STAT 5430: Statistical Computing for Data Science

Problem Set 2

General instructions: Each problem of this assignment asks you to write one or more SAS programs. 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.

Some of your programs will read data from an external file. As a basic ground rule of the assignment, nowhere in your code should you declare the location of an external file directly in an INFILE statement. Rather, near the top of your document (before any of your solutions), declare a sequence of FILENAME statements which create file-reference associations with the external directories and filenames. Subsequently, in any INFILE statement refer to the file-reference instead of the external file.

What to turn in: Turn in both electronic and hard copies of your solutions. Save your solutions in a single document and save it as SAS file (*i.e.*, a file with a .sas extension), and name the file in such a way that identifies you by name or by computing id. For instance, I might name the file "ProblemSet2DJS4Y.sas" or "PS2DanSpitzner.sas." If you have completed the problems correctly, it should be possible to load the document into the Enhanced Editor, submit it as a single batch of code, and it should run with no errors (assuming that the required external files are stored in the correct directory). Upload the SAS file using the "Assignments" tool in UVaCollab. (Click the assignment "Problem Set 2," and follow the instructions.) Also print out a hard copy of your file and turn it in. I only need your program. Do not print out or turn in anything that appears in the Output or Log Window when you run your programs.

Problem 1: Download the file "PS2Prob1.txt," which accompanies this assignment on the course web page, and place it into the directory "C:\LocalData." The file contains information on 249 subjects who participated in a physiology experiment where they responded to various questions (on a scale of 1-5), and were measured for the time it took to complete a simple task under controlled conditions. Each record corresponds to the measurements of an individual subject. Data fields are delimited by spaces, and are suitable to be read in using the basic syntax for "list" input. They are described, in order, as follows, together with variable names that you are to use when you read in the data.

Field	Description	Variable name	Field	Description	Variable name
1	Subject ID	**ID**	10	Pre-question 5	**Pre5**
2	Gender	**Gender**	11	Experiment class	**Expt**
3	Weight	**Weight**	12	Time	**Time**
4	Age	**Age**	13	Post-question 1	**Post1**
5	Resting pulse rate	**Pulse**	14	Post-question 2	**Post2**
6	Pre-question 1	**Pre1**	15	Post-question 3	**Post3**
7	Pre-question 2	**Pre2**	16	Post-question 4	**Post4**

8	Pre-question 3	**Pre3**	17	Post-question 5	**Post5**
9	Pre-question 4	**Pre4**			

Part A: Create SAS Data Set named **expt** by reading in the file using "list" input; assign the variable names listed above, but do not store all of the data in the SAS Data Set: store only the variables **ID**, **Gender**, **Pre5**, and **Time**, and furthermore throw out the last 15 observations and any for which the answer to **Pre5** is 1, 2, or 3. The variables **ID**, **Gender**, and **Pre5** are to be stored as character type with **ID** having length 7 and the others length 1. The variable **Time** is to be numeric. Use PROC PRINT to list all of the data stored in **expt**.

Part B: Repeat **Part A**, but this time store all of the data listed in the file (all variables and all observations) in the SAS Data Set **expt**. The variables **ID**, **Gender**, **Pre1–Pre5**, **Expt**, and **Post1–Post5** are to be stored as character type with **ID** having length 7 and the others length 1. All other variables are to be stored as numeric type.

Then use PROC PRINT, together with a certain syntax that we discussed in class, to generate a listing of just the variables **ID**, **Gender**, **Pre5**, and **Time**, with the last 15 observations, as well as any for which the answer to **Pre5** is 1, 2, or 3, omitted. Do this without declaring a VAR statement in your PROC PRINT Step.

Problem 2: A formula for predicting the "goodput" (y) of a computer network from the "pause time" (x_1) and average "node speed" (x_2) is

$$y = 96.0240 - 1.8245x_1 + 0.5652x_2 + 0.0247x_1x_2 + 0.0140x_1^2 - 0.0118x_2^2.$$

Use the DATALINES statement in a DATA Step to read in all possible pairs of values (x_1, x_2) for $x_1 = 5, 20, 40$ and $x_2 = 10, 30, 50$, storing the individual values in variables named **x1** and **x2**, respectively. Create a variable **y** that stores the results of the prediction formula above, and store everything in a SAS Data Set named **preds**. Use PROC PRINT to list out the data in **preds**.

Problem 3 (Based on Problems 2 and 4 from Chapter 3 of the textbook): Download the file "PS2Prob3.csv," which accompanies this assignment on the course web page, and place it into the directory "C:\LocalData." The file lists information about political leanings of individuals, and is in a standard "comma-separated values" format. In each record the first data field lists a state abbreviation, the second lists a political label, and the third lists the age of the individual.

Create a SAS Data Set named **vote** which reads this information into variables named **state**, **party**, and **age**, respecting the order of the file. The first two are to be specified as character and the last as numeric.

Problem 4 (Based on Problem 6 from Chapter 3 of the textbook): Download the file "PS2Prob4.txt," which accompanies this assignment on the course web page, and place it into the directory "C:\LocalData." The file contains information about accounts in a bank, with each record corresponding to an individual account. The data fields are

arranged in strict columns, with the name of the account holder appearing in columns 1-15, the account number in columns 16-20, the account balance in columns 21-26, and the interest rate in columns 27-30.

For each part below, your task is to create a temporary SAS Data Set named **bank**, using a DATA Step to read this information into variables named **name**, **acct**, **balance**, and **rate**, respecting the order of the file. The first two variables are to be of character type and the last two are to be numeric. You should also store in your SAS Data Set another variable named **interest**, whose values are calculated by multiplying **balance** and **rate** and then dividing by 100. Use PROC PRINT to list out the data in **bank**.

Part A: Carry out the task using the syntax of "column" input.

Part B: Carry out the task using the syntax of "formatted" input.

Problem 5: Repeat **Problem 1**, **Part A**, but this time read in the data using "column" input, and use that syntax to select which variables are stored in the SAS Data Set. Do not use a DROP or KEEP statement.

Problem 6 (Based on Problem 10 from Chapter 3 of the textbook): Download the file "PS2Prob6.txt," which accompanies this assignment on the course web page, and place it into the directory "C:\LocalData." The file contains information about stock prices of companies, with each record corresponding to an individual company. The data fields are arranged in strict columns, according to the following layout

Field	Description	Starting Column	Field width (cols)	Formatting
1	Stock symbol	1	4	Character
2	Purchase date	5	10	mm/dd/yyyy
3	Purchase price	15	6	Dollar currency
4	Number of shares	21	4	Numeric
5	Selling date	25	10	mm/dd/yyyy
6	Selling price	35	6	Dollar currency

(Observe that the type of formatting is indicated in the last column.)

Create a SAS Data Set named **stocks**, which to reads this information into variables named **stock**, **purdate**, **purprice**, **number**, **selldate**, and **sellprice**, respecting the order of the file. The first variable is to be of character type and all of the last five are to be numeric (with dates stored as the number of days since January 1, 1960).

In addition, store in your SAS Data Set another variable named **totalpur**, whose values are calculated by multiplying **number** and **purprice**, another one named

totalsell, whose values are calculated by multiplying **number** and **sellprice**, and another named **profit**, whose values are calculated by subtracting **totalpur** from **totalsell**. Use PROC PRINT to list out the data in **stocks**.

Problem 7 (Based on Problem 11 from Chapter 3 of the textbook): Download the files "PS2Prob7A.txt," "PS2Prob7B.txt," "PS2Prob7C.txt," and "PS2Prob7D.txt," which accompany this assignment on the course web page, and place them into the directory "C:\LocalData." Each file contains the same information about employees at a company, but in a different format than the others.

The data fields are employee identifications number, employee name, department, date of hire, and annual salary, in that order. These are to be read into variables named **ID**, **name**, **dept**, **datehire**, and **salary**, respectively. The first three variables are to be of character type, and the last two of numeric type, for which **datehire** is to store values translated to the number of days since January 1, 1960. The task of each part below is to read these data into a SAS Data Set named **employee**, but from a different file, subject to its specific formatting.

The data set created in each part below should be identical to that created in every other part.

Part A: The formatting of "PS2Prob7A.csv" follows that of "comma-separated values," but with formatted date and dollar values. Create the SAS Data Set **emplA** by reading data from this file.

Part B: The formatting of "PS2Prob7B.txt" is the same as that of **Part A**, except the delimiter is a dollar sign ($), not a comma. Create the SAS Data Set **emplB** by reading data from this file.

Part C: The formatting of "PS2Prob7C.txt" uses asterisks (*) to delimit the data, following the convention that two in a row indicate a missing value in between. Create the SAS Data Set **emplC** by reading data from this file.