counties. Make sure to use the macro variable CNTY.
13. In the readme file:
    a. What is the average number of patients discharged over all hospitals in 2011 quarter 2?
    b. How many hospitals discharged more patients than the average number in 2011 quarter 2?
    c. Which counties are this hospitals located in?

Readme file (Identical to the Yellow highlights above):

1. First, copy and paste the all lines of code that are using either a macro variable or functions. (Hint: there are three)
2. For each line, how is the line of code resolved? Type in the regular SAS program that is generated by the macro used.
3. First, copy and paste the one line using %concatQ
4. How is this line of code resolved? That is, type in the regular SAS program that is generated by the macro used.
5. Now change %concatQ with.%concatI. How is this line of code resolved?
6. What if %concatM was used instead? How is this line of code resolved?
7. What if %concatMC was used instead? How is this line of code resolved? In the readme file: Explain succinctly in plain English what each of the `%concatX` macro functions does.
8. Copy and paste the regular SAS code you wrote to label the 12 variables
9. What is the average number of patients discharged over all hospitals in 2011 quarter 2?
10. How many hospitals discharged more than the average number of patients discharged in 2011 quarter 2?
11. Which counties are these hospitals located in?