Stat 604
Assignment 10 - SAS

OBJECTIVES: The purpose of this assignment is to practice creating variables conditionally and controlling the input and output of rows of data. Throughout this assignment we will also be concentrating on various techniques to improve the efficiency of our programs.

You should have all of the information you need to complete this assignment by viewing Lectures SAS01 through SAS06. This assignment will utilize almost every technique covered in lectures SAS05 and SAS06.

Perform each of the exercises listed below. To the extent you have been trained to control it, your output should match that in the PDF file posted on eCampus.

1. Begin your program with the required header, filename, and libname statements. As always, your program must include comments in the appropriate places. In addition to the comment boxes you have been using above data and proc steps, use single comment lines to identify the sub-steps that are being accomplished within the data step. This requirement will be in place for this and subsequent SAS assignments.
2. We will begin by using as input the **tabled1x2016** data set downloaded for the previous assignment. From this data set we want to create a "narrow" data set that is conducive to certain types of reporting. The resulting data set will contain multiple rows for each row in the original data set. The following steps describe how this temporary data set is to be created:
    a. The resulting data set will only contain the variables industry, state, month, year, and jobs.
    b. Write a SAS statement that will change the misspelled Industry TRADE, TRANSPORTATION, AND UITLITIES to TRADE, TRANSPORTATION, AND UTILITIES.
    c. Change the Industry names to proper case.
    d. The remainder of this data step will consist of blocks of code – one for each of the monthly variables in the original data set. The block of code for a specific month will only output a record if the value of that month for that observation is not missing. Create a month variable that contains the full name of the month that matches the variable being processed. Likewise, create a year variable with the matching year stored as text. Create a jobs variable that contains the number from the monthly variable being processed. Later in the semester we will look at a method to accomplish this that is more automated. For now, hard code the values of the month and year in the 13 different blocks as needed to create a new observation for each month.
3. Download the **bls_jobs1516** data set from eCampus to the homework data folder on your computer. From this data set we will create six temporary data sets in a **<u>single</u>** data step as efficiently as possible. The following steps describe how the 6 data sets are to be created:
    a. The variables rep_date and ann_chg will not be in any of the new data sets.
    b. Correct the spelling of TRADE, TRANSPORTATION, AND UITLITIES as you did in the previous step.
    c. Create a new variable labeled Average Jobs that will contain the average number of jobs on each observation. The value should display only one decimal place. Since we have not covered the functions that do descriptive statistics, we are going to compute the

average the old fashioned way by taking the sum of monthly values divided by the number of months. Use a variable list to refer to all the months August 2015 through August 2016 in a sum function as part of this calculation.

   d. Do not process any observations where the value of Average Jobs is missing. Some of you will correctly observe that, given the pattern of the data, we could accomplish this with another, more efficient test but we are going to base our logic on Average Jobs in order to practice using one of the techniques described in these lectures.
   e. We have observed in the **bls_jobs1516** data, based on Average Jobs, that almost 6% of our observations represent large markets of over 1000 jobs(in thousands). About 62% are medium markets with values between 100 and 1000. The remaining 32% of observations are small markets with values greater than 0 but less than 100. Use a series of conditional (if) statements to create a new variable labeled Market Size that will contain the values Large, Med., and Small according to the criteria described above. In addition to assigning the value to Market Size, each conditional statement should output the observation to the appropriate large, medium, or small data set. These three data sets will only have the variables Industry, State and Average Jobs. Construct the statements so the program executes as efficiently as possible. The third statement does not need a conditional test but should assign a value for and output observations that have not already been included in the previous two conditional statements.
   f. Use a SELECT group to create three additional data sets based on the value of Industry. Create a **government** data set of observations where the Industry is GOVERNMENT. This data set only contains variables for State, Average Jobs and Market Size. Create a **goods** data set of observations where the Industry is CONSTRUCTION or MANUFACTURING. The third data set contains **services** based on the remaining 5 values for Industry that appear in the input data. Specify these Industries in your conditional statement rather than allowing them to be a default at the end. The **goods** and **services** data sets will contain the variables for Industry, State, Average Jobs, and Market Size. Order the statements in your select group for the most efficient processing and include a statement so that the program will not generate an error should it encounter a missing value for Industry or a different value from those currently in the data.

4. Set up your PDF output file so that bookmarks are created but not shown by default.
5. Using two separate print procedures, print the first 50 and the last 50 observations of the data set created in step 2 above. Manually compute the numbers you need and hard code them into the appropriate options to get the desired results. Suppress the printing of observation numbers in this step. You are to use Title statements on the line above each proc print statement to identify the source of your output. This is the first Title statement you will use as an example since we haven't covered Titles yet:
   title "5a – First 50 Observations from Monthly Jobs Data Set";
   Use this pattern and replace the text inside the quotation to match the output on eCampus. (Beware of OFFICE quotation marks if you copy and paste.)
6. Print selected observations from each of the temporary data sets created from the **bls_jobs1516** data in step 3 above. Make sure labels are printed for each data set. Print observation numbers unless otherwise directed.
   a. Print the first 30 observations of the small data set.
   b. Print the first 30 observations of the medium data set.
   c. Print all the observations from the large data set.
   d. Print 30 observations beginning with observation 75 from the goods data set. Suppress the printing of observation numbers.

    e. Print the first 30 observations having a market size of Small from the services data set.
    f. Print all observations from the government data set.
7. In the SASHELP library is a view named VTABLE that contains information about all of the data sets in all of the libraries in your SAS session. A view can be accessed as if it were a SAS data set. You can use a where clause on a print procedure the same as on a data step. Use a print procedure to print the variables libname, memname, crdate, nobs,and nvar from vtable where the libname is WORK. Show the variable labels in your output. If you have written your program correctly and executed it from a fresh SAS session, the numerical values in your output should match that in the output shown on eCampus.
8. Convert the program and log to PDF files and submit them to WebAssign along with your SAS output.