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 this tutorial: https://jovian.ml/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.ml/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.ml/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 staring the assignment, let's save a snapshot of the assignment to your Jovian.ml profile, so that you can access it later, and continue your work.

In [3]:
```
import jovian
```

In [4]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[4]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

In [5]:
```
# Run the next line to install Pandas
!pip install pandas
```
```
Collecting pandas
  Downloading pandas-1.1.2-cp37-cp37m-manylinux1_x86_64.whl (10.5 MB)
     |████████████████████████████████| 10.5 MB 3.2 MB/s eta 0:00:01
Collecting numpy>=1.15.4
  Downloading numpy-1.19.2-cp37-cp37m-manylinux2010_x86_64.whl (14.5 MB)
     |████████████████████████████████| 14.5 MB 60.3 MB/s eta 0:00:01
Requirement already satisfied: python-dateutil>=2.7.3 in /srv/conda/env
s/notebook/lib/python3.7/site-packages (from pandas) (2.8.1)
Collecting pytz>=2017.2
  Downloading pytz-2020.1-py2.py3-none-any.whl (510 kB)
     |████████████████████████████████| 510 kB 40.2 MB/s eta 0:00:01
Requirement already satisfied: six>=1.5 in /srv/conda/envs/notebook/lib/
python3.7/site-packages (from python-dateutil>=2.7.3->pandas) (1.15.0)
Installing collected packages: numpy, pytz, pandas
Successfully installed numpy-1.19.2 pandas-1.1.2 pytz-2020.1
```

In [7]:
```
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.

In [8]:
```
from urllib.request import urlretrieve

urlretrieve('https://hub.jovian.ml/wp-content/uploads/2020/09/countries.
csv',
            'countries.csv')
```
Out[8]:
```
('countries.csv', <http.client.HTTPMessage at 0x7f9256d93590>)
```

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

In [9]:
```
countries_df = pd.read_csv('countries.csv')
```

In [10]:
```
countries_df
```
Out[10]:

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capi
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.98
1	Albania	Europe	2877800.0	78.57	2.89	11803.43
2	Algeria	Africa	43851043.0	76.88	1.90	13913.83
3	Andorra	Europe	77265.0	83.73	NaN	Na
4	Angola	Africa	32866268.0	61.15	NaN	5819.49
...	...	...	...	...	...	
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.88
206	Western Sahara	Africa	597330.0	70.26	NaN	Na
207	Yemen	Asia	29825968.0	66.12	0.70	1479.14
208	Zambia	Africa	18383956.0	63.89	2.00	3689.25
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.77

210 rows × 6 columns

Q: How many countries does the dataframe contain?

Hint: Use the `.shape` method.

In [14]:
```
num_countries =  countries_df.location.shape
```

In [15]:
```
print('There are {} countries in the dataset'.format(num_countries))
```
```
There are (210,) countries in the dataset
```

In [16]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[16]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

Q: Retrieve a list of continents from the dataframe?

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

In [17]:
```
continents = countries_df['continent'].unique()
```

In [19]:
```
continents
```
Out[19]:
```
array(['Asia', 'Europe', 'Africa', 'North America', 'South America',
       'Oceania'], dtype=object)
```

In [20]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[20]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

In [21]:
```
total_population = countries_df['population'].sum()
```

In [22]:
```
print('The total population is {}.'.format(int(total_population)))
```
```
The total population is 7757980095.
```

In [24]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[24]:
```
'https://jovian.ml/kirankumarmb0002/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.

In [25]:
```
expectancy=countries_df[['life_expectancy']].mean()
```

In [27]:
```
print('The life expectancy is {}.'.format(int(expectancy )))
```
```
The life expectancy is 73.
```

In [28]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[28]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

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

In [49]:
```
most_populous_df =countries_df.sort_values(['population'],ascending=Fals
e).head(10)
```

In [50]:
```
most_populous_df
```
Out[50]:

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_ca
41	China	Asia	1.439324e+09	76.91	4.34	15308
90	India	Asia	1.380004e+09	69.66	0.53	6426
199	United States	North America	3.310026e+08	78.86	2.77	54225
91	Indonesia	Asia	2.735236e+08	71.72	1.04	11188
145	Pakistan	Asia	2.208923e+08	67.27	0.60	5034
27	Brazil	South America	2.125594e+08	75.88	2.20	14103
141	Nigeria	Africa	2.061396e+08	54.69	NaN	5338
15	Bangladesh	Asia	1.646894e+08	72.59	0.80	3523
157	Russia	Europe	1.459345e+08	72.58	8.05	24765
125	Mexico	North America	1.289328e+08	75.05	1.38	17336

In [51]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[51]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

In [52]:
```
countries_df['gdp'] =countries_df['gdp'] = countries_df['population']*co
untries_df['gdp_per_capita']
```

In [54]:
```
countries_df
```
Out[54]:

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capi
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.98
1	Albania	Europe	2877800.0	78.57	2.89	11803.43
2	Algeria	Africa	43851043.0	76.88	1.90	13913.83
3	Andorra	Europe	77265.0	83.73	NaN	Na
4	Angola	Africa	32866268.0	61.15	NaN	5819.49
...	...	...	...	...	...	
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.88
206	Western Sahara	Africa	597330.0	70.26	NaN	Na
207	Yemen	Asia	29825968.0	66.12	0.70	1479.14
208	Zambia	Africa	18383956.0	63.89	2.00	3689.25
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.77

210 rows × 7 columns

In [55]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[55]:
```
'https://jovian.ml/kirankumarmb0002/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.

In [58]:
```
lowest_GDP=countries_df[['location','population','gdp_per_capita']].sort
_values(['gdp_per_capita'],ascending=True)
```

In [59]:
```
lowest_GDP[lowest_GDP.population>100000000].head(10)
```
Out[59]:

	location	population	gdp_per_capita
63	Ethiopia	1.149636e+08	1729.927
15	Bangladesh	1.646894e+08	3523.984
145	Pakistan	2.208923e+08	5034.708
141	Nigeria	2.061396e+08	5338.454
90	India	1.380004e+09	6426.674
151	Philippines	1.095811e+08	7599.188
58	Egypt	1.023334e+08	10550.206
91	Indonesia	2.735236e+08	11188.744
27	Brazil	2.125594e+08	14103.452
41	China	1.439324e+09	15308.712

In [60]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[60]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

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

In [62]:
```
country_counts_df = countries_df.groupby('continent')['location'].count
()
```

In [63]:
```
country_counts_df
```
Out[63]:
```
continent
Africa           55
Asia             47
Europe           51
North America    36
Oceania           8
South America    13
Name: location, dtype: int64
```

In [64]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[64]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

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

In [67]:
```
continent_populations_df = countries_df.groupby('continent')['populatio
n'].sum()
```

In [68]:
```
continent_populations_df
```
Out[68]:
```
continent
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
Name: population, dtype: float64
```

In [69]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[69]:
```
'https://jovian.ml/kirankumarmb0002/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.

In [70]:
```
urlretrieve('https://hub.jovian.ml/wp-content/uploads/2020/09/covid-coun
tries-data.csv',
            'covid-countries-data.csv')
```
Out[70]:
```
('covid-countries-data.csv', <http.client.HTTPMessage at 0x7f9256a24490
>)
```

In [71]:
```
covid_data_df = pd.read_csv('covid-countries-data.csv')
```

In [72]:
```
covid_data_df
```
Out[72]:

	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

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

Hint: Use the `.isna` method.

In [76]:
```
total_tests_missing = covid_data_df['total_tests'].isna()
m=total_tests_missing.value_counts(sort=False)
total_tests_missing=m[False]
```

In [77]:
```
print("The data for total tests is missing for {} countries.".format(int
(total_tests_missing)))
```
```
The data for total tests is missing for 90 countries.
```

In [78]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[78]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

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

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

In [80]:
```
combined_df =pd.merge(countries_df,covid_data_df,on='location')
```

In [81]:
```
combined_df
```
Out[81]:

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capi
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.98
1	Albania	Europe	2877800.0	78.57	2.89	11803.43
2	Algeria	Africa	43851043.0	76.88	1.90	13913.83
3	Andorra	Europe	77265.0	83.73	NaN	Na
4	Angola	Africa	32866268.0	61.15	NaN	5819.49
...	...	...	...	...	...	
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.88
206	Western Sahara	Africa	597330.0	70.26	NaN	Na
207	Yemen	Asia	29825968.0	66.12	0.70	1479.14
208	Zambia	Africa	18383956.0	63.89	2.00	3689.25
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.77

210 rows × 10 columns

In [82]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[82]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

In [ ]:
```
combined_df['tests_per_million'] = combined_df['total_tests'] * 1e6 / co
mbined_df['population']
```

In [83]:
```
combined_df['cases_per_million'] = combined_df['total_cases'] * 1e6 / co
mbined_df['population']
```

In [84]:
```
combined_df['deaths_per_million'] = combined_df['total_deaths'] * 1e6 /
combined_df['population']
```

In [85]:
```
combined_df
```
Out[85]:

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capi
0	Afghanistan	Asia	38928341.0	64.83	0.50	1803.98
1	Albania	Europe	2877800.0	78.57	2.89	11803.43
2	Algeria	Africa	43851043.0	76.88	1.90	13913.83
3	Andorra	Europe	77265.0	83.73	NaN	Na
4	Angola	Africa	32866268.0	61.15	NaN	5819.49
...	...	...	...	...	...	
205	Vietnam	Asia	97338583.0	75.40	2.60	6171.88
206	Western Sahara	Africa	597330.0	70.26	NaN	Na
207	Yemen	Asia	29825968.0	66.12	0.70	1479.14
208	Zambia	Africa	18383956.0	63.89	2.00	3689.25
209	Zimbabwe	Africa	14862927.0	61.49	1.70	1899.77

210 rows × 12 columns

In [86]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[86]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

In [97]:
```
highest_tests_df = combined_df[['location','total_tests']]
a=highest_tests_df.sort_values(['total_tests'],ascending=False)
highest_tests_df =a.head(10)
```

In [98]:
```
highest_tests_df
```
Out[98]:

	location	total_tests
199	United States	83898416.0
90	India	44337201.0
157	Russia	37176827.0
198	United Kingdom	13447568.0
73	Germany	11208091.0
193	Turkey	7247935.0
197	United Arab Emirates	7177430.0
177	Spain	6416533.0
10	Australia	6255797.0
35	Canada	5545473.0

In [99]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[99]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

In [111]:
```
highest_cases_df = combined_df.sort_values(['cases_per_million'],ascendi
ng=False)
highest_cases_df =b.head(10)
```

In [112]:
```
highest_cases_df
```
Out[112]:

	location	cases_per_million
155	Qatar	41375.743650
14	Bahrain	30818.361490
147	Panama	21805.112117
40	Chile	21695.671982
162	San Marino	21657.139490
9	Aruba	20708.839893
105	Kuwait	20249.789079
150	Peru	20121.318048
27	Brazil	18808.224105
199	United States	18472.377957

In [113]:
```
jovian.commit(project='pandas-practice-assignment', environment=None)
```
```
[jovian] Attempting to save notebook..
[jovian] Updating notebook "kirankumarmb0002/pandas-practice-assignment" on https://jovian.ml/
[jovian] Uploading notebook..
[jovian] Committed successfully! https://jovian.ml/kirankumarmb0002/pand
as-practice-assignment
```
Out[113]:
```
'https://jovian.ml/kirankumarmb0002/pandas-practice-assignment'
```

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

In [116]:
```
highest_deaths_df = combined_df.sort_values(['deaths_per_million'],ascen
ding=False)
highest_deaths_df =c.head(10)
```

In [117]:
```
highest_deaths_df
```
Out[117]:

	location	deaths_per_million
162	San Marino	1237.550828
150	Peru	887.393445
18	Belgium	854.040375
3	Andorra	685.950948
177	Spain	624.406705
198	United Kingdom	611.525168
40	Chile	593.423100
97	Italy	587.097697
27	Brazil	582.331314