Homework 3

Due: Tuesday August 29th

Instructions: For the General Knowledge Questions, answer as succinctly as possible. For the Programming Assignment, show your code as well as a partial screenshot of your output.

<u>Knowledge Questions</u>

1. When using pointer control to read data in fixed columns, the _____ symbol moves the pointer <u>to</u> a specific column? This method is considered _____ pointer control.

2. When using pointer control to read data in fixed columns, the _____ symbol moves the pointer <u>over</u> a specific number of columns? This method is considered _____ pointer control.

3. When used during a data step, the _____ statement selects specific variables to include in the <u>output</u> SAS dataset, while the _____ option in the SET statement selects specific variables to be read in from the <u>input</u> SAS dataset.

4. Using Label and Format statements during a data step _____ assigns these attributes to variables, while using these statements during PROC PRINT _____ assigns these attributes to variables.

5. There are three statements that can be used to eliminate specific observations during a data step. These are the _____ statement, the _____ statement and the _____ statement.

<u>Programming Assignment</u>

1.Reading SAS Data Sets and Creating Variables

a. Use the **fltattnd** data set to create a temporary SAS data set named **work.bonus**.

b. Create a variable named **BonusAmt** that contains an annual bonus amount for each employee calculated as 12% of Salary.

c. Create a variable named **AnnivMo** that contains the employment month for each employee. Hint: Determine the month portion of the employee's date of hire (**HireDate**).

d. The bonus dataset should contain only the variables **EmpID, Salary, BonusAmt, HireDate, and AnnivMo.**

e. Use the PRINT procedure to display the data portion of the bonus data set. Display the values of **Salary** and **BonusAmt** with dollar signs, commas, and no decimal places.

2.Creating Variables Using Conditional Execution

a. Use the **fltattnd** data set to create a temporary SAS data set named **work.raises**. Create a variable named **Increase** that contains an annual salary increase amount for each employee. Calculate the Increase values as shown below:

- 15% of Salary when **JobCode**='FLTAT1'
- 10% of Salary when **JobCode**='FLTAT2'
- 5% of Salary when **JobCode**='FLTAT3'

b. Create a variable named **NewSal** that contains the new annual salary for each employee by adding the raise to the original salary. The **raises** data set should contain only the variables **EmpID, Salary, Increase,** and **NewSal**.

c. Use the PRINT procedure to display the data portion of the raises data set. Display the values of **Salary, Increase**, and **NewSal** with dollar signs, commas, and two decimal places.

3. Concatenating SAS Data Sets

The goal is to create a second-quarter data set for International Airlines' Vienna hub. Write a data step that combines target information for April, May, and June into one data set. This data is currently stored in separate data sets by month as follows:

- **aprtarget**
- **maytarget**
- **juntarget**

a. As a first step, <u>browse the descriptor portion</u> of each data set to determine the number of observations, as well as the number of variables and their attributes. How many observations does each data set contain?

- **aprtarget** _____
- **maytarget** _____
- **juntarget** _____

What are the names of the variables in each data set?

aprtarget

_____

maytarget

_____

juntarget

_____

b. Now write a data step to concatenate the three data sets and create a new data set called **work.q2vienna**. Use the RENAME= option of the Set statement to rename any variables necessary to combine the datasets.

c. Browse the SAS log. There should be no warning or error messages.

- How many observations are written to the new data set?

· How many variables does the new data set contain?

d. Submit a PROC PRINT step to verify the data.

e. Now modify the DATA step to create two new variables: **TotalTar** and **TotalRev**.
- · **TotalTar** is the total targeted number of economy and first class passengers.
- · **TotalRev** is the total revenue expected from economy and first class passengers.

Keep only the variables **FlightID, Destination, Date, TotalTar**, and **TotalRev**.
Submit a PROC PRINT step to verify the data.

4.Merging SAS Data Sets
The weather in Birmingham, Alabama on December 15, 1999, might have caused some customers to alter their shipping plans. Investigate how much cargo revenue was lost on all flights out of Birmingham by comparing the targeted revenue with the actual revenue.

a. Sort the data set **target121999** into a temporary data set called **work.sort_b**. Sort by the variable **FlightID**. Use the WHERE statement to create a subset for Birmingham on December 15, 1999.
where Date='15dec1999'd and Origin='BHM';

b. Sort the data set **sales121999** into a temporary data set called **sort_s**. Sort by the variable **FlightID**. Use the WHERE statement to create a subset for Birmingham on December 15,1999.
where Date='15dec1999'd and Origin='BHM';

c. Create a new temporary data set called **compare** by merging the **sort_b** and **sort_s** data sets by the variable **FlightID**. Subtract **CargoRev** from **CargoTarRev** to create a new variable called **LostCargoRev**.

d. Print the merged data set **compare** (print only the variables **CargoTarRev, CargoRev**, and **LostCargoRev**) and label the **LostCargoRev** variable. Format the **LostCargoRev** variable with a dollar sign and two decimal digits.