STAT 6430: Statistical Computing for Data Science

Problem Set 3

General instructions: Each problem of this assignment asks you to write one or more SAS programs, and generate one or more new data sets. Write your solutions in SAS's Enhanced Editor Window, placing all of your programs in the same document, in the correct order, and using comments to indicate which section of code corresponds to which problem and part of a problem. Declare any LIBNAME statements near the top of your document, before any of your solutions.

What to turn in: Turn in both electronic and hard copies of your program (as a ".sas" file), and electronic copies SAS data sets you are required to create (as ".sas7bdat" files). Put all of your files into a single compressed (.zip) folder, and upload it to the "Assignments" tool in UVaCollab. (Click the assignment "Problem Set 3," and follow the instructions.) Name the folder in such a way that identifies you by name or by computing id. For instance, I might name it "ProblemSet2DJS4Y.zip" or "PS2DanSpitzner.zip." Also print out a hard copy of your program and turn it in. I only need your program. Do not print out or turn in anything that appears in the Results Viewer or Log Window when you run your program.

Problem 1: Download the files "PS3Prob1A.txt" and "PS3Prob1C.sas7bdat," which accompany this assignment on the course web page, and place it into the directory "C:\LocalData."

Part A: The file "PS3Prob1A.txt" lists survey responses of thirty-seven subjects, with each record associated with the responses of an individual subject. The data fields store subject ID numbers, followed by gender and date of birth, followed by the responses of twelve questions: the first four are on a Likert scale (labeled 1-5), the next three are dollar values, and the last five are yes/no, but coded numerically as 0 = "No" and 1= "Yes." The data fields are arranged in strict columns, with spaces in between all but the Likert and yes/no data fields, and with conventional formatting applied to date of birth and dollar values.

Use the DATA Step to read all of the data in "PS3Prob1A.txt" into a SAS Data Set. Read the data into variables named **`subjID`**, **`gender`**, **`dob`**, **`qA1-qA4`**, **`qB1-qB3`**, and **`qC1-qC5`**, respective to the given ordering. In the same DATA Step, use ASSIGNMENT statements to create three additional variables: **`totB`**, which stores the total of the values in **`qB1-qB3`**; **`pctC`**, which stores the proportion of "Yes" responses among **`qC1-qC5`**; and **`adultyrs`**, which stores the number of years since a subject's 18th birthday, as of August 24, 2010. (For instance, a subject turning 18 on July 31, 2010 would be 24 days past that date on August 24, 2010, and this translates to a value for **`adultyrs`** of 24/365.25 = 0.0657 years.)

Use PROC PRINT to list out (only) the data in the variables **`gender`**, **`adultyrs`**, **`totB`**, and **`pctC`**. Use a FORMAT statement to assign formats such that **`adultyrs`** is written with no decimal values (but still stored in the SAS Data Set with its decimal values), **totB** is written in standard dollar-currency format, with two decimal values, and **pctB** is written in the format of a percentage, with no decimal values. (For the latter, you should apply the PERCENTw.d format, which we have not discussed in class, but uses the exact parallel syntax as the formats w.d, DOLLARw.d, and COMMAw.d.)

In addition, declare **subjID** as the ID variable in PROC PRINT. Use a LABEL statement to assign descriptive labels (of your invention) to each of the variables **gender**, **adultyrs**, **totB**, and **pctC**, and specify that they are to appear as headings in the PROC PRINT output.

Part B: Repeat **Part A**, but this time skip the PROC PRINT step and instead write the output data to an eternal file named “PS3Prob1Bout.txt.” That is, write values of the variables **subjID**, **gender**, **adultyrs**, **totB**, and **pctC** to the external file. (Notice this includes **subjID**, which was the ID variable in **Part A**.) Apply the same formatting to **adultyrs**, **totB**, and **pctC** as you did in **Part A**, and insert at least one space between each data field written. Carry out this task without creating a SAS Data Set.

Part C: Repeat **Part B**, but this time read the data from the SAS Data Set stored in the file “PS3Prob1C.sas7bdat,” and write it to the data to an eternal file named “PS3Prob1Cout.txt.”

Problem 2: Download the file “PS3Prob2.txt,” which accompanies this assignment on the course web page, and place it into the directory “C:\LocalData.”

The file lists new-member information at a health club. The first field is member ID, the second is gender, the third is date of birth, the fourth is height, and the last is weight. The data fields do not appear in straight columns.

Read in the data from “PS3Prob2.txt,” and store it in a SAS Data Set. Be sure the padded zeros of the member ID numbers are kept, and that dates-of-birth are stored as SAS Dates (the number of days since January 1, 1960).

Use a PROC PRINT statement to list the raw data. Then insert a FORMAT statement in PROC PRINT, specifying that the dates should be listed using the DATE9. format, and run PROC PRINT again.

Run PROC CONTENTS, declaring the data set you’ve created, and glance at the “Variables and Attributes Table.”

Next, move the FORMAT statement in PROC PRINT to the DATA Step. Run PROC CONTENTS again. What has changed in the “Variables and Attributes” table?

Problem 3: Download the file “PS3Prob3.txt,” which accompanies this assignment on the course web page, and place it into the directory “C:\LocalData.”

The file lists responses to a survey of five questions, with each record storing the measurements of an individual respondent. The first field records the age of the respondent, the second records his or her gender, and the last five list his or her responses to the questions. Observe that all the fields appear in straight columns, and there are no delimiters between the last five.

Use the LIBNAME statement to declare a SAS Library. Read in the data from “PS3Prob3.txt” and store it in a permanent SAS Data Set in the library.

Close the SAS session. Then, start a new SAS Session, and, without again reading in the data from the file “PS3Prob3.txt,” write a SAS program that calculates mean age of the respondents and the frequencies of responses to each question.

Again without re-reading in the data from the file “PS3Prob3.txt,” write a SAS Program that writes to an external file the age, gender, and question responses of each female subject who answered “3” on Question 5. Formatting should be such that the output data-file looks like this:

```
age=30 gender=F Responses = 1 1 1 2 3
age=62 gender=F Responses = 3 3 3 3 3
```

Problem 4: Download file “PS3Prob4.sas7bdat,” which accompany this assignment on the course web page, and place it into the directory “C:\LocalData.”

This file lists a portion of responses from a survey of voters. The first data field is the age of the respondent, the second is his or her party affiliation, and the remaining four are responses to individual statements, on a Likert scale.

Use PROC FORMAT to create user-defined formats as follows:

- The age data are to be formatted into groups that are defined as: 0 to under 30, 30 to under 50, 50 to under 70, and 70 or older.
- Party affiliations are to be written “Left” for L, “Right” for R, “Center” for C, and “Independent” for I, and any other party should be written “Other.” Be sure to take into account the possibility of missing values.
- The format for the response-values in the last four data fields should reflect that they are on a Likert scale: 1 is “Strongly Disagree,” 2 is “Disagree,” 3 is “No Opinion,” 4 is “Agree,” and 5 is “Strongly Agree.”

Rather than using the variable names supplied in the SAS Data Set descriptive labels are preferred for the last four data fields, to reflect the statement to which the response was made. The first variable is to be labeled, “Congressperson X is doing a good job;” the second, “Institution Y is doing a good job;” the third “Plan Z is the correct solution for Issue A;” and the last “The country is on track regarding Issue B.”

Use PROC PRINT to list out the voter data, formatted as above, using a LABEL statement to associate the question variables to the descriptions above. (You will also need to declare a LABEL option in PROC PRINT.) Do not create a new SAS Data Set (or carry out any DATA Step).

Part B: Use PROC FORMAT to create two new formats:

- Create a crude version of the Likert scale format that groups the responses 1 and 2 as “General Disagreement,” 4 and 5 as “General Agreement,” and leaves 3 as “No Opinion.”
- Create a crude version of the party affiliation format that groups the “Center” and “Independent” affiliations together with those previously formatted as “Other.”

Use PROC FREQ to calculate frequencies of the party affiliations and statement-response values, using the cruder format groupings (not those defined in **Part A**).

Problem 5: The SAS Data Set **bicycles** lists sales information on several bicycle models. To access the data set, save the file “bicycles.sas7bdat” in a folder, of your choice, that you have associated with a SAS Library. You may want (but are not required) to run PROC PRINT and PROC CONTENTS to view the data and examine its attributes.

Part A: “Split” the **bicycles** data set into two (permanent) SAS Data Set, one named **Mountain_USA** and the other **Road_France**. The former is to contain all observations for which the variable values are such that COUNTRY=’USA’ and MODEL=’Mountain Bike’; the latter is such that COUNTRY=’France’ and MODEL=’Road Bike’. (Note this splits just a subset of the full data set; some observations in **bicycle** will neither be assigned to **Mountain_USA** nor **Road_France**.) You will be awarded more points for creating both of these data sets in a single DATA Step.

Part B: Run the following SAS program, which creates a temporary SAS Data Set storing “markup” values bicycles of each manufacturer.

```
data markup;
input manuf : $10. Markup;
datalines;
Cannondale 1.05
Trek 1.07
;
run;
```

Combine this data set with the SAS Data Set **bicycles** in such a way that each observation is assigned a value of the variable MARKUP, with assignments respecting by the values of the variable MANUF. Name this combined data set **markup_prices**. Also, create a variable in **markup_prices**, named **newtotal**, that stores for each observation the product of the values of the variables TOTALSALES and MARKUP.

Problem 6: The SAS Data Set **inventory** lists pricing information on items of an unspecified type; the SAS Data Set **newproducts** lists the same information for new items of the same type. The SAS Data Set **purchase** lists the models of these items that were purchased, as well as purchase quantities. To access these data sets, save the files “inventory.sas7bdat,” “newproducts.sas7bdat,” and “purchase.sas7bdat,” in a folder, of your choice, that you have associated with a SAS Library. You may want (but are not required) to run PROC PRINT and PROC CONTENTS one each data set to examine their contents and attributes.

Part A: Sort each of the data sets **`inventory`** and **`newproducts`** by the values of the variable MODEL. Then create a new data set, named **`updated`**, that stores all of the observations of both data sets, interleaving them so observations are arranged in order according to the variable MODEL.

Part B: Merge the data sets **`inventory`** and **`purchase`** by the values of the variable MODEL, but then “split” the combined data set into data sets, called **`pur_price`** and **`unpurchased`**, respectively; the former is to store the observations whose values of the variable MODEL appear in both the data sets **`inventory`** and **`purchase`**, while the latter is to store the observations for which such values are found only in **`inventory`**. Drop any variables in **`unpurchased`** whose every value is missing, and include in the data set **`pur_price`** a variable, named **`totcost`**, that stores for each observation the product of the values of the variables QUANTITY and PRICE. You will be awarded more points for creating both of these data sets in a single DATA Step.

Part C: Run the following SAS program, which creates a temporary SAS Data Set that stores new prices for two of item models stored in the data set **`inventory`**.

```
data changes;
input Model : $8. price;
datalines;
M567 25.95
X999 35.99
;
run;
```

Combine this data set “side-by-side” with the SAS Data Set **`inventory`** in such a way that the values of the variable PRICE are those of the data set **`changes`** (created above) for observations whose values of the variable MODEL appear in both data sets (**`changes`** and **`inventory`**), but are those of the data set **`inventory`** for observations whose MODEL values appear only in **`inventory`**. Call this combined data set **`newprices`**. You will be awarded more points for creating this data set in a single DATA Step.

Problem 7: The SAS Data Set **`monthsales`** stores a month identifiers (in the variable MONTH) and monthly total sales (in the variable SALES) for a firm over one year. To access the data set, save the file “monthsales.sas7bdat” in a folder, of your choice, that you have associated with a SAS Library. You may want (but are not required) to run PROC PRINT and PROC CONTENTS to view the data and examine its attributes.

Create a new (permanent) data set, named **`newsales`**, that is identical to **`monthsales`** but includes another variable SUMSALES which stores the cumulative total sales for the year up to the current month, ignoring missing values.