Assignment 3 - Pandas Data Analysis Practice

This assignment is a part of the course "Data Analysis with Python: Zero to Pandas"

In this assignment, you'll get to practice some of the concepts and skills covered in this tutorial: https://jovian.ai/aakashns/python-pandas-data-analysis

As you go through this notebook, you will find a **???** in certain places. To complete this assignment, you must replace all the **???** with appropriate values, expressions or statements to ensure that the notebook runs properly end-to-end.

Some things to keep in mind:

- Make sure to run all the code cells, otherwise you may get errors like `NameError` for undefined variables.
- Do not change variable names, delete cells or disturb other existing code. It may cause problems during evaluation.
- In some cases, you may need to add some code cells or new statements before or after the line of code containing the **???**.
- Since you'll be using a temporary online service for code execution, save your work by running `jovian.commit` at regular intervals.
- Questions marked **(Optional)** will not be considered for evaluation, and can be skipped. They are for your learning.

You can make submissions on this page: https://jovian.ai/learn/data-analysis-with-python-zero-to-pandas/assignment/assignment-3-pandas-practice

If you are stuck, you can ask for help on the community forum: https://jovian.ai/forum/t/assignment-3-pandas-practice/11225/3. You can get help with errors or ask for hints, describe your approach in simple words, link to documentation, but **please don't ask for or share the full working answer code** on the forum.

How to run the code and save your work

The recommended way to run this notebook is to click the "Run" button at the top of this page, and select "Run on Binder". This will run the notebook on mybinder.org, a free online service for running Jupyter notebooks.

Before starting the assignment, let's save a snapshot of the assignment to your Jovian.ai profile, so that you can access it later, and continue your work.

```
import jovian
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

```
# Run the next line to install Pandas
!pip install pandas --upgrade
```

```
Requirement already satisfied: pandas in /opt/conda/lib/python3.9/site-packages (1.3.4)
Collecting pandas
  Downloading pandas-2.1.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(12.3 MB)
     |████████████████████████████████| 12.3 MB 24.5 MB/s
Requirement already satisfied: pytz>=2020.1 in /opt/conda/lib/python3.9/site-packages
(from pandas) (2021.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /opt/conda/lib/python3.9/site-
packages (from pandas) (2.8.2)
Collecting tzdata>=2022.1
  Downloading tzdata-2023.3-py2.py3-none-any.whl (341 kB)
     |████████████████████████████████| 341 kB 94.7 MB/s
Collecting numpy>=1.22.4
  Downloading numpy-1.26.0-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(18.2 MB)
     |████████████████████████████████| 18.2 MB 73.7 MB/s
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.9/site-packages (from
python-dateutil>=2.8.2->pandas) (1.16.0)
Installing collected packages: tzdata, numpy, pandas
  Attempting uninstall: numpy
    Found existing installation: numpy 1.20.3
    Uninstalling numpy-1.20.3:
      Successfully uninstalled numpy-1.20.3
  Attempting uninstall: pandas
    Found existing installation: pandas 1.3.4
    Uninstalling pandas-1.3.4:
      Successfully uninstalled pandas-1.3.4
ERROR: pip's dependency resolver does not currently take into account all the packages
that are installed. This behaviour is the source of the following dependency conflicts.
scipy 1.7.2 requires numpy<1.23.0,>=1.16.5, but you have numpy 1.26.0 which is
incompatible.
numba 0.54.1 requires numpy<1.21,>=1.17, but you have numpy 1.26.0 which is
incompatible.
Successfully installed numpy-1.26.0 pandas-2.1.1 tzdata-2023.3
```

```python
import pandas as pd
```

In this assignment, we're going to analyze an operate on data from a CSV file. Let's begin by downloading the CSV file.

```python
from urllib.request import urlretrieve

urlretrieve('https://gist.githubusercontent.com/aakashns/28b2e504b3350afd9bdb157893f972
            'countries.csv')
```

```
('countries.csv', <http.client.HTTPMessage at 0x7fc666a67730>)
```

Let's load the data from the CSV file into a Pandas data frame.

```
countries_df = pd.read_csv('countries.csv')
```

```
countries_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.987
1	Albania	Europe	2877800.0	78.57	2.89	11803.431
2	Algeria	Africa	43851043.0	76.88	1.90	13913.839
3	Andorra	Europe	77265.0	83.73	NaN	NaN
4	Angola	Africa	32866268.0	61.15	NaN	5819.495
...	...	...	...	...	...	...
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.884
206	Western Sahara	Africa	597330.0	70.26	NaN	NaN
207	Yemen	Asia	29825968.0	66.12	0.70	1479.147
208	Zambia	Africa	18383956.0	63.89	2.00	3689.251
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.775

210 rows × 6 columns

Q1: How many countries does the dataframe contain?

Hint: Use the `.shape` method.

```
num_countries = countries_df['location'].shape
```

```
print('There are {} countries in the dataset'.format(num_countries))
```

```
There are (210,) countries in the dataset
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q2: Retrieve a list of continents from the dataframe?

Hint: Use the `.unique` method of a series.

```
continents = countries_df['continent'].unique()
```

```
continents
```

```
array(['Asia', 'Europe', 'Africa', 'North America', 'South America',
       'Oceania'], dtype=object)
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q3: What is the total population of all the countries listed in this dataset?

```
total_population = countries_df['population'].sum()
```

```
print('The total population is {}.'.format(int(total_population)))
```

```
The total population is 7757980095.
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q: (Optional) What is the overall life expectancy across in the world?

Hint: You'll need to take a weighted average of life expectancy using populations as weights.

```
import pandas as pd

# Assuming you have already loaded your dataframe 'countries_df'

# Calculate the weighted sum of life expectancy
weighted_sum = (countries_df['population'] * countries_df['life_expectancy']).sum()

# Calculate the total population
total_population = countries_df['population'].sum()

# Calculate the overall life expectancy
overall_life_expectancy = weighted_sum / total_population
```

```
overall_life_expectancy
```

```
72.72165193409664
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q4: Create a dataframe containing 10 countries with the highest population.

Hint: Chain the `sort_values` and `head` methods.

```
most_populous_df = countries_df.sort_values(by='population', ascending=False, ignore_ir
```

```
most_populous_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita
0	China	Asia	1.439324e+09	76.91	4.34	15308.712
1	India	Asia	1.380004e+09	69.66	0.53	6426.674
2	United States	North America	3.310026e+08	78.86	2.77	54225.446
3	Indonesia	Asia	2.735236e+08	71.72	1.04	11188.744
4	Pakistan	Asia	2.208923e+08	67.27	0.60	5034.708
5	Brazil	South America	2.125594e+08	75.88	2.20	14103.452
6	Nigeria	Africa	2.061396e+08	54.69	NaN	5338.454
7	Bangladesh	Asia	1.646894e+08	72.59	0.80	3523.984
8	Russia	Europe	1.459345e+08	72.58	8.05	24765.954
9	Mexico	North America	1.289328e+08	75.05	1.38	17336.469

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q5: Add a new column in `countries_df` to record the overall GDP per country (product of population & per capita GDP).

```
countries_df['gdp'] = countries_df['population'] * countries_df['gdp_per_capita']
```

```
countries_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	gdp
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.987	7.022622e+10
1	Albania	Europe	2877800.0	78.57	2.89	11803.431	3.396791e+10
2	Algeria	Africa	43851043.0	76.88	1.90	13913.839	6.101364e+11
3	Andorra	Europe	77265.0	83.73	NaN	NaN	NaN
4	Angola	Africa	32866268.0	61.15	NaN	5819.495	1.912651e+11

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	gdp
...	...	...	...	...	...	...	...
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.884	6.007624e+11
206	Western Sahara	Africa	597330.0	70.26	NaN	NaN	NaN
207	Yemen	Asia	29825968.0	66.12	0.70	1479.147	4.411699e+10
208	Zambia	Africa	18383956.0	63.89	2.00	3689.251	6.782303e+10
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.775	2.823622e+10

210 rows × 7 columns

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q: (Optional) Create a dataframe containing 10 countries with the lowest GDP per capita, among the counties with population greater than 100 million.

```
lowest_gdp_countries_df = countries_df[countries_df['population'] > 100].sort_values(by
```

```
lowest_gdp_countries_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	gdp
38	Central African Republic	Africa	4829764.0	53.28	1.0	661.240	3.193633e+09
32	Burundi	Africa	11890781.0	61.58	0.8	702.225	8.350004e+09
111	Liberia	Africa	5057677.0	64.10	0.8	752.788	3.807359e+09
52	Democratic Republic of Congo	Africa	89561404.0	60.68	NaN	808.133	7.237753e+10
140	Niger	Africa	24206636.0	62.42	0.3	926.000	2.241534e+10
118	Malawi	Africa	19129955.0	64.26	1.3	1095.042	2.094810e+10
132	Mozambique	Africa	31255435.0	60.85	0.7	1136.103	3.550939e+10
168	Sierra Leone	Africa	7976985.0	54.70	NaN	1390.300	1.109040e+10
43	Comoros	Africa	869595.0	64.32	2.2	1413.890	1.229512e+09
117	Madagascar	Africa	27691019.0	67.04	0.2	1416.440	3.922267e+10

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q6: Create a data frame that counts the number countries in each continent?

Hint: Use `groupby`, select the `location` column and aggregate using `count`.

```
country_counts_df = countries_df.groupby('continent')['location'].count().reset_index()
country_counts_df = country_counts_df.set_index('continent')
```

```
country_counts_df
```

continent	location
Africa	55
Asia	47
Europe	51
North America	36
Oceania	8
South America	13

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q7: Create a data frame showing the total population of each continent.

Hint: Use `groupby`, select the population column and aggregate using `sum`.

```
continent_populations_df = countries_df.groupby('continent')['population'].sum().reset_
continent_populations_df = continent_populations_df.set_index('continent')
```

```
continent_populations_df
```

continent	population
Africa	1.339424e+09
Asia	4.607388e+09
Europe	7.485062e+08
North America	5.912425e+08
Oceania	4.095832e+07
South America	4.304611e+08

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
```

```
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Let's download another CSV file containing overall Covid-19 stats for various countires, and read the data into another Pandas data frame.

```
urlretrieve('https://gist.githubusercontent.com/aakashns/b2a968a6cfd9fbbb0ff3d6bd0f2626
            'covid-countries-data.csv')
```

```
('covid-countries-data.csv', <http.client.HTTPMessage at 0x7fc6669f0a60>)
```

```
covid_data_df = pd.read_csv('covid-countries-data.csv')
```

```
covid_data_df
```

	location	total_cases	total_deaths	total_tests
0	Afghanistan	38243.0	1409.0	NaN
1	Albania	9728.0	296.0	NaN
2	Algeria	45158.0	1525.0	NaN
3	Andorra	1199.0	53.0	NaN
4	Angola	2729.0	109.0	NaN
...	...	...	...	...
207	Western Sahara	766.0	1.0	NaN
208	World	26059065.0	863535.0	NaN
209	Yemen	1976.0	571.0	NaN
210	Zambia	12415.0	292.0	NaN
211	Zimbabwe	6638.0	206.0	97272.0

212 rows × 4 columns

Q8: Count the number of countries for which the `total_tests` data is missing.

Hint: Use the `.isna` method.

```
total_tests_missing = covid_data_df['total_tests'].isna().sum()
```

```
print("The data for total tests is missing for {} countries.".format(int(total_tests_mi
```

```
The data for total tests is missing for 122 countries.
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Let's merge the two data frames, and compute some more metrics.

Q9: Merge `countries_df` with `covid_data_df` on the `location` column.

*Hint: Use the `.merge` method on `countries_df` .

```
combined_df = countries_df.merge(covid_data_df, on='location')
```

```
combined_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	total_cases	tota
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.987	38243.0	
1	Albania	Europe	2877800.0	78.57	2.89	11803.431	9728.0	
2	Algeria	Africa	43851043.0	76.88	1.90	13913.839	45158.0	
3	Andorra	Europe	77265.0	83.73	NaN	NaN	1199.0	
4	Angola	Africa	32866268.0	61.15	NaN	5819.495	2729.0	
...	...	...	...	...	...	...	...	
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.884	1046.0	
206	Western Sahara	Africa	597330.0	70.26	NaN	NaN	766.0	
207	Yemen	Asia	29825968.0	66.12	0.70	1479.147	1976.0	
208	Zambia	Africa	18383956.0	63.89	2.00	3689.251	12415.0	
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.775	6638.0	

210 rows × 9 columns

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q10: Add columns `tests_per_million` , `cases_per_million` and `deaths_per_million` into `combined_df` .

```
combined_df['tests_per_million'] = combined_df['total_tests'] * 1e6 / combined_df['popu
```

```
combined_df['cases_per_million'] =  combined_df['total_cases'] * 1e6 / combined_df['pop
```

```
combined_df['deaths_per_million'] = combined_df['total_deaths'] * 1e6 / combined_df['pc
```

```
combined_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	total_cases	tota

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	total_cases	tota
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.987	38243.0	
1	Albania	Europe	2877800.0	78.57	2.89	11803.431	9728.0	
2	Algeria	Africa	43851043.0	76.88	1.90	13913.839	45158.0	
3	Andorra	Europe	77265.0	83.73	NaN	NaN	1199.0	
4	Angola	Africa	32866268.0	61.15	NaN	5819.495	2729.0	
...	...	...	...	...	...	...	...	
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.884	1046.0	
206	Western Sahara	Africa	597330.0	70.26	NaN	NaN	766.0	
207	Yemen	Asia	29825968.0	66.12	0.70	1479.147	1976.0	
208	Zambia	Africa	18383956.0	63.89	2.00	3689.251	12415.0	
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.775	6638.0	

210 rows × 12 columns

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q11: Create a dataframe with 10 countires that have highest number of tests per million people.

```
highest_tests_df = combined_df.sort_values(by='tests_per_million', ascending=False).hea
```

```
highest_tests_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	total_cases	to
197	United Arab Emirates	Asia	9890400.0	77.97	1.200	67293.483	71540.0	
14	Bahrain	Asia	1701583.0	77.29	2.000	43290.705	52440.0	
115	Luxembourg	Europe	625976.0	82.25	4.510	94277.965	7928.0	
122	Malta	Europe	441539.0	82.53	4.485	36513.323	1931.0	
53	Denmark	Europe	5792203.0	80.90	2.500	46682.515	17195.0	
96	Israel	Asia	8655541.0	82.97	2.990	33132.320	122539.0	
89	Iceland	Europe	341250.0	82.99	2.910	46482.958	2121.0	
157	Russia	Europe	145934460.0	72.58	8.050	24765.954	1005000.0	
199	United States	North America	331002647.0	78.86	2.770	54225.446	6114406.0	
10	Australia	Oceania	25499881.0	83.44	3.840	44648.710	25923.0	

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q12: Create a dataframe with 10 countires that have highest number of positive cases per million people.

```
highest_cases_df = combined_df.sort_values(by='cases_per_million', ascending=False).hea
```

```
highest_cases_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	total_cases	total_
155	Qatar	Asia	2881060.0	80.23	1.20	116935.600	119206.0	
14	Bahrain	Asia	1701583.0	77.29	2.00	43290.705	52440.0	
147	Panama	North America	4314768.0	78.51	2.30	22267.037	94084.0	2
40	Chile	South America	19116209.0	80.18	2.11	22767.037	414739.0	11
162	San Marino	Europe	33938.0	84.97	3.80	56861.470	735.0	
9	Aruba	North America	106766.0	76.29	NaN	35973.781	2211.0	
105	Kuwait	Asia	4270563.0	75.49	2.00	65530.537	86478.0	
150	Peru	South America	32971846.0	76.74	1.60	12236.706	663437.0	29
27	Brazil	South America	212559409.0	75.88	2.20	14103.452	3997865.0	123
199	United States	North America	331002647.0	78.86	2.77	54225.446	6114406.0	185

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Q13: Create a dataframe with 10 countires that have highest number of deaths cases per million people?

```
highest_deaths_df = combined_df.sort_values(by='deaths_per_million', ascending=False).h
```

```
highest_deaths_df
```

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	total_cases	total_
162	San Marino	Europe	33938.0	84.97	3.80	56861.470	735.0	
150	Peru	South America	32971846.0	76.74	1.60	12236.706	663437.0	2
18	Belgium	Europe	11589616.0	81.63	5.64	42658.576	85817.0	

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	total_cases	total_
3	Andorra	Europe	77265.0	83.73	NaN	NaN	1199.0	
177	Spain	Europe	46754783.0	83.56	2.97	34272.360	479554.0	2
198	United Kingdom	Europe	67886004.0	81.32	2.54	39753.244	338676.0	4
40	Chile	South America	19116209.0	80.18	2.11	22767.037	414739.0	1
97	Italy	Europe	60461828.0	83.51	3.18	35220.084	271515.0	3
27	Brazil	South America	212559409.0	75.88	2.20	14103.452	3997865.0	12
182	Sweden	Europe	10099270.0	82.80	2.22	46949.283	84532.0	

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

(Optional) Q: Count number of countries that feature in both the lists of "highest number of tests per million" and "highest number of cases per million".

```
# Create DataFrames for the highest tests per million and highest cases per million
highest_tests_df = combined_df.sort_values(by='tests_per_million', ascending=False).hea
highest_cases_df = combined_df.sort_values(by='cases_per_million', ascending=False).hea
```

```
# Perform an inner merge to find common countries
common_countries_df = pd.merge(highest_tests_df, highest_cases_df, on='location', how='
```

```
# Count the number of common countries
common_country_count = len(common_countries_df)
print("Number of countries in both lists:", common_country_count)
```

```
Number of countries in both lists: 2
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

(Optional) Q: Count number of countries that feature in both the lists "20 countries with lowest GDP per capita" and "20 countries with the lowest number of hospital beds per thousand population". Only consider countries with a population higher than 10 million while creating the list.

```

```

```
import jovian
```

```
jovian.commit(project='pandas-practice-assignment', environment=None)
```

```
[jovian] Updating notebook "ay29020/pandas-practice-assignment" on https://jovian.com
[jovian] Committed successfully! https://jovian.com/ay29020/pandas-practice-assignment
```

```
'https://jovian.com/ay29020/pandas-practice-assignment'
```

Submission

Congratulations on making it this far! You've reached the end of this assignment, and you just completed your first real-world data analysis problem. It's time to record one final version of your notebook for submission.

Make a submission here by filling the submission form: https://jovian.ai/learn/data-analysis-with-python-zero-to-pandas/assignment/assignment-3-pandas-practice

Also make sure to help others on the forum: https://jovian.ai/forum/t/assignment-3-pandas-practice/11225/2