STAT 506: Homework 4 Key (15 + 9 + 9 + 11 + 11 + 15 = 70 pts)

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. ***Prev. sem were >50, 55***
- Otherwise, **Speeds** should be "Low", and **Name** should be in lowercase.

Write a PROC EXPORT step that saves **ToCSV** as a .csv file, using the `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.

```
data ToCsv;                                          *1 pt;
  set pg1.storm_range;
  length Speeds $ 6;                                 *1 pt;
                    *(or rearrange order below so full value is saved);
  if mean(of Wind1-Wind4) > 90 then do;
    Speeds='High';
    Name=upcase(Name);
  end;
  else if mean(of Wind1-Wind4) > 60 then do;
    Speeds='Medium';
    Name=propcase(Name);
  end;
  else do;
    Speeds='Low';
    Name=lowcase(Name);
  end;                          *9 pts for all of if/then/else/do above;
run;

%let outpath=W:\pu.data\Desktop\PG1\output;                *1 pt;

proc export data=tocsv outfile="&outpath/tocsv.csv" dbms=csv replace;
run;                                              *2 pts, replace is optional;
```

tocsv.csv

```
1  Season,Basin,Name,Wind1,Wind2,Wind3,Wind4,Speeds
2  1980,EP,AGATHA,100,95,90,85,High
3  1980,EP,blas,50,50,50,45,Low
4  1980,EP,Celia,65,65,65,65,Medium
5  1980,EP,darby,45,45,35,30,Low
6  1980,EP,estelle,40,35,35,25,Low
```

(15 pts total: 14 pts code, 1 pt output)

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.

```
libname xl xlsx "&outpath/StormsByYear.xlsx";                    *1 pt;

data xl.Storms1980s;                                             *1 pt;
  set pg1.storm_final;
  where 1980 <= Season <= 1989;                                  *1 pt;
  Decade='1980s';                                                *1 pt;
run;

data xl.Storms1990s;                                             *1 pt;
  set pg1.storm_final;
  where 1990 <= Season <= 1999;                                  *1 pt;
  Decade='1990s';                                                *1 pt;
run;

libname xl clear;                                                *1 pt;
```

```
199  libname xl xlsx "&outpath/StormsByYear.xlsx";
NOTE: Libref XL was successfully assigned as follows:
      Engine:        XLSX
      Physical Name: W:\pu.data\Desktop\PG1\output/StormsByYear.xlsx
200
201  data xl.Storms1980s;
202    set pg1.storm_final;
203    where 1980 <= Season <= 1989;
204    Decade='1980s';
205  run;

NOTE: There were 815 observations read from the data set PG1.STORM_FINAL.
      WHERE (Season>=1980 and Season<=1989);
NOTE: The data set XL.Storms1980s has 815 observations and 16 variables.
NOTE: The export data set has 815 observations and 16 variables.
NOTE: DATA statement used (Total process time):
      real time           0.65 seconds
      cpu time            0.12 seconds


206
207  data xl.Storms1990s;
208    set pg1.storm_final;
209    where 1990 <= Season <= 1999;
210    Decade='1990s';
211  run;

NOTE: There were 793 observations read from the data set PG1.STORM_FINAL.
      WHERE (Season>=1990 and Season<=1999);
NOTE: The data set XL.Storms1990s has 793 observations and 16 variables.
NOTE: The export data set has 793 observations and 16 variables.
NOTE: DATA statement used (Total process time):
      real time           0.60 seconds
      cpu time            0.12 seconds


212
213  libname xl clear;
NOTE: Libref XL has been deassigned.
```

(9 pts total: 8 pts code, 1 pt log)

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". ***Prev. sem. were "Detail" or "Storm Detail Data" or "SP Detail Data"***
- 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;
```

Prev. sem. were 2014-2016, 2013-2015

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.*

Solution on next page

```
ods excel file="&outpath/StormStats.xlsx" style=snow
    options(sheet_name='South Pacific Summary');                    *3 pts;
ods noproctitle;                                                    *1 pt;

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

ods excel options(sheet_name='Data for SP');                        *2 pts;

proc print data=pg1.storm_detail noobs;
    where Basin='SP' and Season in (2012,2013,2014);
    by Season;
run;

ods excel close;                                                    *1 pt;
ods proctitle;                                                      *1 pt;
```

```
60   ods excel file="&outpath/StormStats.xlsx" style=snow
61       options(sheet_name='South Pacific Summary');
62   ods noproctitle;
63
64   proc means data=pg1.storm_detail maxdec=0 median max;
65       class Season;
66       var Wind;
67       where Basin='SP' and Season in (2012,2013,2014);
68   run;

NOTE: There were 903 observations read from the data set PG1.STORM_DETAIL.
      WHERE (Basin='SP') and Season in (2012, 2013, 2014);
NOTE: PROCEDURE MEANS used (Total process time):
      real time           0.42 seconds
      cpu time            0.04 seconds


69
70   ods excel options(sheet_name='Data for SP');
71
72   proc print data=pg1.storm_detail noobs;
73       where Basin='SP' and Season in (2012,2013,2014);
74       by Season;
75   run;

NOTE: There were 903 observations read from the data set PG1.STORM_DETAIL.
      WHERE (Basin='SP') and Season in (2012, 2013, 2014);
NOTE: PROCEDURE PRINT used (Total process time):
      real time           1.34 seconds
      cpu time            1.18 seconds


76
77   ods excel close;
NOTE: Writing EXCEL file: W:\pu.data\Desktop\PG1\output\StormStats.xlsx
78   ods proctitle;
```

(9 pts total: 8 pts code, 1 pt log)

If they did some modifications to make the PROC PRINT output ('Data for SP') show up in one worksheet instead of 3, that's OK. It just means that they actually followed the "notice how the PROC PRINT output was stored" part of the instructions.

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.

Solution on next page

```
options orientation=landscape;                                          *1 pt;
ods pdf file="&outpath/StormSummary.pdf" style=Journal;                 *2 pts;
ODS LAYOUT GRIDDED rows=1 columns=2;     * "rows=1" is optional;        *2 pts;

ods region;                                                             *1 pt;

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 region;                                                             *1 pt;

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;

ODS LAYOUT END;                                                         *1 pt;
ods pdf close;                                                          *1 pt;
options orientation=portrait;                                           *1 pt;
```

23:32 Friday, Fel

2016 Northern Atlantic Storms

Name	StartDate	MaxWindMPH	StormLength
ALEX	JAN2016	86	10
BONNIE	MAY2016	46	13
COLIN	JUN2016	58	3
DANIELLE	JUN2016	46	3
EARL	AUG2016	86	4
FIONA	AUG2016	52	8
GASTON	AUG2016	121	13
EIGHT	AUG2016	35	5
HERMINE	AUG2016	81	11
IAN	SEP2016	63	5
KARL	SEP2016	69	14
JULIA	SEP2016	52	8
LISA	SEP2016	52	7
MATTHEW	SEP2016	167	12
NICOLE	OCT2016	138	15
OTTO	NOV2016	115	8

StartDate	N Obs	Variable	Minimum	Mean	Maximum
January	1	MaxWindMPH	86.0	86.0	86.0
		StormLength	10.0	10.0	10.0
May	1	MaxWindMPH	46.0	46.0	46.0
		StormLength	13.0	13.0	13.0
June	2	MaxWindMPH	46.0	52.0	58.0
		StormLength	3.0	3.0	3.0
August	5	MaxWindMPH	35.0	75.0	121.0
		StormLength	4.0	8.2	13.0
September	5	MaxWindMPH	52.0	80.6	167.0
		StormLength	5.0	9.2	14.0
October	1	MaxWindMPH	138.0	138.0	138.0
		StormLength	15.0	15.0	15.0
November	1	MaxWindMPH	115.0	115.0	115.0
		StormLength	8.0	8.0	8.0

(11 pts total: 10 pts code, 1 pt output)

It's OK if they reordered the title statements so the title is just over one side instead of both.

You can be forgiving on the partial credit here. The whole goal of the question was to get them to play around with some options they hadn't seen before.

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). ***[Previous semesters were 25, 27, 28, 29, 30… inflation!]***
- 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.

```
proc sql;
  select parkname, propcase(location) as Location,                    /*3 pts*/
         Count*31 as EntranceFees format=dollar8.                     /*3 pts*/
                                          /*8 is min., decimials optional*/
  from pg1.np_traffic                                                 /*1 pt*/
  order by parkname, entrancefees desc;                               /*2 pts*/
quit;                                                                 /*1 pt*/
```

Park Name	Location	EntranceFees
Abraham Lincoln Birthplace National Historical Park	Traffic Count At Main Entrance	$40,362
Abraham Lincoln Birthplace National Historical Park	Traffic Count At Knob Creek	$15,376
Abraham Lincoln Birthplace National Historical Park	Traffic Count At Picnic Parking Lot	$11,191
Acadia National Park	Traffic Count At Sand Beach	$124,899
Acadia National Park	Traffic Count At Schoodic	$60,450

(11 pts total: 10 pts code, 1 pt output)

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. ***[Prev. sem. were 120, 125, 115, 130, 140]***
- 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.

```
proc sql;
  create table pg1.Storm_2017join as                                    /*2 pts*/
  select Year, s.Basin, BasinName, Name,                                /*2 pts*/
         StartDate format=mmddyy10., EndDate format=mmddyy10.           /*2 pts*/
  from pg1.Storm_2017 as s inner join pg1.Storm_basincodes as b         /*3 pts*/
  on s.Basin=b.Basin                                                    /*2 pts*/
  where MaxWindMPH > 135;                                               /*1 pt*/
quit;                                                                   /*1 pt*/

proc print data=pg1.Storm_2017join;                                     /*1 pt*/
run;
```

Obs	Year	Basin	BasinName	Name	StartDate	EndDate
1	2017	NA	North Atlantic	IRMA	08/30/2017	09/12/2017
2	2017	NA	North Atlantic	JOSE	09/05/2017	09/22/2017
3	2017	NA	North Atlantic	MARIA	09/16/2017	09/30/2017
4	2017	EP	East Pacific	FERNANDA	07/12/2017	07/22/2017
5	2017	SI	South Indian	ERNIE	04/05/2017	04/10/2017

(15 pts total: 14 pts code, 1 pt output)