**Using .pivot\_table() to count medals by type**

Rather than ranking countries by total medals won and showing that list, you may want to see a bit more detail. You can use a *pivot table* to compute how many separate bronze, silver and gold medals each country won. That pivot table can then be used to repeat the previous computation to rank by total medals won.

In this exercise, you will use .pivot\_table() first to aggregate the total medals by type. Then, you can use .sum() along the columns of the pivot table to produce a new column. When the modified pivot table is sorted by the total medals column, you can display the results from the last exercise with a bit more detail.

**INSTRUCTIONS**

* Construct a pivot table counted from the DataFrame medals aggregating by count. Use 'NOC' as the index, 'Athlete'for the values, and 'Medal' for the columns.
* Modify the DataFrame counted by adding a column counted['totals']. The new column 'totals' should contain the result of taking the sum along the columns (i.e., use .sum(axis='columns')).
* Overwrite the DataFrame counted by sorting it with the .sort\_values() method. Specify the keyword argument ascending=False.
* Print the first 15 rows of counted using .head(15).

# Construct the pivot table: counted

print(medals.head())

counted = medals.pivot\_table(index='NOC', columns='Medal', values='Athlete', aggfunc='count')

# Create the new column: counted['totals']

counted['totals'] = counted.sum(axis='columns')

# Sort counted by the 'totals' column

counted = counted.sort\_values('totals', ascending=False)

# Print the top 15 rows of counted

print(counted.head(15))