STAT 6430: Statistical Computing for Data Science

Problem Set 1

What to turn in: Submit your solutions to Problems 2, 3, 4 and 6 on a separate sheet, neatly handwritten or typed, and clearly indicating your final answers to each problem. Problems 1, 5, and 7 each ask you to write a SAS program, and you should turn in both electronic and hard copies of those programs. Write your solutions to those problems in SAS's Enhanced Editor Window, placing all three in the same document, in the correct order, and using comments to indicate which section of code corresponds to which problem. Save the document as a SAS file (*i.e.*, a file with a .sas extension), named in such a way that identifies you by name or by computing id. For instance, I might name the file "ProblemSet1DJS4Y.sas" or "PS1DanSpitzner.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 1," and follow the instructions.) Also print out a hard copy of your file and turn it in with your solution to Problems 2, 3, 4 and 6. You do not need to turn in anything that appears in the Output Window, Log Window, or Results Viewer when you run your programs.

Problem 1: Download the file "PS1Prob1.txt," which accompanies this assignment on the course web page, and place it into the directory "C:\LocalData." Then select exactly three of the blocks of code that appear below which, when put together and submitted in the Enhanced Editor Window, will (*i.*) produce in the Results Viewer the output listing shown below, and (*ii.*) execute with no error messages listed in the Log Window.

The blocks of code are:

`Data Prob1Data;`
`Infile 'C:\LocalData\PS1Prob1.txt';`
`Input obj hgt wid;`
`Run;`
		`Data PS1Prob1;`
		`Infile 'C:\LocalData\Prob1Data.txt';`
		`Input obj hgt wid;`
		`Run;`

`Title 'Problem Set 1';`
`Title 'Problem 1';`
`Options nocenter;`
	`Proc print;`
	`Var obj $ wid hgt;`
	`Run;`
		`Title1 'Problem Set 1';`
		`Title2 'Problem 1';`
		`Options nonumber;`

`Proc contents;`
`Var obj wid hgt;`
`Run;`
	`Title1 'Problem Set 1';`
	`Title2 'Problem 1';`
	`Options nocenter;`
		`Proc print`
		`Var obj wid hgt;`
		`Run;`

`Title 'Problem Set 1';`
`Title 'Problem 1';`
`Options nonumber;`
	`Proc print`
	`Var obj $ wid hgt;`
	`Run;`
		`Proc print;`
		`Var obj wid hgt;`
		`Run;`

`Data Prob1Data;`
`Infile 'C:\LocalData\PS1Prob1.txt';`
`Input obj $ hgt wid;`
`Run;`
		`Data PS1Prob1;`
		`Infile 'C:\LocalData\Prob1Data.txt';`
		`Input obj $ hgt wid;`
		`Run;`

The desired output listing is:

```
Problem Set 1
Problem 1

Obs     obj      wid     hgt

 1      sqre      45      45
 2      rect      90      45
 3      bar       90       5
 4      pole       5      90
```

Problem 2: Classify each of the following items as valid or invalid for use as a variable name:

A. **`Hgt-Lbs`**
B. **`Xyz123abc987`**
C. **`789cba321zyx`**
D. **`FieldData`**
E. **`field-data`**
F. **`fIeLdDaTa`**
G. **`Data_2006`**
H. **`2006Data`**
I. **`Data2005-2006`**
J. **`Data06`**
K. **`LastWeek'sData`**
L. **`DataFromLastWeek`**
M. **`Income`**
N. **`Average_Income`**
O. **`Average_Income_In_Adjusted_Dollars`**

Problem 3: Answer true or false for each part below.

A. A TITLE2 statement clears all headings and replaces the sub-heading with the specified text

B. OPTIONS and TITLE statements are part of the PROC Step

C. A variable is one of three types: character, numeric, integer

D. A single statement is ended with either a semi-colon or a carriage return

E. A TITLE statement clears only the main heading and replaces it with the specified text

F. A single line may contain more than one statement

G. OPTIONS and TITLE statements are part of the DATA Step

H. A TITLE statement clears all headings and replaces the main heading with the specified text

I. A TITLE2 statement clears only the sub-heading and replaces it with the specified text

J. A variable is one of four types: character, numeric, integer, date

K. A single statement may stretch across several lines

L. A variable is one of three types: character, numeric, date

Problem 4: A programmer wishes to annotate the following INPUT statement with comments:

```
Input ID $ Height Weight SBP DBP;
```

For each block of code below (some of which consist of more than one statement), indicate whether it annotates the statement correctly, in the sense of having the correct syntax, if directly substituted.

Block 1:
```
Input ID $ Height Weight SBP /* systolic */ DBP /* diastolic */;
```

Block 2:
```
Input ID $ Height Weight SBP */ systolic */ DBP */ diastolic */;
```

Block 3:
```
Input ID $ Height Weight SBP *systolic DBP *diastolic;
```

Block 4:
```
Input ID $ Height Weight SBP *systolic; DBP *diastolic;
```

Block 5:
```
*SBP – systolic; DBP – diastolic;
Input ID $ Height Weight SBP DBP;
```

Block 6:
```
*SBP – systolic; *DBP – diastolic;
Input ID $ Height Weight SBP DBP;
```

Problem 5: Assume there is an external file named "PS1Prob5.txt," which is located in the directory "C:\LocalData" and whose data is in a format that is compatible with the DATA Step code

```
Data expt;
Infile 'C:\LocalData\PS1Prob5.txt';
Input sample $ temp_degF press_psi;
Run;
```

Observe that this code creates a data set with three variables, named **`sample`**, **`temp_degF`**, and **`press_psi`**. Revise the code so that the data set stores five additional variables named **`temp_degC`**, **`press_Pa`**, **`quadT2`**, **`quadP2`**, and **`quadTP`**, which are defined according to the following descriptions:

- The variable **`temp_degF`** stores temperature measurements in °F. The variable **`temp_degC`** is to store those same measurements translated to °C. The conversion formula is °C = 5/9 × (°F – 32).
- The variable **`press_psi`** stores pressure measurements in pounds per square inch (psi). The variable **`press_Pa`** is to store those same measurements translated to Pascals. The conversion formula is 1 psi = 6894.757 Pa.
- The variables **`quadT2`**, **`quadP2`**, and **`quadTP`** are to store values of the quadratic monomials x^2, y^2, and xy, respectively, where $x =$ **`temp_degC`** and $y =$ **`press_Pa`**.

Problem 6: Assume there is an external file named "PS1Prob6.txt," which is located in the directory "C:\LocalData" and whose contents are:

```
34 66
99 87
54 79
```

Identify four errors, each in a separate statement, that prevents the following DATA Step code from running correctly:

```
Data new-data;
Infile PS1Prob6.txt;
Input x1 x2;
y1 = 3(x1)+2(x2);
y2 = x1 / x2;
new_variable_with_a_complicated_formula = x1 + x2 - 37;
run;
```

Problem 7: Download the file "PS1Prob7.txt," which accompanies this assignment on the course web page, and place it into the directory "C:\LocalData." Then supplement the DATA Step

```
Data PS1Prob7;
Infile 'C:\LocalData\PS1Prob7.txt';
Input class $ x1 y1 y2;
Run;
```

with at least one PROC Step so that the combined code (the DATA Step followed by the PROC Steps) will produce the following:

- A table that lists the mean and median of each numeric variable in the data set, but no other statistics.
- A table of frequency counts and percentages of the values stored in the variable **`class`**, which does not list the corresponding cumulative frequencies or percentages.
- A high-resolution quantile plot based on the normal distribution of the variable **`x1`**.
- A matrix of correlations between the variables **`x1`** and **`y1`** (and no other variables), but with any p-values associated with those correlations suppressed from the output.