STSCI 5010 Lab 2

(Lab Date: 10/19/2018; Report Due: 11:59PM, 10/21/2018)

General instruction:

- Download four data files from the course website and save them in the lab2 library, which you are required to create. Keep all your files of this lab in the lab2 library.
- Use SAS comments for 1) marking the beginning each practice and its steps if any, 2) briefly documenting what you do in your SAS program at the beginning of each code segment and, 3) including your answer to each question at the end of your relevant code segment if required. If the comments are missing, up to 5 points are deducted.
- Specify a title for each output. If the titles are missing, up to 5 points are subtracted.
- Submit three files (combined with 7-zip or a similar tool, named last-name_first-name_lab2.7z) *to the course website*:
    - Your SAS code (in one single file), named last-name_first-name_ lab2_code_.sas.
    - Your SAS HTML output, named last-name_first-name_ lab2_HTML_output.mht. (5 points off if not included)
    - Your SAS log file, named last-name_first-name_ lab2_log.log. (5 points off if not included)

Practice One (40 points)

There are 4 variables and 12 observations in the raw file **Sales.txt**. The program below creates lab2.sales from the file that is referenced by the fileref **saledata**, adding a new variable named Total, which is the sum of Residential and Commercial.

Perform the following steps:

1. Write LIBNAME and FILENAME statements to assign the libref **lab2** and fileref **saledata**. (2 points)
2. Copy the program below and paste it into the code editing window. Do not run the program yet.

```
data lab2.sales;
      infile saledata;
      input LastName 1-7 Month 9-11 Residential 13-21
            Commercial 23-31;
      Total=residential+commercial;
run;
```

Modify this DATA step, so that you can test the program without reading any observations. Run your updated program and examine the SAS log for errors.

Is there any error message shown in the log? How many records and variables were read from the saledata? Why? (*Hint: use the **obs** option*) (4 points)

3. Modify the DATA step so that it reads all the observations, but does not create a data set. Resubmit the program and examine the log. What are the errors in the log referring to? What is the value of automatic variable _ERROR_ now? (4 points)
4. Fix the errors in the DATA step. How did you fix the errors? (4 points)
5. Edit the DATA step to create a new data set called **Sales** in lab2 library, and add a PROC PRINT step to display the new data set. Run the revised program and view the output. Now, how many records and variables were read from the raw data file? (4 points)
6. Do the following:
   A. Using the FREQ procedure, obtain the frequency output of Month. (5 points)
   B. Create a new dataset called **Salemonths** in the lab2 library. Identify any values of months other than JAN, FEB, and MAR. You should create a new variable called "Type" to indicate if a month value is correct or not. If correct, the variable is assigned a value of "correct." If not, using a Do group, assign a value of "incorrect" to Type and use a **Put** statement to write your own message indicating the error to SAS log, including the DATA step iteration number, the values of month and Type. Make sure that you can display the whole word "incorrect" in your SAS log. Display the Salemonths dataset. (17 points)

Practice Two (15 points)

1. Use the **empdata** dataset and sort it by variable **Location** and output the sorted dataset, **empdata_sorted**, to the lab2 library. (2 points)
2. Based on empdata_sorted, use a sum statement to calculate the total salary (named **Total_salary**) for each location. You are required to create a new dataset called **Location_total** in the lab2 library, which only contains two variables, Location and Total_salary. (10 points)
3. Display your Location_total dataset and calculate the grand total salary from all the locations. Use an appropriate format to display the salary values so that the dollar sign and comma(s) (at the thousand and million positions) are displayed. Do not print the observation numbers. (3 points)

Practice Three (20 points)

Achieve what is shown below. Using the datalines statement, first you create two datasets, **Table1** and **Table2**. Then, you use the match-merging method to

create the dataset, **All**. Display the All dataset exactly as shown including the title, All.

Table 1 + Table 2 =

Year	Var_X
1991	X1
1992	X2
1993	X3
1994	X4
1995	X5

Year	Var_Y
1991	Y1
1991	Y2
1993	Y3
1994	Y4
1995	Y5

All

Obs	Year	Var_X	Var_Y
1	1991	X1	Y1
2	1991	X1	Y2
3	1992	X2	
4	1993	X3	Y3
5	1994	X4	Y4
6	1995	X5	Y5

Practice Four (25 points)

Your final goal of this practice is to create a dataset, **heavy_female_patients**, based on two datasets, **demog** and **visit**. The Date variable in the demog dataset means date of birth, and the Date variable in the visit dataset means date of visit.

First, you create and then display a dataset called **all_matched** that does not contain any unmatched observations, based on the values of ID. This new dataset should include the following variables: ID, Sex, BirthDate, Visit, Weight and VisitDate. In your SAS log, display the number of data step iterations, the values of temporary variables (for example, `indemog and invisit`) of each data step iteration. If a data step outputs to the target data set (n is the number of data step iteration), for example, you display (with the same layout as below):

```
_N_=n indemog=1 invisit=1
    Data step n has output to the target data set.
```

Otherwise, for example, you display (with the same layout as below):

```
_N_=n indemog=1 invisit=0
    Data step n DOES NOT output to the target data set.
```

Second, you create and then display the heavy_female_patients dataset based on the all_matched dataset, only including those who are female and the body weights are heavier than 210 pounds.

Optional (but important): Practice all the SAS code covered in the class.