Assignment 3

Submission format:

* Submit your code as a .sas file. The filename MUST have the following format:

LastnameFirstname\_Username\_AsssignmentNumber.sas.

In this file the problems must be listed in order. Each problem must also have a comment line that precedes it. For example, for question 2, use:

\*\*\* q2 \*\*\*;

Each *part* of a problem must also have a comment line that precedes it. For example for question 2b, use:

\*\*\* q2b \*\*\*;

This ensures that I and the graders can follow your work.

* Include your problem #’s in your solution (as a comment).
* The output of the code and SAS LOG is submitted in a Word document. You have to specify clearly which problem does the output of the code and the SAS LOG belong to.The filename MUST have the same format ( but with .doc extension). In the Word document, you should have problem #’s too.

One way of obtaining the LOG and the ouput:

Run your code then go to RESULTS tab. Click on Download results as a RTF file.

Then you can click on the LOG tab and copy the LOG of the run.

Save both the log and the output in a DOC file and name it :

LastnameFirstname\_Username\_AssignmentNumber\_output.doc

* The code should be written so that, except for file-directory locations, the code can be directly run on any computer with SAS or SAS University ed.
* Any text that is not code ( such as responses to a question) will be included in the .sas file as comment lines. (IMPORTANT). The word document is strictly for SAS output. Everything else should be in the .sas file.
* You must submit the HW to the correct myCourses dropbox.
* Duedate: 5pm Monday, Sep 30th, 2019.
* Make sure you have both files in the submission.

1. (2 pts) The file *tempArray.csv* contains the temperature in °F at 24 locations in an unheated barn on 5 consecutive winter days in upstate NY. Using SAS:
   1. Read in the data, and then convert the temperatures from °F to °C. Calculate the °C temperatures to the same number of decimal places as the °F temperatures. Keep both sets of temperatures.
   2. Add the variable Day to the SAS data set, so that Day = 1, 2, 3, 4, 5 for the first, second, …, fifth observations, respectively. (Do this using SAS code, not reading in new data.)
   3. The first day of the test was January 23, 2012. Add the variable Date to the SAS data set and format it to look like this: 01/23/2012. (Do this using SAS code, not reading in new data.)
   4. Save the results into a permanent SAS data set
   5. Print out the results, in this order: Day, Date, 24 temps in °F, 24 temps in °C. Include an appropriate title. Do not include observation numbers.
   6. Do this again, but in this order: Day, Date, 1st temp in °F, 1st temp in °C, 2nd temp in °F, 2nd temp in °C, …, 24th temp in °F, 24th temp in °C.

Submit your code and your two printouts (only a fraction if you want—just modify PROC PRINT); please avoid unnecessarily ugly formatting. Also, comment on the ease of doing step e. vs. doing step f.

1. (3 pts) The file *top-1000-baby-boy-names.txt* contains the names and rankings of the most common 1000 baby-boy names in the US, as reported by the Social Security Administration (for 2012). (The html source of that data is contained at the beginning of that file.) Using SAS:
   1. Read in the 1000 names and ranks (where rank 1 is the most popular name). You will need to look at the file to see how to read these values in. (I obtained this file by saving a pdf file from a web page and then converting it into a text file. Aside from changing a line or two to make your job easier, this is how the resulting text file looked.) A hint:

* Here is a “trick” for reading in the ranks. The comma informat is often used for reading in money, including how money is written in accounting, where "(50,000)" means -50000.

Also, when you look at the text file (you do always look at the text file in a text editor before reading it in SAS or R, don’t you?), you will find that the longest record, record #317, is 415 character long. This is much longer than older data files would normally be—the maximum record length a long time ago was only 80 characters (same as 80 bytes). Somewhat remarkably, before SAS 9.4, SAS assumed that no records would be over 256 bytes in length. With this information, look at the SAS log after you read in this data set. (You do always look at the log when you run a DATA or PROC step, don’t you?). You will see that the log will include this information:

...LRECL=32767...

The maximum record length was 415

* 1. Table the frequencies of names by the first letter of the name. (So, your table should contain at most 26 categories.)
  2. Do this again, but now produce the table so that:
     1. The frequencies are listed in descending frequency order instead of alphabetical order. You will probably need to learn more about PROC FREQ to do this. (Please do not ask how to do this or post the answer to how to do this in the Discussion module of myCourses.)
     2. Only the frequencies are listed. (Same note as in last step—figure this out on your own.)

Submit your code and the output of the two tables.

1. (2 pts) (Same problem, continued).
   1. For the most popular first letter, make a frequency table of the first two letters of the name. For example, if H is the most popular first letter, make a frequency table of Ha, He, Hi, and all the other “Hx” combinations. If you were able to do the last step correctly (descending frequency order, frequencies only), do this step the same way. (For this problem and the next, it will be OK to just look at the output of an earlier answer to decide how to help write the code. In a more general setting, it would be important to try to figure out how to write code that would *extract* the answer directly and use that result in the code instead of looking at the output to decide how to write the code.)
   2. Now, for the two most popular first two letters, print out a listing, sorted by the first two letters and then by rank, of all such names. For example, if the two most popular first two letters are Hi and then Ha, then print all the names and ranks for those names, listing the highest ranked Hi first, and the lowest ranked Ha last.  
       Make a separate listing (BY statement) for Hi, e.g. and then Ha. Show only the names and ranks in the listing (not the observation). As always, it is good to use a descriptive title.
   3. (0.5 ***Bonus***). Using code, see whether the ranks you read in are indeed 1, 2, 3, …, 999, 1000. (After sorting, of course.) If so, write a note to the SAS log that "Ranks are from 1 to 1000"; if not, write a note to the SAS log each time the ranks were not what they were supposed to be. (Same note as earlier—figure this out on your own.)

* This is a simple example of *Data Checking*, which is an important part of almost every good statistical analysis.

For the non-bonus work, submit your code, your frequency table, and enough of the listing to show that did it correctly. For the bonus, submit your code and the results of the SAS log.

1. (3 pts) Look again at the data set *preprocessedgooglestatstab.txt* from week 2. Using SAS:
   1. (0 pts—just to get you started) Read in the data again to a SAS data set.
   2. Create a table that uses day of the week as the row dimension and the following three summaries as the column dimension: average number of impressions, average number of clicks, and average revenue. Use appropriate titles, formats, and labels so that the report looks professional. Display the results so that an appropriate number of digits is shown.
   3. (0.5 ***Bonus***) Figure out how to arrange the output so that the weekdays are listed in the table in their natural time order, starting with Monday and ending with Sunday. (Same note as earlier—figure this out on your own.) (Hint: ORDER option in PROC TABULATE.)

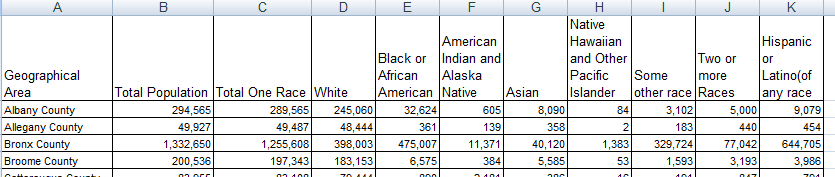
For the non-bonus work, submit your code and your summary-table output. For the bonus, submit your code and your summary-table output.

The following HW assignments are based on extracts from two databases on the internet a number of years ago. The voter registration data was gathered from <http://www.elections.state.ny.us/voting/voting.htm> (a site whose name has since changed) and the population statistics were gathered from the census bureau at [www.census.gov](http://www.census.gov). The data had been stored as three worksheets in an Excel Workbook. The population of each county in New York was in the NYPopulation worksheet; the voter registration numbers are stored in the other two worksheets, NYCVoters for counties in New York City, and NonNYCVoters for the counties not in New York City. These worksheets have been distributed as csv files.

A snapshot of the first few rows of each file is also shown (in the Excel format and, for the first file, the CSV file itself, with indents to show line wraps).

Please decide whether you will need to include the DSD option or the MISSOVER option on the INFILE statement.

* nypopulation.csv

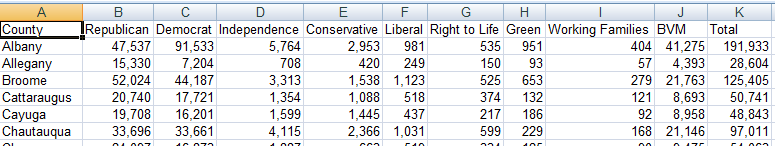


Geographical Area,Total Population,Total One Race,White,Black or African American,American Indian and Alaska Native,Asian,Native Hawaiian and Other Pacific Islander,Some other race,Two or more Races,Hispanic or Latino(of any race

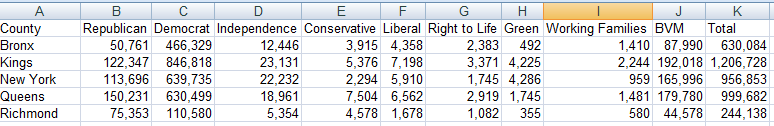
Albany County,"294,565","289,565","245,060","32,624",605,"8,090",84,"3,102","5,000","9,079"

Allegany County,"49,927","49,487","48,444",361,139,358,2,183,440,454

* nonnycvoters.csv



* nycvoters.csv



1. (5 + 1 Bonus) Do the following work in SAS.
   1. (1) Read in the data.
      1. Write a SAS program that reads in each the three files using DATA steps (not PROC Import) and stores the data described into a three different *permanent* SAS data sets. Use these names for the SAS data sets as the file names: nypop, nycvote, nonnycvote.
      2. You will need only the county and total population from the nypopulation file. (Just keep the variables that you need.)
      3. You will need only the county, number of republicans, number of democrats and total number registered from the other two files.
      4. Make sure you look at all three data sets and prepare them to be concatenated end-to-end and merged side-to-side. If you need to change any data to do this work, do so as minimally as possible.
      5. (SAS only) Include commas when you print out the population values.

Submit your code and a printout of the first 10 lines of each data set.

* 1. (2) First merge attempt.
     1. Write a SAS program that merges the nypop data set with the nonnycvote voter data set created in question (a.) and creates a new temporary SAS data set. Make sure that the counties merge properly. Include all the data here in which the merge was successful (inner join).
     2. Include in your code the ability to do this:
        1. Produce a listing (entire records) of any counties that are in the nypop data set but not the nonnycvote data set.
        2. Produce a listing (entire records) of any counties that are in the nonnycvote data set but not the nypop data set.

Submit your code, a printout of the first 10 lines of the merged data set, and a full printout of any data that did not match, as noted in (ii.).

* 1. (1) Write a SAS program that concatenates the nycvote and nonnycvote end to end and stores the result in a new temporary SAS data set called nyvote.

Submit your code.

* 1. (1) Second merge attempt.

1. Write a SAS program that merges the nypop data with the nyvote data set created in question (c.) and creates a new temporary SAS data set. Make sure that the counties merge properly.
2. Include in your code the ability to do this:
3. Produce a listing (entire records) of any counties that are in the nypop data set but not the nyvote data set
4. Produce a listing (entire records) of any counties that are in the nyvote data set but not the nypop data set

Submit your code, a printout of the first 10 lines of the merged data from question (i), and a full printout of any data that did not match, as noted in (ii.)

* 1. Bonus (1) From the merged data set in question (d.), calculate the percentage of the population that is registered to vote in each county. Create a report using the procedure of your choice that shows the county and percentage, sorted in descending order from highest to lowest percentage. Use an appropriate format for the percentage.

Submit your code and a printout of the first 10 lines.