Stat 604
Assignment 8 - SAS

This assignment reinforces the concepts covered in lectures SAS01 through SAS16. You will create and apply user defined formats. You will create data sets by reading in raw data files.

Download the files **election_codebook.txt**, **election_hist.csv** and **region6.dat** from the Assignment Data Files section on eCampus. Examine them carefully before writing your SAS program. If you are running Windows, the Notepad++ text editor program can be extremely useful for examining these files. If you are using a Mac, Text Wrangler is a similar program. If you have Microsoft Excel installed on your computer, it will probably open the CSV file with Excel by default. If you open the file in Excel, DO NOT save it when Excel closes or you may alter the format and make it unusable for the assignment. Files opened in this manner will appear as though they were Excel files. This can help you get a good visual image of the data but it is also a good idea to open the file in a text editor so you can see how the raw data is actually stored in the file. The election_codebook file provides definitions of the data in the CSV file. You will still need to compare the description with the layout in the data file and accommodate any differences that might exist. Perform each of the exercises listed below. To the extent you have been taught to control it, your output should match that in the PDF file posted on eCampus.

1. As always, your program must include comments in the appropriate places. Begin your program with the required header and filename statements. Use filename statements to reference the location of the two raw data files.
2. Create user defined formats that can be used to display the campaign type, party, and election status (results) as shown in the sample output. The first format will display the type as Incumbent, Challenger, or Open Seat when values are present in the raw data. The second format will be a numeric format that displays Democratic, Republican or Other Party when values are present. The third format will display election status as Won, Lost, or Runoff. Any other status values, even missing, are to be displayed as N/A.
3. Write a data step similar to the one shown in the lectures that converts the **election_hist.csv** file to two SAS datasets in the work library (more details will be provided below). This data step will include length, infile, input, and format statements. You will need to use an option for the infile statement to tell SAS to skip the column titles that are in the raw data. Unlike the data step option by the same name, this option does not use parenthesis. It is expected that you will get data errors but you should not have any other errors or warnings in your log except the warning that the limit of data errors has been reached.
    a. Set the length of all character variables before reading in the raw data. You will be able to determine the length of the variables based on the numbers that are provided in the election_codebook document. The columns must be in the order shown in the printed output without using any statements in the print procedure to control the order.
    b. Even though the end date values in the raw data do not contain any delimiters such as slashes or dashes, your informat will still need to be wide enough to accommodate both the date numbers and delimiters as if they exist.
    c. Apply your user defined formats to the appropriate variables. Apply a format to the end_date variable so that it is displayed as shown in the sample output.

d. Research the INTCK function in SAS Help. Use this function to create a new variable that contains the number of months between the end of the campaign and August 1, 2019. Hard code the August 1 date into the function.

e. If a row of raw data produces a data error, output that row to a temporary data set named **incomplete**. Since many of the data errors are caused by the end date, do not compute the months_since variable for these records and do not include it in this data set. This data set will contain 564 observations.

f. Write out the other rows to a temporary data set named **elections**. This data set will have 11,265 observations.

4. The **region6.dat** file contains fixed width data that will read in as formatted input. Neither file layout nor codebook was provided with this file. You will need to make certain deductions about this data (such as column widths) based on your examination of the data. Write a data step that converts this file to a SAS dataset in your work library. You may need to research SAS Help on the infile statement options to find the correct option to read in the data properly. Look for a section on Reading Past the End of a Line.
   For simplicity, read in the city-state-zip section as a single variable then use assignment statements to parse out the three variables. Make sure the resulting variables are no longer than they need to be for the data being read. Do not include the original city-state-zip variable in the output data set.
5. For this assignment the output file must be created with the pages in a landscape layout. The date is to only be displayed on the final section of the output. The page numbers of your PDF file should start with page number 2.
6. Print a sample of 20 records from the campaign year 2006. Create a temporary label for the "months since" variable. Use the appropriate enhancements to ensure that the label "Months Since End of Campaign" breaks as shown in the eCampus output. Note: The label may not break the same way in the default HTML output so be sure and check the PDF before you panic.
7. Print the descriptor portion of the **incomplete** data set.
8. Use a SAS procedure that will show extreme observations to check the data for candidate loans and other loans that might be unusual or invalid data values.
9. Use the SUMMARY procedure to create an output data set that contains the average candidate contribution and candidate loans grouped by the values in the Party variable.
10. Print out the data portion of the summary data set. Apply a format to the two analysis variables to show them as currency with no decimal places.
11. Print out the number of schools in each county (frequency of counties) based on your Region VI data set. Use an option that will also print out the number of counties in the data.
12. Make sure the date is printed at the top of the pages for this output step. Print out the data portion of the Region6 data set. Suppress the printing of observations.
13. At the end of your program include "housekeeping" statements to ensure that titles and footnotes do not get carried over to any subsequent output generated during this SAS session.
14. Convert the program and log to PDF files and submit them to eCampus along with your SAS output.