Satoshi Ido

34788706

10/06/2023

STAT506

HW4

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:

   a. 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.

   b. 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.

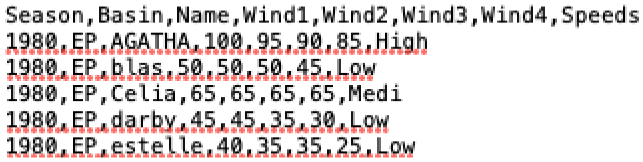
   c. Otherwise, Speeds should be “Low”, and Name should be in lowercase.

   d. 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.

```
data ToCSV;
      set pg1.storm_range;
      length Speeds $ 6;
      windavg = mean(Wind1, Wind2, Wind3, Wind4);
      if windavg > 90 then do;
            Speeds = "High";
            Name = UPCASE(Name);
      end;
      else if windavg > 60 then do;
            Speeds = "Medium";
            Name = PROPCASE(Name);
      end;
      else do;
            Speeds = "Low";
            Name = LOWCASE(Name);
      end;
      drop windavg;
run;

proc export data=ToCSV dbms=csv outfile="&outpath/question1.csv"
      ;
run;
```



```
Season,Basin,Name,Wind1,Wind2,Wind3,Wind4,Speeds
1980,EP,AGATHA,100,95,90,85,High
1980,EP,blas,50,50,50,45,Low
1980,EP,Celia,65,65,65,65,Medi
1980,EP,darby,45,45,35,30,Low
1980,EP,estelle,40,35,35,25,Low
```

2. Using a LIBNAME engine to export data

   Do the following:

   a. Write a LIBNAME statement to create an Excel workbook named StormsByYear.xlsx.
   b. Using two DATA steps, create two worksheets in that workbook named Storms1980s and Storms1990s.
   c. 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).
   d. 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”).
   e. Unassign the libref at the end of the program.

   Show the entire code and the corresponding log notes.

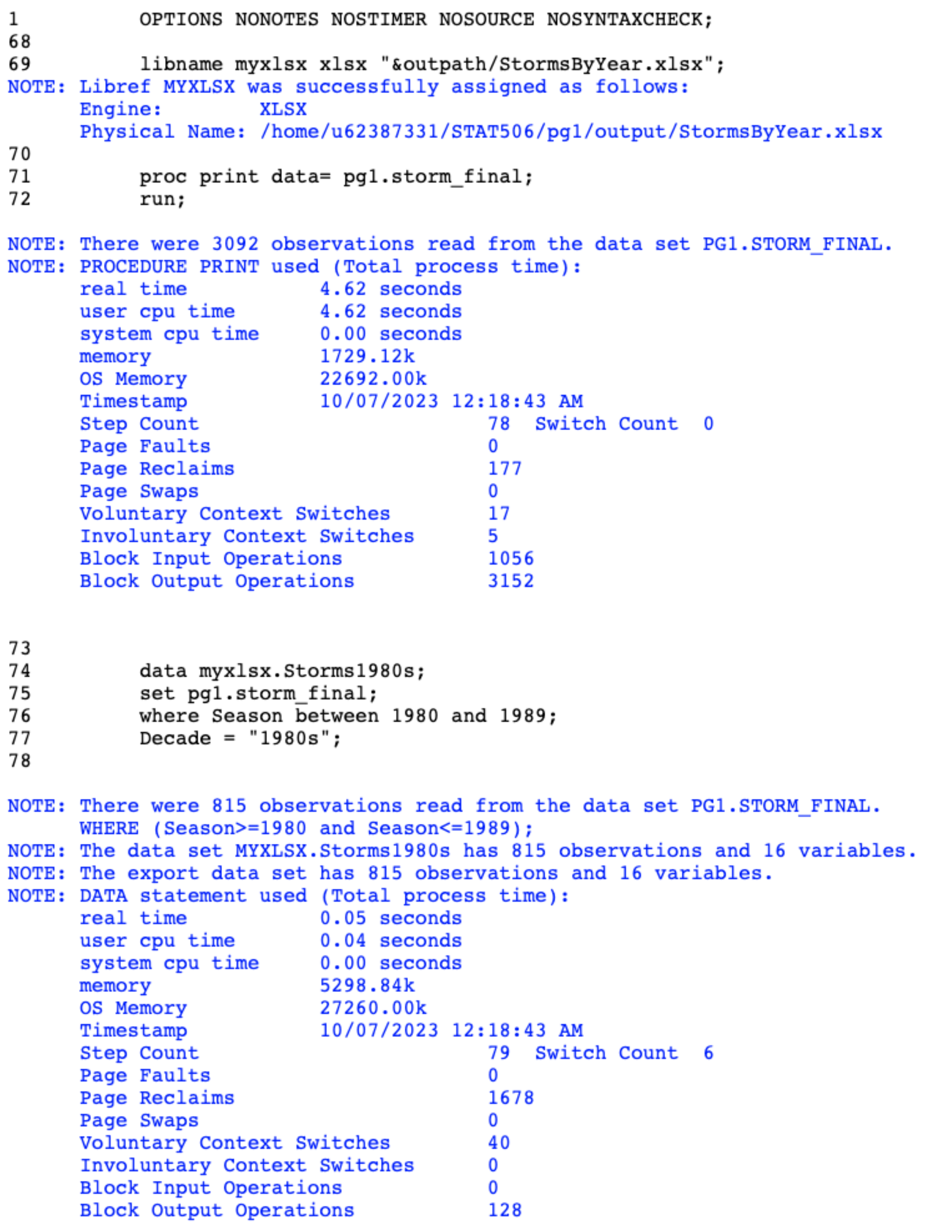
```
libname myxlsx xlsx "&outpath/StormsByYear.xlsx";

data myxlsx.Storms1980s;
      set pg1.storm_final;
      where Season between 1980 and 1989;
      Decade = "1980s";

data myxlsx.Storms1990s;
      set pg1.storm_final;
      where Season between 1990 and 1999;
      Decade = "1990s";
```

```
run;

libname myxlsx clear;
```

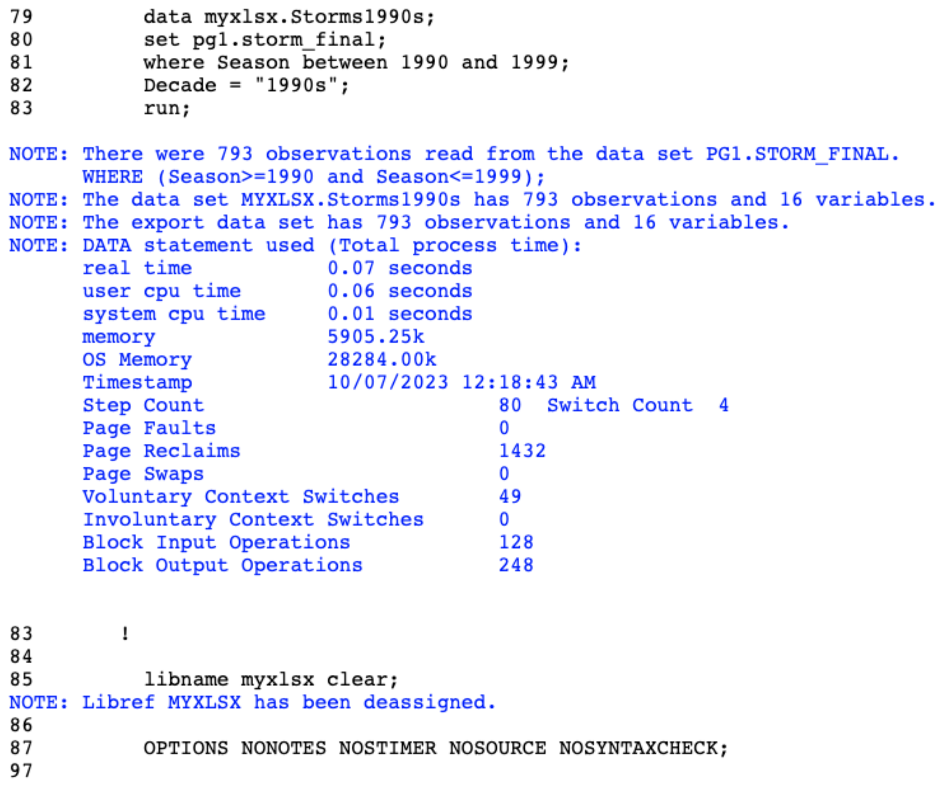


```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69         libname myxlsx xlsx "&outpath/StormsByYear.xlsx";
NOTE: Libref MYXLSX was successfully assigned as follows:
      Engine:        XLSX
      Physical Name: /home/u62387331/STAT506/pg1/output/StormsByYear.xlsx
70
71         proc print data= pg1.storm_final;
72         run;

NOTE: There were 3092 observations read from the data set PG1.STORM_FINAL.
NOTE: PROCEDURE PRINT used (Total process time):
      real time           4.62 seconds
      user cpu time       4.62 seconds
      system cpu time     0.00 seconds
      memory              1729.12k
      OS Memory           22692.00k
      Timestamp           10/07/2023 12:18:43 AM
      Step Count                        78  Switch Count  0
      Page Faults                       0
      Page Reclaims                     177
      Page Swaps                        0
      Voluntary Context Switches        17
      Involuntary Context Switches      5
      Block Input Operations            1056
      Block Output Operations           3152


73
74         data myxlsx.Storms1980s;
75         set pg1.storm_final;
76         where Season between 1980 and 1989;
77         Decade = "1980s";
78

NOTE: There were 815 observations read from the data set PG1.STORM_FINAL.
      WHERE (Season>=1980 and Season<=1989);
NOTE: The data set MYXLSX.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           0.05 seconds
      user cpu time       0.04 seconds
      system cpu time     0.00 seconds
      memory              5298.84k
      OS Memory           27260.00k
      Timestamp           10/07/2023 12:18:43 AM
      Step Count                        79  Switch Count  6
      Page Faults                       0
      Page Reclaims                     1678
      Page Swaps                        0
      Voluntary Context Switches        40
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           128
```



```
79          data myxlsx.Storms1990s;
80          set pg1.storm_final;
81          where Season between 1990 and 1999;
82          Decade = "1990s";
83          run;

NOTE: There were 793 observations read from the data set PG1.STORM_FINAL.
      WHERE (Season>=1990 and Season<=1999);
NOTE: The data set MYXLSX.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           0.07 seconds
      user cpu time       0.06 seconds
      system cpu time     0.01 seconds
      memory              5905.25k
      OS Memory           28284.00k
      Timestamp           10/07/2023 12:18:43 AM
      Step Count                        80  Switch Count  4
      Page Faults                       0
      Page Reclaims                     1432
      Page Swaps                        0
      Voluntary Context Switches        49
      Involuntary Context Switches      0
      Block Input Operations            128
      Block Output Operations           248


83        !
84
85          libname myxlsx clear;
NOTE: Libref MYXLSX has been deassigned.
86
87          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
97
```

3. Using the ODS with Excel

   Modify the below code to do the following:

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

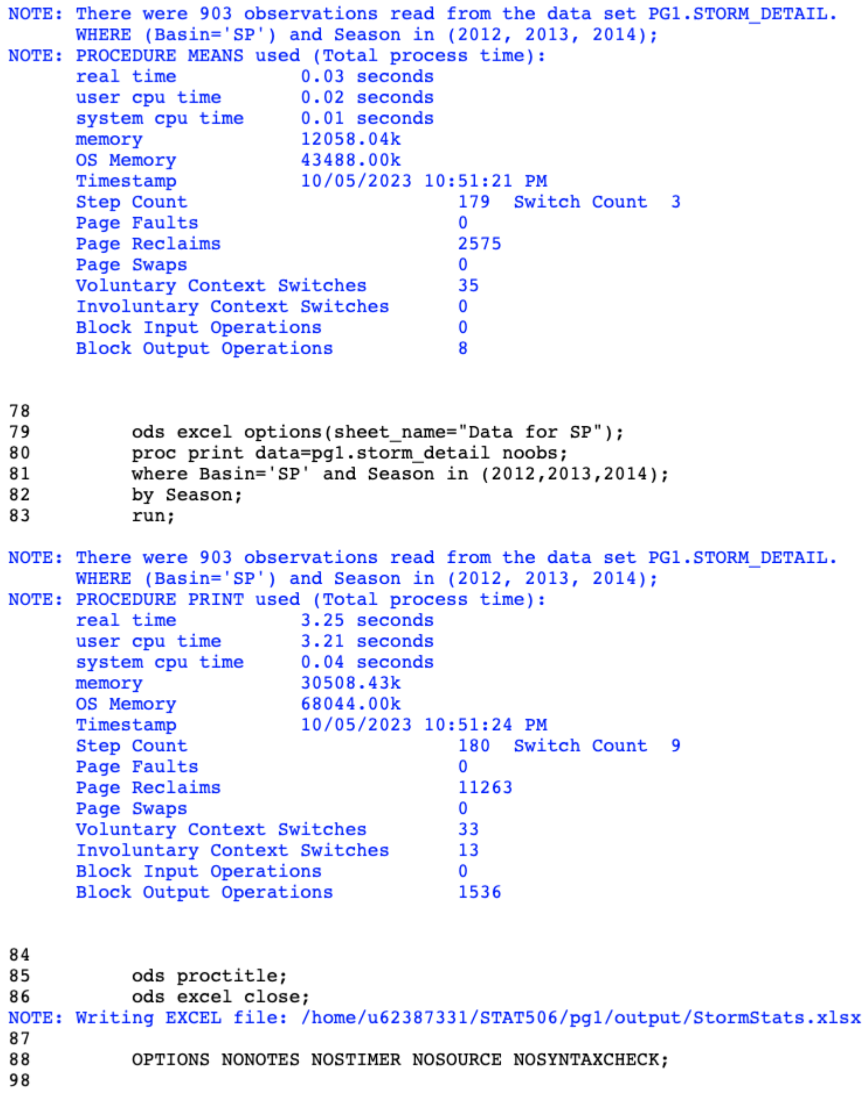
   If you have access to Excel, open the workbook and notice how the PROC PRINT output was stored. Show the entire code and the corresponding log notes, but you don’t need to include anything from the Excel file itself.

```
ods excel file="&outpath/StormStats.xlsx" style=snow
options(sheet_name="South Pacific Summary");
ods noproctitle;
```

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

ods proctitle;
ods excel close;
```



```
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           0.03 seconds
      user cpu time       0.02 seconds
      system cpu time     0.01 seconds
      memory              12058.04k
      OS Memory           43488.00k
      Timestamp           10/05/2023 10:51:21 PM
      Step Count                        179  Switch Count  3
      Page Faults                       0
      Page Reclaims                     2575
      Page Swaps                        0
      Voluntary Context Switches        35
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           8


           ods excel options(sheet_name="Data for SP");
           proc print data=pg1.storm_detail noobs;
           where Basin='SP' and Season in (2012,2013,2014);
           by Season;
           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           3.25 seconds
      user cpu time       3.21 seconds
      system cpu time     0.04 seconds
      memory              30508.43k
      OS Memory           68044.00k
      Timestamp           10/05/2023 10:51:24 PM
      Step Count                        180  Switch Count  9
      Page Faults                       0
      Page Reclaims                     11263
      Page Swaps                        0
      Voluntary Context Switches        33
      Involuntary Context Switches      13
      Block Input Operations            0
      Block Output Operations           1536


           ods proctitle;
           ods excel close;
NOTE: Writing EXCEL file: /home/u62387331/STAT506/pg1/output/StormStats.xlsx

           OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

```

4. Using the ODS with PDF

   Modify the below code to do the following:

   a. Export all of the output to a new PDF file named StormSummary.pdf with the style Journal.
   b. 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.
   c. 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.
   d. Make sure to terminate the connection to the PDF at the end of the program.
   e. If done correctly, you should have the two different reports printed side-by-side on the same page.

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

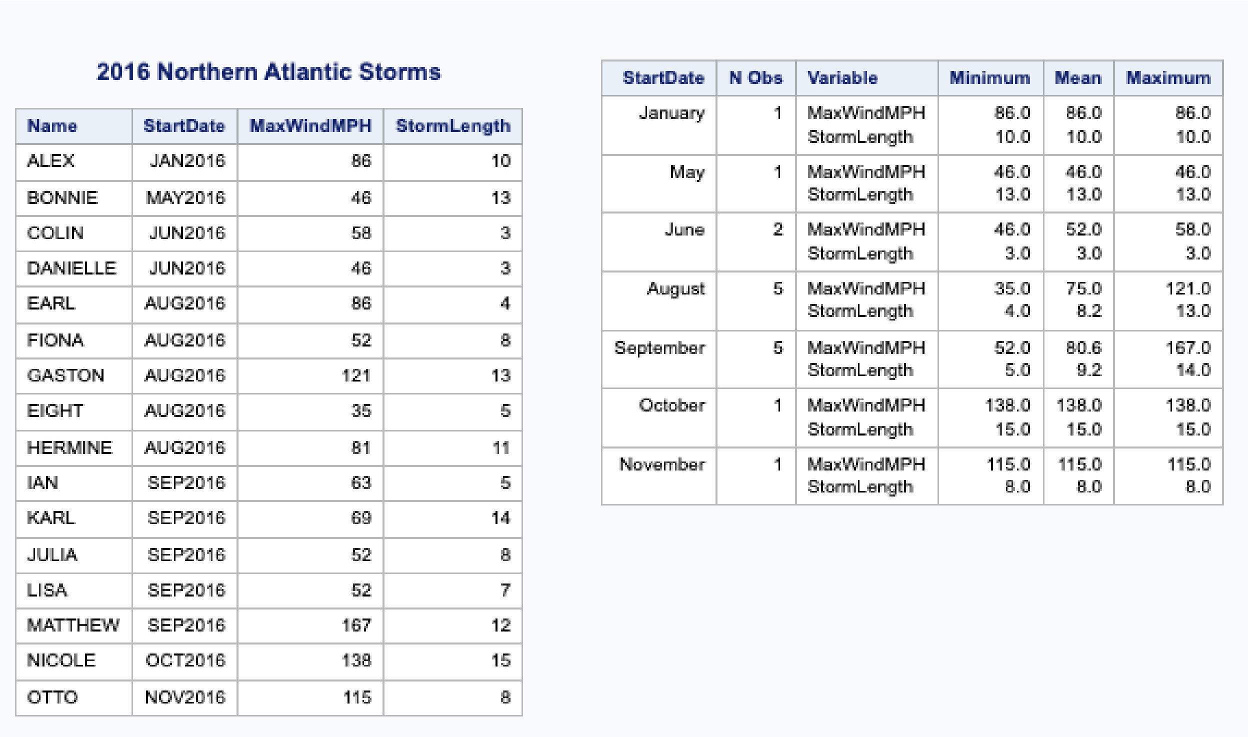
```
options orientation=landscape;
ods pdf file="&outpath/StormSummary.pdf" style=Journal;

ods layout gridded columns=2;
/* first design */
ods region;
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;

/* second design */
ods region;
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;
ods proctitle;
ods pdf close;

options orientation=portrait;
```



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

5. Using PROC SQL to print a table

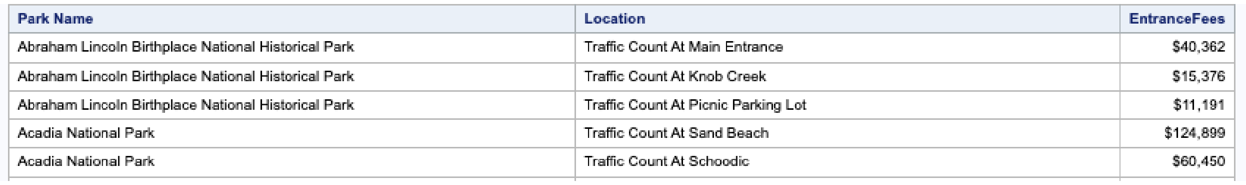
   Write a PROC SQL step to do the following:

   a. Display the column ParkName from the table pg1.np_traffic.
   b. Also display the column Location (after ParkName), but with the values in proper case. Display the name of the column as “Location” (no quotes).
   c. 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).

d. Apply a format to EntranceFees to display its values with a dollar sign and commas.

e. 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
            ,Count * 31 as EntranceFees format=dollar8.
      from
            pg1.np_traffic
      order by
            ParkName asc, EntranceFees desc;
quit;
```



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

6. Using PROC SQL to create a joined table

Write a PROC SQL step to do the following:

a. Perform an Inner Join on the tables pg1.storm_2017 and pg1.storm_basincodes on their common column. Use aliases for the table names.

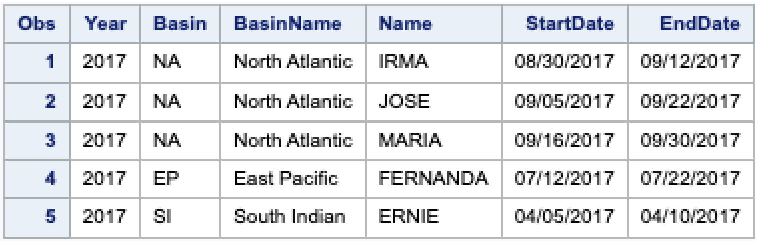
b. Only include rows where MaxWindMPH is greater than 135.

c. Save just the columns Year, Basin, BasinName, Name, StartDate, and EndDate (in that order) as a new permanent SAS table named pg1.storm_2017join.

d. 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
            select
                  Year
                  ,ps2017.Basin
                  ,BasinName
                  ,Name
                  ,StartDate format=mmddyy10.
                  ,EndDate format=mmddyy10.
            from
                  pg1.storm_2017 as ps2017
            inner join
                  pg1.storm_basincodes as psb on psb.Basin = ps2017.Basin
            where
                  ps2017.MaxWindMPH > 135;
quit;
```



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