This notebook is an exercise in the Pandas (https://www.kaggle.com/learn/pandas) course. You can reference the tutorial at this link (https://www.kaggle.com/residentmario/summary-functions-and-maps).

Introduction

Now you are ready to get a deeper understanding of your data.

Run the following cell to load your data and some utility functions (including code to check your answers).

In [1]:

```
import pandas as pd
pd.set_option("display.max_rows", 5)
reviews = pd.read_csv("../input/wine-reviews/winemag-data-130k-v2.csv",
index_col=0)

from learntools.core import binder; binder.bind(globals())
from learntools.pandas.summary_functions_and_maps import *
print("Setup complete.")

reviews.head()
```

```
Setup complete.
```

Out[1]:

	country	description	designation	points	price	province	region_1	region_2	tast
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Keri O'Ke
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Rog
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Greg
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alex Pea
4	US	Much like the regular bottling from 2012, this...	Vintner's Reserve Wild Child Block	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Greg

Exercises

1.

What is the median of the `points` column in the `reviews` DataFrame?

In [2]:

```
median_points = reviews.points.median()

# Check your answer
q1.check()
```

Correct

In [3]:

```
#q1.hint()
#q1.solution()
```

2.

What countries are represented in the dataset? (Your answer should not include any duplicates.)

In [4]:

```
countries = reviews.country.unique()

# Check your answer
q2.check()
```

Correct

In [5]:

```
#q2.hint()
#q2.solution()
```

3.

How often does each country appear in the dataset? Create a Series `reviews_per_country` mapping countries to the count of reviews of wines from that country.

In [6]:

```
reviews_per_country = reviews.country.value_counts()

# Check your answer
q3.check()
```

```
Correct
```

In [7]:

```
#q3.hint()
#q3.solution()
```

4.

Create variable `centered_price` containing a version of the `price` column with the mean price subtracted.

(Note: this 'centering' transformation is a common preprocessing step before applying various machine learning algorithms.)

In [8]:

```
centered_price =  reviews.price - reviews.price.mean()

# Check your answer
q4.check()
```

Correct

In [9]:

```
#q4.hint()
#q4.solution()
```

5.

I'm an economical wine buyer. Which wine is the "best bargain"? Create a variable `bargain_wine` with the title of the wine with the highest points-to-price ratio in the dataset.

In [10]:

```
bargain_wine = reviews.loc[ (reviews.points / reviews.price).idxmax(), 'title']

# Check your answer
q5.check()
```

Correct

In [11]:

```
#q5.hint()
#q5.solution()
```

6.

There are only so many words you can use when describing a bottle of wine. Is a wine more likely to be "tropical" or "fruity"? Create a Series `descriptor_counts` counting how many times each of these two words appears in the `description` column in the dataset.

In [12]:

```
n_trop = reviews.description.map(lambda desc: "tropical" in desc).sum()
n_fruity = reviews.description.map(lambda desc: "fruity" in desc).sum()
descriptor_counts = pd.Series([n_trop, n_fruity], index=['tropical', 'fruity'])

# Check your answer
q6.check()
```

Correct

In [13]:

```
#q6.hint()
#q6.solution()
```

7.

We'd like to host these wine reviews on our website, but a rating system ranging from 80 to 100 points is too hard to understand - we'd like to translate them into simple star ratings. A score of 95 or higher counts as 3 stars, a score of at least 85 but less than 95 is 2 stars. Any other score is 1 star.

Also, the Canadian Vintners Association bought a lot of ads on the site, so any wines from Canada should automatically get 3 stars, regardless of points.

Create a series `star_ratings` with the number of stars corresponding to each review in the dataset.

In [14]:

```
def star_num(index):
    if index.country == 'Canada':
        return 3
    elif index.points >= 95:
        return 3
    elif index.points >= 85:
        return 2
    else:
        return 1

star_ratings = reviews.apply(star_num, axis='columns')

# Check your answer
q7.check()
```

Correct

In [16]:

```
#q7.hint()
#q7.solution()
```

Keep going

Continue to **grouping and sorting (https://www.kaggle.com/residentmario/grouping-and-sorting)**.

Have questions or comments? Visit the Learn Discussion forum (https://www.kaggle.com/learn-forum/161299) to chat with other Learners.