STAT 506: Homework 3

For these problems you will need to access the data in the PG1/data folder. Use the `libname` statement we learned to load this each time you work on your assignments. You should call it '`pg1`' to be consistent with the SAS materials.

I tried to *italicize* the parts where I expect you to actually show me something in your homework solutions if it is not obvious.

1. DATA step processing and filtering

Write a DATA step to do the following:

- Read in the table **pg1.eu_occ**.
- Add a WHERE statement to select only the stays that were reported in the year 2015. Use the `substr()` function. [Note that **YearMon** is a character column, and the first four characters represent the year.]
- Assign the COMMA10. format to the **Hotel**, **ShortStay**, and **Camp** columns.
- Save the new table as **eu_occ2015**, but exclude the columns **Geo** and **Country**.

Print the first 6 observations of **eu_occ2015**. *Show your code and the output.*

2. Creating New Columns

Write a DATA step to do the following:

- Read in the table **pg1.np_summary**.
- Create a new column named **SqMiles** by dividing the column **Acres** by 640.
- Create a new column named **CampersTotal** as the sum of **OtherCamping**, **TentCampers**, **RVCampers**, and **BackcountryCampers**.
- Format **SqMiles** to show one decimal place.
- Save the new table as **np_summary_update**, but only include the column **ParkName** and the new columns created above.

Print the first 10 observations of **np_summary_update**. *Show your code and the output.*

3. Using Conditional Processing to Re-Categorize and Clean Data

a. As we've seen previously, the table **pg1.np_summary** is using some inconsistent codes for the column **Type**. Create a frequency table for **Type**. *Show just your code.*

b. Write a DATA step to create a new table named **park_type** that includes everything from **pg1.np_summary**. Also use IF-THEN/ELSE statements to create a new character column named **ParkType** based on the value of **Type**:

- **Type** = "NP" → **ParkType** = "Park"
- **Type** = "NS" → **ParkType** = "Seashore"
- **Type** = "NM" → **ParkType** = "Monument"
- **Type** = "RVR" or "RIVERWAYS" → **ParkType** = "River"
- **Type** = "PRE", "NPRE", or "PRESERVE" → **ParkType** = "Preserve"

Show your code and the corresponding Log notes.

c. Create a frequency table for **ParkType**. *Show your code and the output.*

4. **Using Labels in PROC PRINT**

   a. Write a PROC CONTENTS step to display the descriptor portion of **pg1.eu_occ** to see the permanent labels assigned to the columns. *Show the relevant part of the output (the table that shows the labels).*

   b. Print the first 6 observations from **pg1.eu_occ**. All the columns should be displayed with their permanent labels, except for **YearMon**, which should have the temporarily assigned label "Time Period" displayed instead. *Show your code and output.*

5. **Two-Way Frequency Reports**

   Make a two-way frequency report for the columns **sex** and **birthdate** in **pg1.class_birthdate**.

   - Use **birthdate** as the column variable.
   - Use a format to group the values of **birthdate** by year instead of by individual date. If done properly, this should result in a table with 6 year columns.
   - Add the label "Year" to **birthdate**.
   - Add the titles "Class Overview" on the first line and "Birth Year versus Sex" on the third line.
   - Add *your* name as a footnote.
   - Use options in the TABLES statement to show only the frequencies and the column percentages in each cell.
   - Add code to clear the titles and footnote after the report is generated.

   Show your code and the output.

6. **Creating an Output Summary Table**

   a. Write a PROC MEANS step that will calculate summary statistics for the variable **hotel** in **pg1.eu_occ** using **country** as the class variable. Save the output as a new temporary table named **med_hotel** which includes the median values for the **hotel** variable as a variable named **MedianHotel**. Use the NOPRINT option. *Show your code and the corresponding Log notes.*

   b. Write a PROC SORT step to sort **med_hotel** by **MedianHotel** in descending order. If you didn't do the PROC MEANS step in a way that automatically removes the row that summarizes the entire table (the row with a blank **Country**), then filter out that row in this PROC SORT step. There should be 29 observations in **med_hotel** now. *Show your code and the corresponding Log notes.*

   c. Write a DATA step to update **med_hotel** by eliminating the columns **_TYPE_** and **_FREQ_**. In this step, also assign **MedianHotel** the permanent label "Median of Hotel Nights". *Show your code and the corresponding Log notes.*

   d. Finally, print the first 16 observations from **med_hotel** and display the labels for the variables. *Show your code and output.*