PHPM 672/677 Assignment #5: Reshaping & Combining Tables (Big)
Due date: Submit in E-Campus by 11:59pm Monday 3/16 (1 week)

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

1. Assignment (4 points)
   - Commented code (lnameN.sas; where N indicates the assignment number and lname is your last name)
   - 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.

Required & recommended readings for this assignment

1. UCLA link Optional 3: Combining Tables (Lec 6-7) on course website
2. The little SAS book (online book available from the library): Chapters 1 & 6

Guideline for assignment grading (Total of 4)

- Assignment
  - 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

Assignment 5: Reshaping and Combining Tables (Big)

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

- `set`: Append multiple tables (e.g stack tables on top of each other to increase the number of rows)
- Link up multiple tables using a shared key (e.g align the rows using the shared key, and link multiple tables to increase the number of variables in the tables) using `merge`.
- Aggregate/Combine multiple rows into one row by group processing `proc summary`
- Reshape tables to flip rows & columns using `proc transpose`
  - Also transpose (flip rows & columns) by groups of rows

[STEP 1: Download] Setting Up: Get data from website into the appropriate folder: Assignment 5 uses the same data files as assignment 4. Put the original data files, that cover all 254 counties in TX, into the data folder for assignment 5.

[STEP 2: Adjust your program for assignment 4] Either use your code from assignment 4, or download the answer from the website.

1. Open the program and READ the program. Especially if you are using the posted answer.
2. Edit the program to have the libname point to the correct folder

[STEP 3: Combine all the tables] If you did everything correctly, the code should run with no problem. Although the code runs, you will see that you have more rows than you should have (254*3). This is due to some data errors in the original data that you have to identify and fix. A strategy for this was discussed on the record linkage lecture. Try to fix this, and answer the questions, below in the readme file, using the full data. Also note that there are additional questions 7 to 9, which require different unit of analysis

There are issues with the data that make some rows not line up properly. Identify which rows these are and figure out how to get them to line up. Whenever you are merging tables this way, make sure to check which rows are lining up in all tables, and which rows are not using the `in` option. For the rows not lining up, check if these are expected to not line up (really missing in some tables, or not lining up due to a

problems in the data). Hint: check the county names in the tables (look up `compress` & `lowcase` functions). There are 254 counties in TX. The county lookup tables has the correct names. One way to check county names in each table is to link up with the lookup table and look at rows that are not linking up properly since all should link up to one of the 254 rows in the lookup table. Once you identify what needs to be fixed, you can use the if conditional statement to reassign to correct values

[Read Question 7 below first] If you are done answer questions upto 6, now we will change the unit of analysis to "PHR (Public Health Region) Year". Aggregate "county year" data to have 1 row per "PHR Year". Think about what is the proper statistics (sum, mean, median, etc) for the variables you need (i.e. you don't need to have all the vars, just what you need to answer the question 7 below) when aggregating.

[Read Questions 8-9 below first] Next we will change the unit of analysis to "MSA (Metropolitan Statistical Area) Border".

- First, aggregate the "county year" data to 1 row per "MSA Border Year". You should remove all rows that do not have the MSA designation
- Second, reshape the table from long (years going down the rows) to wide (years going across columns) per "MSA Border". You should have exactly 31 rows (number of MSAs).
- Third, using arrays, calculate the percent increase in patients discharged each year (i.e. 2011 & 2012).

Readme file: Answer the following questions in a text file called readme.txt

<County Year Analysis>

1. How many total observations do you have when the merge is complete?
2. How many observations are missing patient data?
3. How many counties in your dataset are missing patient data for AT LEAST 1 year? (notice, we want the number of counties, not the number of observations that are missing patient data) ?
4. What is the average size (in terms of the population variable) of all observations (county/years) that have missing patient data?
5. Which observations (which county/years) have more deaths than inpatient discharges? How many observations have more deaths than physicians ?
6. Compare the Ratio of DPC per 100,000 Population between the border and non-border counties. Which has a higher average Ratio of DPC per 100,000 Population? Also give the actual values (DPC=direct primary care physicians)

<PHR Year Analysis>

7. Compare the average number of physicians and the average number of patients discharged per PHR each year. Which PHR has the highest average physician to average patients discharged ratio in each year? (Hint: you will need to calculate this ratio before running a command to compare across PHRs)

<MSA Border Analysis>

8. How many MSAs are in the border region?
9. Which MSA had the largest increase in patient discharges in 2011? What about 2012? (Hint: one way to do this would be to sort your data before printing)

<Improve one thing about your submission for assignment 4>

Most of you will have things you can improve on in your assignment 4 submission. Some examples are

- Improve computer efficiency by reducing number of data steps and combining
- Improve elegance of your program
- Improve readability of your program
- If you think your program is perfect in assignment 4, do something in a different way than you did in assignment 4.

<u>This should be reflect in the program you submitted, but also copy and paste that part of the code in the readme file, and comments the section in your code.</u>