**Practice 25th November 2024**

**Conditionally Creating Columns and Output Tables**

The **pg2.np\_2017** table contains monthly public use figures for national parks.

**Reminder**: If you restarted your SAS session, submit your **libname.sas** program or run your **Autoexec** process flow to access the practice data.

1. Create a new program.  
     
   * Write a DATA step that creates temporary SAS tables named **camping** and **lodging** and reads the **pg2.np\_2017** table.
   * Compute a new column, **CampTotal**, that is the sum of **CampingOther**, **CampingTent**, **CampingRV**, and **CampingBackcountry**.
   * Format **CampTotal** so that values are displayed with commas.
   * The **camping** table has the following specifications:
     + includes rows if **CampTotal** is greater than zero
     + contains the **ParkName**, **Month**, **DayVisits**, and **CampTotal** columns
   * The **lodging** table has the following specifications:
     + includes rows where **LodgingOther** is greater than zero
     + contains only the **ParkName**, **Month**, **DayVisits**, and **LodgingOther** columns
   * Submit the program and verify the output. The notes in the SAS log indicate how many rows are in each table.

1. How many rows are in the **camping** table?
2. How many rows are in the **lodging** table?

**Practice: Producing a Running Total**

The **pg2.np\_yearlytraffic** table contains annual traffic counts at locations in national parks.

**Reminder**: If you restarted your SAS session, submit your **libname.sas** program or run your **Autoexec** process flow to access the practice data.

1. Open the **pg2.np\_yearlytraffic** table. Notice that the **Count** column records the number of cars that have passed through a particular location.
2. Open **p202p01.sas** from the **practices** folder.  
     
   * Modify the DATA step to create a column, **totTraffic**, that is the running total of **Count**.
   * Keep the **ParkName**, **Location**, **Count**, and **totTraffic** columns in the output table.
   * Format **totTraffic** so that values are displayed with commas.
   * Submit the program and examine the output data.

**Producing Multiple Totals**

The **pg2.np\_yearlytraffic** table contains annual traffic counts at locations in national parks. Parks are classified as one of five types: National Monument, National Park, National Preserve, National River, and National Seashore.

**Reminder**: If you restarted your SAS session, submit your **libname.sas** program or run your **Autoexec** process flow to access the practice data.

1. Create a table, **parkTypeTraffic**, from the **pg2.np\_yearlytraffic** table. Use the following specifications:  
     
   * Read only the rows from the input table where **ParkType** is *National Monument* or *National Park*.
   * Create two new columns named **MonumentTraffic**and **ParkTraffic**. The value of each column should be increased by the value of **Count** for that park type.
   * Format the new columns so that values are displayed with commas.

1. Create a listing report of **parkTypeTraffic**.  
     
   * Use **Accumulating Traffic Totals** for **Park Types** as the report title.
   * Display the columns in this order: **ParkType**, **ParkName**, **Location**, **Count**, **MonumentTraffic**, and **ParkTraffic**.
   * Submit the program and view the results.

1. Which row has the first nonzero value for **MonumentTraffic**?
2. What is the value of **ParkTraffic** in row 10?

**Challenge Practice: Creating a New Column with the SCAN Function**

Access SAS Help to learn about the SCAN function.

**Reminder**: If you restarted your SAS session,you must recreate the **PG1** library so you can access your practice files. In SAS Studio, open and submit the **libname.sas** program in the **EPG1V2** folder. In Enterprise Guide, run the **Autoexec** process flow.

1. Create a new program.  
     
   * Create a new temporary table named **np\_summary2** that is based on **pg1.np\_summary**.
   * Use the SCAN function to create a new column named **ParkType** that is the last word in the **ParkName** column. Use a negative number for the second argument to count words from right to left in the character string.
   * Keep **Reg**, **Type**, **ParkName**, and **ParkType** in the output table.
   * Submit the program and view the output data.

1. What is the value of **ParkType** in row four?