Introduction

In this set of exercises we will work with the Wine Reviews dataset

Run the following cell to load your data.

```
import pandas as pd

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

print("Setup complete.")
```

Look at an overview of your data by running the following line.

```
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	Quint 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 V
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 Ha 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	C Vir

Exercises

1.

Select the `description` column from `reviews` and assign the result to the variable `desc`.

```
# Your code here
desc = reviews.description
```

Follow-up question: what type of object is `desc`? If you're not sure, you can check by calling Python's `type` function: `type(desc)`.

2.

Select the first value from the description column of `reviews`, assigning it to variable `first_description`.

```
first_description = reviews.description.iloc[0]
first_description
```

3.

Select the first row of data (the first record) from `reviews`, assigning it to the variable `first_row`.

```
first_row = reviews.iloc[0]
first_row
```

4.

Select the first 10 values from the `description` column in `reviews`, assigning the result to variable `first_descriptions`.

Hint: format your output as a pandas Series.

```
first_descriptions = reviews.description.iloc[:10]
first_descriptions
```

5.

Select the records with index labels `1, 2, 3, 5,` and `8`, assigning the result to the variable `sample_reviews`.

In other words, generate the following DataFrame:

	country	description	designation	points	price	province	region_1	region_2	taster_name	taster_twitter_handle
1	Portugal	This is ripe and fruity, a wine that is smooth...	Avidagos	87	15.0	Douro	NaN	NaN	Roger Voss	@vossroger
2	US	Tart and snappy, the flavors of lime flesh and...	NaN	87	14.0	Oregon	Willamette Valley	Willamette Valley	Paul Gregutt	@paulgwine
3	US	Pineapple rind, lemon pith and orange blossom ...	Reserve Late Harvest	87	13.0	Michigan	Lake Michigan Shore	NaN	Alexander Peartree	NaN
5	Spain	Blackberry and raspberry aromas show a typical...	Ars In Vitro	87	15.0	Northern Spain	Navarra	NaN	Michael Schachner	@wineschach
8	Germany	Savory dried thyme notes accent sunnier flavor...	Shine	87	12.0	Rheinhessen	NaN	NaN	Anna Lee C. Iijima	NaN

```
sample_reviews = reviews.iloc[[1,2,3,5,8]]
sample_reviews
```

6.

Create a variable `df` containing the `country`, `province`, `region_1`, and `region_2` columns of the records with the index labels `0`, `1`, `10`, and `100`. In other words, generate the following DataFrame:

	country	province	region_1	region_2
0	Italy	Sicily & Sardinia	Etna	NaN
1	Portugal	Douro	NaN	NaN
10	US	California	Napa Valley	Napa
100	US	New York	Finger Lakes	Finger Lakes

```
cols = ['country', 'province', 'region_1', 'region_2']
indices = [0, 1, 10, 100]
df = reviews.loc[indices, cols]
df
```

7.

Create a variable `df` containing the `country` and `variety` columns of the first 100 records.

Hint: you may use `loc` or `iloc`. When working on the answer this question and the several of the ones that follow, keep the following "gotcha" described in the tutorial:

> `iloc` uses the Python stdlib indexing scheme, where the first element of the range is included and the last one excluded. `loc`, meanwhile, indexes inclusively.

> This is particularly confusing when the DataFrame index is a simple numerical list, e.g. `0,...,1000`. In this case `df.iloc[0:1000]` will return 1000 entries, while `df.loc[0:1000]` return 1001 of them! To get 1000 elements using `loc`, you will need to go one lower and ask for `df.iloc[0:999]`.

```
cols = ['country', 'variety']
df = reviews.loc[:99, cols]
df
```

8.

Create a DataFrame `italian_wines` containing reviews of wines made in `Italy`. Hint: `reviews.country` equals what?

```
italian_wines = reviews[reviews.country == 'Italy']
italian_wines
```

9.

Create a DataFrame `top_oceania_wines` containing all reviews with at least 95 points (out of 100) for wines from Australia or New Zealand.

```
top_oceania_wines = reviews.loc[
    (reviews.country.isin(['Australia','New Zealand']))
    & (reviews.points >= 95)]
top_oceania_wines
```