Satoshi Ido

34788706

09/27/2023

STAT506

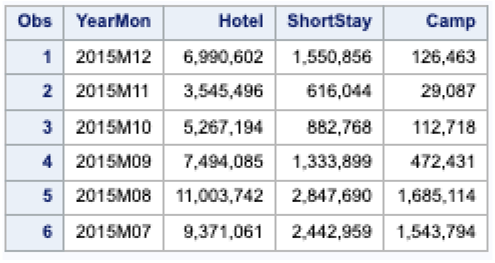
HW3

1. DATA step processing and filtering

   Write a DATA step to do the following:

   - Read in the table pg1.eu_occ.
   - Add a WHERE statement to select only the stays that were reported in the year 2015. Use the substr() function. [Note that YearMon is a character column, and the first four characters represent the year.]
   - Assign the COMMA10. format to the Hotel, ShortStay, and Camp columns.
   - Save the new table as eu_occ2015, but exclude the columns Geo and Country. Print the first 6 observations of eu_occ2015. Show your code and the output.

```
data eu_occ2015;
       set pg1.eu_occ;
       where substr(YearMon, 1, 4) = "2015";
       format Hotel ShortStay Camp COMMA10.;
       drop Geo Country;
run;

proc print data=eu_occ2015(obs=6);
run;
```



Obs	YearMon	Hotel	ShortStay	Camp
1	2015M12	6,990,602	1,550,856	126,463
2	2015M11	3,545,496	616,044	29,087
3	2015M10	5,267,194	882,768	112,718
4	2015M09	7,494,085	1,333,899	472,431
5	2015M08	11,003,742	2,847,690	1,685,114
6	2015M07	9,371,061	2,442,959	1,543,794

2. Creating New Columns

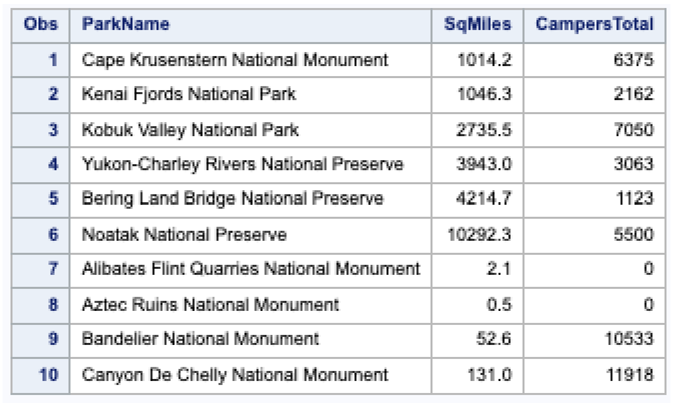
   Write a DATA step to do the following:

   Read in the table pg1.np_summary.

   - Create a new column named SqMiles by dividing the column Acres by 640.

- Create a new column named CampersTotal as the sum of OtherCamping, TentCampers, RVCampers, and BackcountryCampers.
- Format SqMiles to show one decimal place.
- Save the new table as np_summary_update, but only include the column ParkName and the new columns created above.
- Print the first 10 observations of np_summary_update. Show your code and the output.

```
data np_summary_update;
	set pg1.np_summary;
	SqMiles = Acres / 640;
	CampersTotal = sum(OtherCamping, TentCampers, RVCampers,
BackcountryCampers);
	format SqMiles 8.1;
	keep ParkName CampersTotal SqMiles;
run;

proc print data=np_summary_update(obs=10);
run;
```



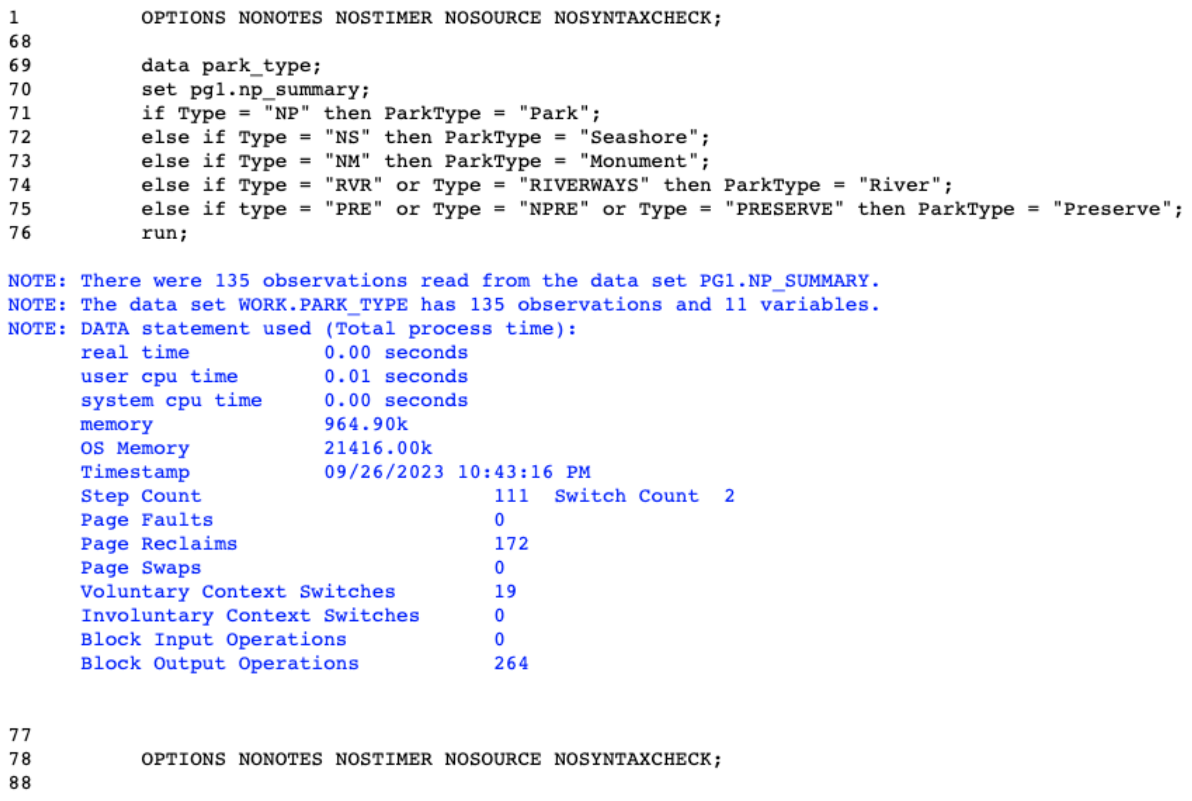
Obs	ParkName	SqMiles	CampersTotal
1	Cape Krusenstern National Monument	1014.2	6375
2	Kenai Fjords National Park	1046.3	2162
3	Kobuk Valley National Park	2735.5	7050
4	Yukon-Charley Rivers National Preserve	3943.0	3063
5	Bering Land Bridge National Preserve	4214.7	1123
6	Noatak National Preserve	10292.3	5500
7	Alibates Flint Quarries National Monument	2.1	0
8	Aztec Ruins National Monument	0.5	0
9	Bandelier National Monument	52.6	10533
10	Canyon De Chelly National Monument	131.0	11918

3. Using Conditional Processing to Re-Categorize and Clean Data
   - As we've seen previously, the table pg1.np_summary is using some inconsistent codes for the column Type. Create a frequency table for Type. Show just your code.

```
proc sort data=pg1.np_summary
	out=pg1.np_summary_sorted;
	by Type;
run;

proc freq data=pg1.np_summary_sorted;
	table Type;
run;
```

- Write a DATA step to create a new table named park_type that includes everything from pg1.np_summary. Also
    - use IF-THEN/ELSE statements to create a new character column named ParkType based on the value of Type:
    - Type = “NP” → ParkType = “Park”
    - Type = “NS” → ParkType = “Seashore”
    - Type = “NM” → ParkType = “Monument”
    - Type = “RVR” or “RIVERWAYS” → ParkType = “River”
    - Type = “PRE”, “NPRE”, or “PRESERVE” → ParkType = “Preserve” Show your code and the corresponding Log notes.

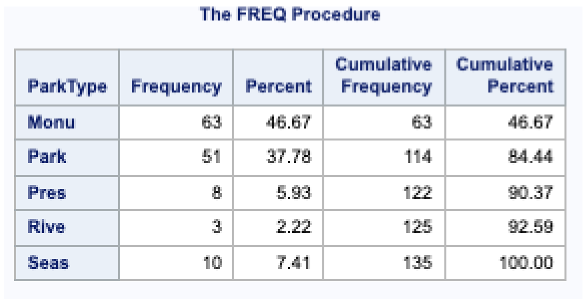
```
data park_type;
	set pg1.np_summary;
	if Type = "NP" then ParkType = "Park";
	else if Type = "NS" then ParkType = "Seashore";
	else if Type = "NM" then ParkType = "Monument";
	else if Type = "RVR" or Type = "RIVERWAYS" then ParkType =
"River";
	else if type = "PRE" or Type = "NPRE" or Type = "PRESERVE" then
ParkType = "Preserve";
run;
```



```
          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

          data park_type;
          set pg1.np_summary;
          if Type = "NP" then ParkType = "Park";
          else if Type = "NS" then ParkType = "Seashore";
          else if Type = "NM" then ParkType = "Monument";
          else if Type = "RVR" or Type = "RIVERWAYS" then ParkType = "River";
          else if type = "PRE" or Type = "NPRE" or Type = "PRESERVE" then ParkType = "Preserve";
          run;

NOTE: There were 135 observations read from the data set PG1.NP_SUMMARY.
NOTE: The data set WORK.PARK_TYPE has 135 observations and 11 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      user cpu time       0.01 seconds
      system cpu time     0.00 seconds
      memory              964.90k
      OS Memory           21416.00k
      Timestamp           09/26/2023 10:43:16 PM
      Step Count                        111  Switch Count  2
      Page Faults                       0
      Page Reclaims                     172
      Page Swaps                        0
      Voluntary Context Switches        19
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           264



          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;

```

- Create a frequency table for ParkType. Show your code and the output.

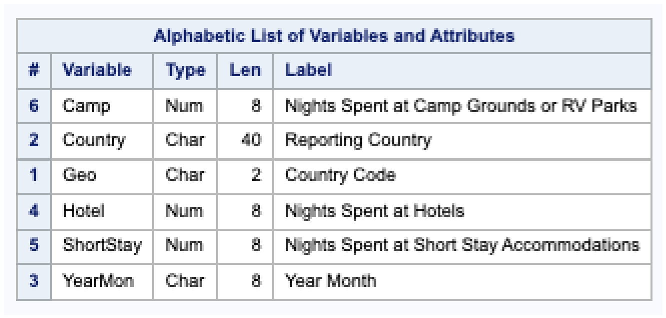
```
proc sort data=park_type
       out=park_type_sorted;
       by ParkType;
run;

proc freq data=park_type_sorted;
       table ParkType;
run;
```



The FREQ Procedure

ParkType	Frequency	Percent	Cumulative Frequency	Cumulative Percent
Monu	63	46.67	63	46.67
Park	51	37.78	114	84.44
Pres	8	5.93	122	90.37
Rive	3	2.22	125	92.59
Seas	10	7.41	135	100.00

4. Using Labels in PROC PRINT
   - Write a PROC CONTENTS step to display the descriptor portion of pg1.eu_occ to see the permanent labels assigned to the columns. Show the relevant part of the output (the table that shows the labels).

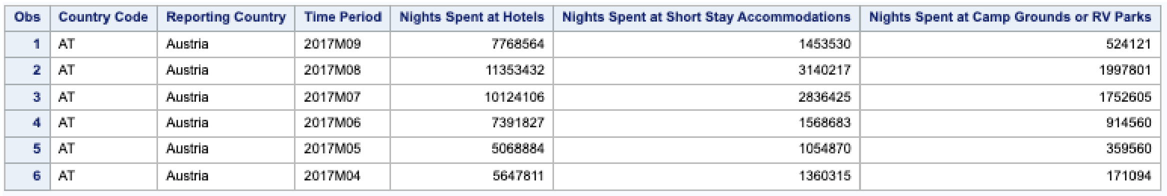


Alphabetic List of Variables and Attributes

#	Variable	Type	Len	Label
6	Camp	Num	8	Nights Spent at Camp Grounds or RV Parks
2	Country	Char	40	Reporting Country
1	Geo	Char	2	Country Code
4	Hotel	Num	8	Nights Spent at Hotels
5	ShortStay	Num	8	Nights Spent at Short Stay Accommodations
3	YearMon	Char	8	Year Month

   - Print the first 6 observations from pg1.eu_occ. All the columns should be displayed with their permanent labels, except for YearMon, which should have the temporarily assigned label “Time Period” displayed instead. Show your code and output.

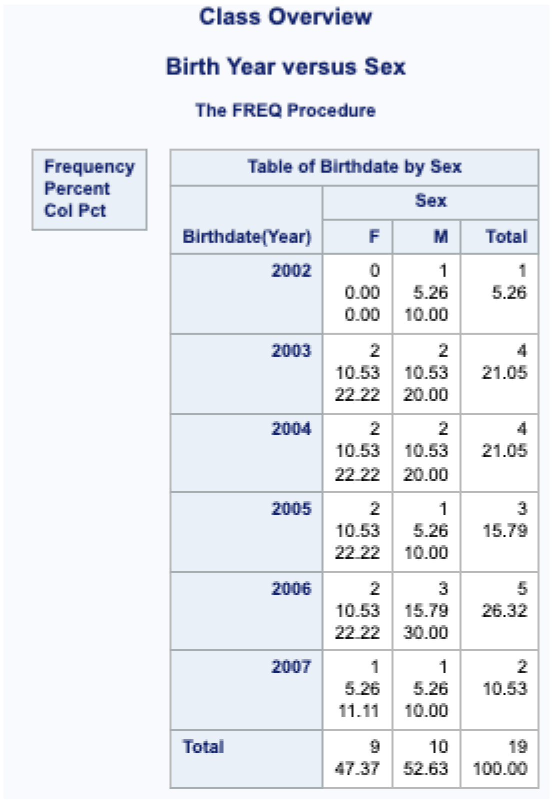
```
proc print data=pg1.eu_occ(obs=6) label;
       label YearMon = "Time Period";
run;
```



Obs	Country Code	Reporting Country	Time Period	Nights Spent at Hotels	Nights Spent at Short Stay Accommodations	Nights Spent at Camp Grounds or RV Parks
1	AT	Austria	2017M09	7768564	1453530	524121
2	AT	Austria	2017M08	11353432	3140217	1997801
3	AT	Austria	2017M07	10124106	2836425	1752605
4	AT	Austria	2017M06	7391827	1568683	914560
5	AT	Austria	2017M05	5068884	1054870	359560
6	AT	Austria	2017M04	5647811	1360315	171094

5. Two-Way Frequency Reports
   - Make a two-way frequency report for the columns sex and birthdate in pg1.class_birthdate.
     - Use birthdate as the column variable.
     - Use a format to group the values of birthdate by year instead of by individual date. If done properly, this should result in a table with 6 year columns.
     - Add the label “Year” to birthdate.
     - Add the titles “Class Overview” on the first line and “Birth Year versus Sex” on the third line.
     - Add your name as a footnote.
     - Use options in the TABLES statement to show only the frequencies and the column percentages in each cell. Add code to clear the titles and footnote after the report is generated.
   - Show your code and the output.

```
title1 "Class Overview";
title3 "Birth Year versus Sex";
footnote1 "Satoshi Ido";
proc freq data=pg1.class_birthdate;
      tables birthdate * sex / norow nocum;
      format birthdate YEAR.;
      label birthdate="Year";
run;
title;
footnote;
```


Class Overview

Birth Year versus Sex

The FREQ Procedure

Frequency Percent Col Pct

Table of Birthdate by Sex

	Sex		
Birthdate(Year)	F	M	Total
2002	0 0.00 0.00	1 5.26 10.00	1 5.26
2003	2 10.53 22.22	2 10.53 20.00	4 21.05
2004	2 10.53 22.22	2 10.53 20.00	4 21.05
2005	2 10.53 22.22	1 5.26 10.00	3 15.79
2006	2 10.53 22.22	3 15.79 30.00	5 26.32
2007	1 5.26 11.11	1 5.26 10.00	2 10.53
Total	9 47.37	10 52.63	19 100.00

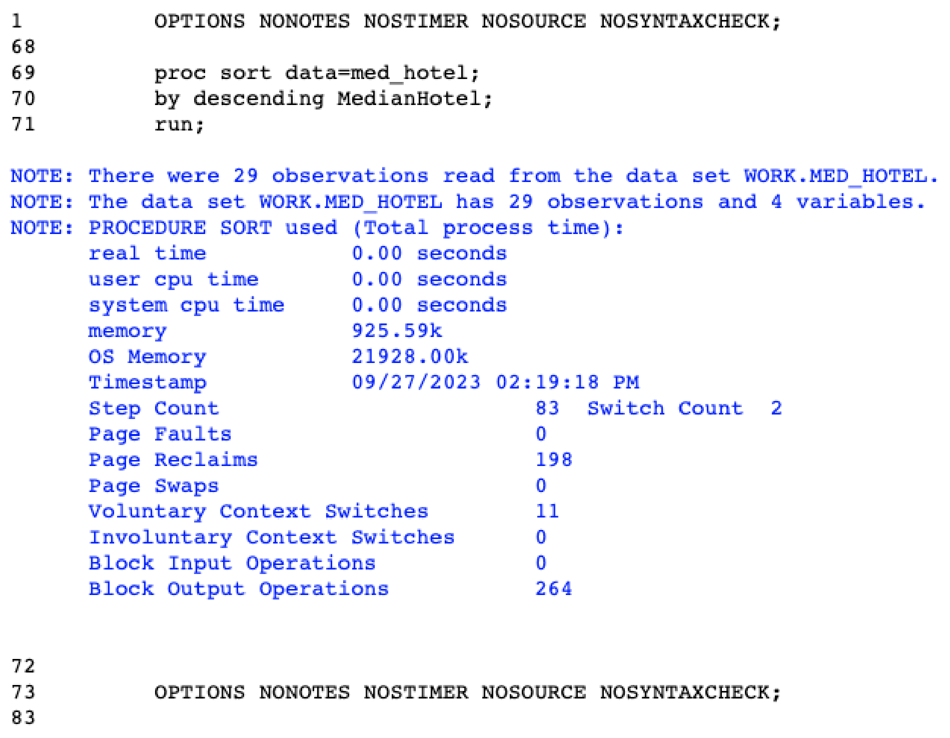
6. Creating an Output Summary Table
   - Write a PROC MEANS step that will calculate summary statistics for the variable hotel in pg1.eu_occ using country as the class variable. Save the output as a new temporary table named med_hotel which includes the median values for the hotel variable as a variable named MedianHotel. Use the NOPRINT option. Show your code and the corresponding Log notes.

```
proc means data=pg1.eu_occ noprint;
      var hotel;
      class country;
      output out=med_hotel median=MedianHotel;
run;
```

```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69         proc means data=pg1.eu_occ noprint;
70         var hotel;
71         class country;
72         output out=med_hotel median=MedianHotel;
73         run;

NOTE: There were 4785 observations read from the data set PG1.EU_OCC.
NOTE: The data set WORK.MED_HOTEL has 30 observations and 4 variables.
NOTE: PROCEDURE MEANS used (Total process time):
      real time           0.00 seconds
      user cpu time       0.01 seconds
      system cpu time     0.00 seconds
      memory              8188.59k
      OS Memory           28476.00k
      Timestamp           09/27/2023 02:03:06 PM
      Step Count                        53  Switch Count  3
      Page Faults                       0
      Page Reclaims                     1869
      Page Swaps                        0
      Voluntary Context Switches        38
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           272


74
75         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
85
```

- Write a PROC SORT step to sort med_hotel by MedianHotel in descending order. If you didn’t do the PROC MEANS step in a way that automatically removes the row that summarizes the entire table (the row with a blank Country), then filter out that row in this PROC SORT step. There should be 29 observations in med_hotel now. Show your code and the corresponding Log notes.

```
data med_hotel;
  set med_hotel (firstobs=2);
run;

proc sort data=med_hotel;
       by descending MedianHotel;
run;
```



```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69         proc sort data=med_hotel;
70         by descending MedianHotel;
71         run;

NOTE: There were 29 observations read from the data set WORK.MED_HOTEL.
NOTE: The data set WORK.MED_HOTEL has 29 observations and 4 variables.
NOTE: PROCEDURE SORT used (Total process time):
      real time           0.00 seconds
      user cpu time       0.00 seconds
      system cpu time     0.00 seconds
      memory              925.59k
      OS Memory           21928.00k
      Timestamp           09/27/2023 02:19:18 PM
      Step Count                        83  Switch Count  2
      Page Faults                       0
      Page Reclaims                     198
      Page Swaps                        0
      Voluntary Context Switches        11
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           264


72
73         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
83
```

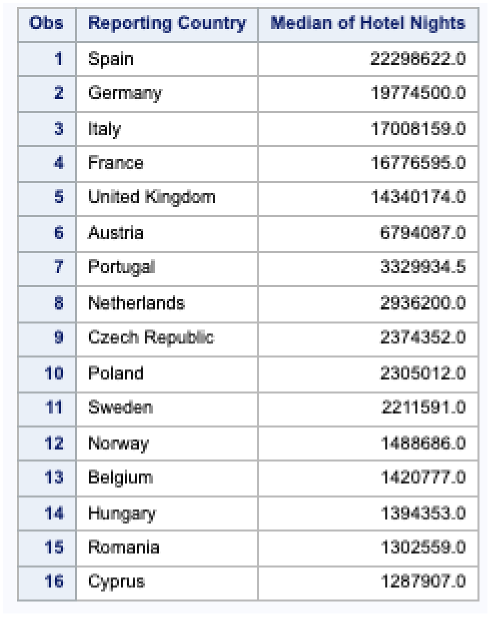
- Write a DATA step to update med_hotel by eliminating the columns _TYPE_ and _FREQ_. In this step, also assign MedianHotel the permanent label “Median of Hotel Nights”. Show your code and the corresponding Log notes.

```
data med_hotel;
       set med_hotel;
       label MedianHotel="Median of Hotel Nights";
       drop _TYPE_ _FREQ_;
run;
```

```
1          OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
68
69         data med_hotel;
70         set med_hotel;
71         label MedianHotel="Median of Hotel Nights";
72         drop _TYPE_ _FREQ_;
73         run;

NOTE: There were 29 observations read from the data set WORK.MED_HOTEL.
NOTE: The data set WORK.MED_HOTEL has 29 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time           0.00 seconds
      user cpu time       0.00 seconds
      system cpu time     0.00 seconds
      memory              941.50k
      OS Memory           21928.00k
      Timestamp           09/27/2023 02:24:29 PM
      Step Count                        89  Switch Count  2
      Page Faults                       0
      Page Reclaims                     171
      Page Swaps                        0
      Voluntary Context Switches        13
      Involuntary Context Switches      0
      Block Input Operations            0
      Block Output Operations           272


74
75         OPTIONS NONOTES NOSTIMER NOSOURCE NOSYNTAXCHECK;
85
```

- Finally, print the first 16 observations from med_hotel and display the labels for the variables. Show your code and output.

```
proc print data=med_hotel(obs=16) label;
run;
```



Obs	Reporting Country	Median of Hotel Nights
1	Spain	22298622.0
2	Germany	19774500.0
3	Italy	17008159.0
4	France	16776595.0
5	United Kingdom	14340174.0
6	Austria	6794087.0
7	Portugal	3329934.5
8	Netherlands	2936200.0
9	Czech Republic	2374352.0
10	Poland	2305012.0
11	Sweden	2211591.0
12	Norway	1488686.0
13	Belgium	1420777.0
14	Hungary	1394353.0
15	Romania	1302559.0
16	Cyprus	1287907.0