


## STAT 506: Homework 4

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.

There will obviously be a lot of exporting happening in this assignment.

- If you're using the SAS Windowing Environment on your local machine, it likely won't be complicated to access your exported files.
- If you're using [SAS OnDemand for Academics](#), you'll want to download (  ) any exported file from the remote environment and then open it locally.
- If you're using [GoRemote](#), you should be able to navigate to text files using the Explorer window in SAS and then click the file to open it in Notepad. PDF files might open automatically in Edge within the Remote environment. For Excel files, you'll want to download them. See the videos posted in "Videos for Working with Software Remote" on Brightspace under "Getting SAS & Certification Info" for help if needed.

### 1. Using PROC EXPORT

Write a DATA step that creates temporary table named **ToCSV** based on **pg1.storm\_range**. Use a combination of IF/THEN/ELSE statements and DO groups to do the following:

- If the average of the columns **Wind1**, **Wind2**, **Wind3**, and **Wind4** is above 90, create a new column named **Speeds** with the value "High". Also edit the column **Name** so that the name is entirely in uppercase.
- Otherwise, if that average is above 60, the column **Speeds** should have the value "Medium", and **Name** should be edited so that the name is in proper case.
- Otherwise, **Speeds** should be "Low", and **Name** should be in lowercase.

Write a PROC EXPORT step that saves **ToCSV** as a .csv file, using an `outpath` macro variable like the one discussed in the lecture slides.

*Show the entire code, and also open the .csv file in a text editor (like Notepad) and include a screenshot showing the first 5 rows.*

### 2. Using a LIBNAME engine to export data

Do the following:

- Write a LIBNAME statement to create an Excel workbook named **StormsByYear.xlsx**.
- Using two DATA steps, create two worksheets in that workbook named **Storms1980s** and **Storms1990s**.
- In each worksheet, include all the data from **pg1.storm\_final** where the column **Season** fits the worksheet's decade (e.g. Rows that have a Season between 1980 and 1989, inclusive, should go to the **Storms1980s** worksheet.).
- Additionally, create a new character column in each worksheet named **Decade** that has the corresponding appropriate value for each worksheet (e.g. In the **Storms1980s** worksheet, all the values of **Decade** should be "1980s").
- Unassign the libref at the end of the program.

*Show the entire code and the corresponding log notes.*

### 3. Using the ODS with Excel

Modify the below code to do the following:

- Export all of the output to a new Excel workbook named StormStats.xlsx with the style snow.
- Set the sheet name for the PROC MEANS output to “South Pacific Summary”.
- Set the sheet name for the PROC PRINT output to “Data for SP”.
- Turn off the procedure titles (e.g. “The MEANS Procedure”) at the beginning of the program, and turn them back on at the end.
- At the end of the program, make sure to terminate the connection to the Excel workbook.

```
proc means data=pg1.storm_detail maxdec=0 median max;  
  class Season;  
  var Wind;  
  where Basin='SP' and Season in (2012,2013,2014);  
run;  
  
proc print data=pg1.storm_detail noobs;  
  where Basin='SP' and Season in (2012,2013,2014);  
  by Season;  
run;
```

If you have access to Excel, open the workbook and notice how the PROC PRINT output was stored. *Show the entire code and the corresponding log notes, but you don't need to include anything from the Excel file itself.*

### 4. Using the ODS with PDF

Modify the below code to do the following:

- Export all of the output to a new PDF file named StormSummary.pdf with the style Journal.
- Use SAS Help and/or Google to find a SAS system option that changes the page layout to landscape and apply that option. Reset that option at the end of the code so future output is back in the default portrait layout.
- Use SAS Help and/or Google to learn about the ODS LAYOUT GRIDDED statement as way that you can control the layout of multiple result objects. Force the results to be arranged in one row and two columns. *Hints:* You'll definitely want to look at the examples on the help page (specifically the ODS REGION; statements). You'll also want to make sure any ODS LAYOUT statements are within the ODS PDF statements.
- Make sure to terminate the connection to the PDF at the end of the program.
- If done correctly, you should have the two different reports printed side-by-side on the same page.

```
title "2016 Northern Atlantic Storms";  
proc print data=pg1.storm_final noobs;  
  var name StartDate MaxWindMPH StormLength;  
  where Basin="NA" and Season=2016;  
  format StartDate monyy7.;  
run;  
title;  
  
ods noproctitle;  
proc means data=pg1.storm_final min mean max maxdec=1;  
  var MaxWindMPH StormLength;  
  where Basin="NA" and Season=2016;  
  class StartDate;  
  format StartDate monname.;  
run;
```

*Show the entire code and a screenshot of the resulting output in the PDF.*

## 5. Using PROC SQL to print a table

Write a PROC SQL step to do the following:

- Display the column **ParkName** from the table **pg1.np\_traffic**.
- Also display the column **Location** (after **ParkName**), but with the values in proper case. Display the name of the column as “Location” (no quotes).
- Also display the new column named **EntranceFees** (after those previously listed) which is the variable **Count** multiplied by 31 (a very rough estimate of the median entrance fee per vehicle at national parks based on some quick and lazy Googling).
- Apply a format to **EntranceFees** to display its values with a dollar sign and commas.
- Order the rows first by the column **ParkName** in ascending order and then by **EntranceFees** in descending order.

*Show the entire code and a screenshot of the output showing the first 5 rows.*

## 6. Using PROC SQL to create a joined table

Write a PROC SQL step to do the following:

- Perform an Inner Join on the tables **pg1.storm\_2017** and **pg1.storm\_basincodes** on their common column. Use aliases for the table names.
- Only include rows where **MaxWindMPH** is greater than 135.
- Save just the columns **Year**, **Basin**, **BasinName**, **Name**, **StartDate**, and **EndDate** (in that order) as a new permanent SAS table named **pg1.storm\_2017join**.
- Assign the permanent format **mmddyy10.** to both **StartDate** and **EndDate**.

Write a PROC PRINT step to display the table.

*Show the entire code and a screenshot of the PROC PRINT output.*