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.

⊞ Code ⊞ Text

```
import pandas as pd
pd.set_option("display.max_rows", 5)
reviews = pd.read_csv("https://raw.githubusercontent.com/ltdaovn/dataset/master/wine-reviews/winemag-data-130k-v2.csv", index_col=0)
reviews.head()
```

	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle	
0	Italy	Aromas include tropical fruit, broom, brimston...	Vulkà Bianco	87	NaN	Sicily & Sardinia	Etna	NaN	Kerin O’Keefe	@kerinokeefe	N 2013 E (
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger	Quin Avi Avi (D
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	Rain 2013 (Willa \
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN	St. Re H Ries
4	US	Much like the regular bottling from 2012	Vintner's Reserve Wild	87	65.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine	S C Vir

Exercises

1.

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

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

2.

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

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

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.

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

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.)

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

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.

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

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.

```
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'])
descriptor_counts
```

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.

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

star_ratings = reviews.apply(stars, axis='columns')
star_ratings
```