Practice

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

Level 1

1. Restructuring a Table Using the DATA Step: Wide to Narrow

The **pg2.np_2017Camping** table contains public use statistics for camping in 2017 from the National Park Service. To enable statistics to be calculated for all camping locations, restructure the table as a narrow table.

a. Open the **p207p01.sas** program in the **practices** folder. Highlight the PROC PRINT step and run the selected code. Note that the **Tent**, **RV**, and **Backcountry** columns contain visitor counts.

Obs	ParkName	Tent	RV	Backcountry
1	Acadia NP	152,586	55,812	1,597
2	Amistad NRA	0	11,019	0
3	Aniakchak NM & PRES	0	0	235
4	Apostle Islands NL	0	0	11,550

b. To convert this wide table to a narrow table, the DATA step must create a new column named **CampType** with the values *Tent*, *RV*, and *Backcountry*, and another new column named **CampCount** with the numeric counts. The DATA step includes statements to output a row for **CampType**='*Tent*'. Modify the DATA step to output additional rows for *RV* and *Backcountry*.

c. Add a LENGTH statement to ensure that the values of the **CampType** column are not truncated.

d. Run the DATA step. Confirm that each **ParkName** value has three rows corresponding to the *Tent*, *RV*, and *Backcountry* visitor counts.

	ParkName	CampType	CampCount
1	Acadia NP	Tent	152,586
2	Acadia NP	RV	55,812
3	Acadia NP	Backcountry	1,597
4	Amistad NRA	Tent	0
5	Amistad NRA	RV	11,019
6	Amistad NRA	Backcountry	0
7	Aniakchak NM...	Tent	0
8	Aniakchak NM...	RV	0

Level 2

2. Restructuring a Table Using the DATA Step: Narrow to Wide

The **pg2.np_2016Camping** table contains public use statistics for camping in 2016 from the National Park Service. To enable statistics to be calculated for individual camping locations, restructure the table as a wide table.

a. Examine the **pg2.np_2016Camping** table to determine the three unique values of the **CampType** column.

b. Write a DATA step to read **pg2.np_2016camping** and create **camping_wide**. Use IF-THEN/ELSE statements to assign **CampCount** to the **Tent**, **RV**, and **Backcountry** columns based on the value of **CampType**.

c. Use the RETAIN statement to hold the values of **ParkName**, **Tent**, **RV**, and **Backcountry** in the PDV each time that the PDV reinitializes.

d. Use the BY statement to group the data by **ParkName**. Add a subsetting IF statement to output the last row for each value of **ParkName**.

e. Keep the **ParkName**, **Tent**, **RV**, and **Backcountry** columns. Format **Tent**, **RV**, and **Backcountry** with commas.

f. Run the program and confirm that a column exists for each unique camping location (**Tent**, **RV**, and **Backcountry**).

	ParkName	Tent	RV	Backcountry
1	Acadia NP	152,811	46,629	1,324
2	Amistad NRA	38	8,265	0
3	Aniakchak NM...	0	0	235
4	Apostle Islands...	0	0	11,220
5	Arches NP	28,046	18,658	1,174
6	Assateague Isl...	40,826	20,735	973
7	Badlands NP	7,934	1,500	1,410
8	Bandelier NM	5,704	4,164	665

Challenge

3. Using Arrays to Restructure a Table

The **pg2.np_lodging_array** table contains statistics for stays at lodging facilities in 2015, 2016, and 2017. Create a table that contains two rows for each year (2015, 2016, and 2017), corresponding to the lodge counts for individual parks.

Note: An array enables you to perform the same action on a group of similar columns. In this example, **Lodge2015**, **Lodge2016**, and **Lodge2017** are all numeric columns that represent the same measure for different years. Using an array with a DO loop can simplify repetitive code. Access SAS Help for more information about arrays.

a. Examine the **np_lodging_array** table. In addition to **ParkName**, notice that there are three columns containing visitor lodging counts. **Lodge2015**, **Lodge2016**, and **Lodge2017** contain counts for visitors staying at lodges.

b. Open the **p207p03.sas** program from the **practices** folder. Run the program and confirm that the output table stacks the values of the **Lodge** columns.

c. Create a copy of the DATA step and paste it at the end of the program. Modify the second DATA step to use an array to simplify the repetitive processing.

1) Delete all statements between the FORMAT and RUN statements.

2) Add the following ARRAY statement after the FORMAT statement to define an array named **lodge** that includes the columns **Lodge2015**, **Lodge2016**, and **Lodge2017**.

```
array Lodge[2015:2017] Lodge2015-Lodge2017;
```

3) Add a DO loop with an index variable, **year**, that will loop three times for the values 2015 to 2017.

4) Inside the DO loop, perform the following actions:

   a) Create a column named **Stays** that will be equal to the value of each column in the **lodge** array.

      Note: The array name can be used in combination with the DO loop index variable to represent each column in the array. For example, **lodge[year]** will be replaced by **lodge[2015]** the first time through the DO loop. **Lodge[2015]** represents the first column in the **lodge** array, which is **Lodge2015**.

   b) Output the row to the new table.

d. Run the second DATA step and verify that the table includes three rows for each value of **ParkName**.

	ParkName	Stays	Year
1	Acadia NP	0	2015
2	Acadia NP	0	2016
3	Acadia NP	0	2017
4	Badlands NP	9,474	2015
5	Badlands NP	9,875	2016
6	Badlands NP	9,646	2017
7	Big Bend NP	50,747	2015
8	Big Bend NP	48,280	2016

End of Practices