



## Practice

If you restarted your SAS session, open and submit the **libname.sas** program in the course files.

### Level 1

#### 4. Using the SCAN and PROPCASE Functions

The **pg2.np\_monthlytraffic** table contains monthly traffic statistics for national parks. However, the data has some inconsistencies. There is no column containing park type, and the gate location does not use proper case.

- a. Open the **p203p04.sas** program from the **practices** folder. Run the program and examine the data. Notice that **ParkName** includes a code at the end of each value that represents the park type. Also notice that some of the values for **Location** are in uppercase.
- b. Add a LENGTH statement to create a new five-character column named **Type**.
- c. Add an assignment statement that uses the SCAN function to extract the last word from the **ParkName** column and assigns the resulting value to **Type**.
- d. Add an assignment statement to use the UPCASE and COMPRESS functions to change the case of **Region** and remove any blanks.
- e. Add an assignment statement to use the PROPCASE function to change the case of **Location**.

	ParkName	ParkCode	Region	Location	Month	Count	Type
1	Acadia NP	ACAD	NORTHEAST	Traffic Count At Sand Beach	1	3,561	NP
2	Acadia NP	ACAD	NORTHEAST	Traffic Count At Sand Beach	2	3,345	NP
3	Acadia NP	ACAD	NORTHEAST	Traffic Count At Sand Beach	3	3,849	NP
4	Acadia NP	ACAD	NORTHEAST	Traffic Count At Sand Beach	4	11,101	NP
5	Acadia NP	ACAD	NORTHEAST	Traffic Count At Sand Beach	5	25,473	NP
6	Acadia NP	ACAD	NORTHEAST	Traffic Count At Sand Beach	6	50,576	NP
7	Acadia NP	ACAD	NORTHEAST	Traffic Count At Sand Beach	7	75,152	NP

### Level 2

#### 5. Searching for Character Strings

- a. Open the **p203p05.sas** program from the **practices** folder. Notice that the DATA step creates a table named **parks** and reads only those rows where **ParkName** ends with **NP**.
- b. Modify the DATA step to create or modify the following columns:
  - 1) Use the SUBSTR function to create a new column named **Park** that reads each **ParkName** value and excludes the NP code at the end of the string.
 

**Note:** Use the FIND function to identify the position number of the **NP** string. That value can be used as the third argument of the SUBSTR function to specify how many characters to read.
  - 2) Convert the **Location** column to proper case. Use the COMPBL function to remove any extra blanks between words.
  - 3) Use the TRANWRD function to create a new column named **Gate** that reads **Location** and converts the string *Traffic Count At* to a blank.

- 4) Create a new column names **GateCode** that concatenates **ParkCode** and **Gate** together with a single hyphen between the strings.

Obs	Park	GateCode	Month	Count
1	Acadia	ACAD-Sand Beach	1	3,561
2	Acadia	ACAD-Sand Beach	2	3,345
3	Acadia	ACAD-Sand Beach	3	3,849
4	Acadia	ACAD-Sand Beach	4	11,101
5	Acadia	ACAD-Sand Beach	5	25,473
6	Acadia	ACAD-Sand Beach	6	50,576

## Challenge

### 6. Determining the Maximum Length of a Column

The **pg2.np\_unstructured\_codes** table contains a single column whose contents include location codes and names. Create a table that efficiently stores the location code and location name.

- Open the **p203p06.sas** program from the **practices** folder. Run the program and examine the output report. Notice that the **Column1** column contains raw data with values separated by various symbols. The **SCAN** function is used to extract the **ParkCode** and **ParkName** values.
- Examine the PROC CONTENTS report. Notice that **ParkCode** and **ParkName** have a length of 200, which is the same as **Column1**.

**Note:** When the **SCAN** function creates a new column, the new column will have the same length as the column listed as the first argument.

- The **ParkCode** column should include only the first four characters in the string. Add a **LENGTH** statement to define the length of **ParkCode** as 4.
- The length for the **ParkName** column can be optimized by determining the longest string and setting an appropriate length. Modify the **DATA** step to create a new column named **NameLength** that uses the **LENGTH** function to return the position of the last non-blank character for each value of **ParkName**.
- Use a **RETAIN** statement to create a new column named **MaxLength** that has an initial value of zero.
- Use an assignment statement and the **MAX** function to set the value of **MaxLength** to either the current value of **NameLength** or **MaxLength**, whichever is larger.
- Use the **END=** option in the **SET** statement to create a temporary variable in the PDV named **LastRow**. **LastRow** will be zero for all rows until the last row of the table, when it will be 1. Add an **IF-THEN** statement to write the value of **MaxLength** to the log if the value of **LastRow** is 1.

```
data parklookup;
  set pg2.np_unstructured_codes end=LastRow;
  ...
  if LastRow=1 then putlog MaxLength=;
run;
```