In [1]:

```
import pandas as pd
```

Load Local csv file

In [3]:

```
countries_df = pd.read_csv(r'C:\Users\Downloads\countries.csv')
covid_data_df = pd.read_csv(r'C:\Users\Downloads\covid-countries-data.csv')
```

Q1: Count the number of countries for which the `total_tests` data is missing.

Hint: Use the `.isna` method.

In [24]:

```
covid_data_df.total_tests.isna().sum()
```

Out[24]:

```
122
```

Q2: Merge `countries_df` with `covid_data_df` on the `location` column.

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

In [13]:

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

Q3: Add columns `tests_per_million` , `cases_per_million` and `deaths_per_million` into `combined_df` .

In [25]:

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

In [26]:

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

In [27]:

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

In [28]:

```
combined_df
```

Out[28]:

ocation	total_cases	total_deaths	total_tests	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita	tests_per_
anistan	38243.0	1409.0	NaN	Asia	38928341.0	64.83	0.50	1803.987	
Albania	9728.0	296.0	NaN	Europe	2877800.0	78.57	2.89	11803.431	
Algeria	45158.0	1525.0	NaN	Africa	43851043.0	76.88	1.90	13913.839	
Andorra	1199.0	53.0	NaN	Europe	77265.0	83.73	NaN	NaN	
Angola	2729.0	109.0	NaN	Africa	32866268.0	61.15	NaN	5819.495	
...	...	...	...	...	...	...	...	...	
Vietnam	1046.0	35.0	261004.0	Asia	97338583.0	75.40	2.60	6171.884	2681
Western Sahara	766.0	1.0	NaN	Africa	597330.0	70.26	NaN	NaN	
Yemen	1976.0	571.0	NaN	Asia	29825968.0	66.12	0.70	1479.147	
Zambia	12415.0	292.0	NaN	Africa	18383956.0	63.89	2.00	3689.251	
mbabwe	6638.0	206.0	97272.0	Africa	14862927.0	61.49	1.70	1899.775	6544

× 12 columns

Q4: Create a dataframe with 10 countires that have highest number of tests per million people.

In [50]:
```
highest_tests = combined_df.sort_values('tests_per_million', ascending=False).head(10)
highest_tests
```

Out[50]:

	location	total_cases	total_deaths	total_tests	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_cap
197	United Arab Emirates	71540.0	387.0	7177430.0	Asia	9890400.0	77.97	1.200	67293.4
14	Bahrain	52440.0	190.0	1118837.0	Asia	1701583.0	77.29	2.000	43290.7
115	Luxembourg	7928.0	124.0	385820.0	Europe	625976.0	82.25	4.510	94277.9
122	Malta	1931.0	13.0	188539.0	Europe	441539.0	82.53	4.485	36513.3
53	Denmark	17195.0	626.0	2447911.0	Europe	5792203.0	80.90	2.500	46682.5
96	Israel	122539.0	969.0	2353984.0	Asia	8655541.0	82.97	2.990	33132.3
89	Iceland	2121.0	10.0	88829.0	Europe	341250.0	82.99	2.910	46482.9
157	Russia	1005000.0	17414.0	37176827.0	Europe	145934460.0	72.58	8.050	24765.9
199	United States	6114406.0	185744.0	83898416.0	North America	331002647.0	78.86	2.770	54225.4
10	Australia	25923.0	663.0	6255797.0	Oceania	25499881.0	83.44	3.840	44648.7

Q5: Create a dataframe with 10 countires that have highest number of positive cases per million people.

In [51]:
```
highest_positive = combined_df.sort_values('cases_per_million', ascending=False).head(10)
highest_positive
```

Out[51]:

	location	total_cases	total_deaths	total_tests	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita
155	Qatar	119206.0	199.0	634745.0	Asia	2881060.0	80.23	1.20	116935.600
14	Bahrain	52440.0	190.0	1118837.0	Asia	1701583.0	77.29	2.00	43290.705
147	Panama	94084.0	2030.0	336345.0	North America	4314768.0	78.51	2.30	22267.037
40	Chile	414739.0	11344.0	2458762.0	South America	19116209.0	80.18	2.11	22767.037
162	San Marino	735.0	42.0	NaN	Europe	33938.0	84.97	3.80	56861.470
9	Aruba	2211.0	12.0	NaN	North America	106766.0	76.29	NaN	35973.781
105	Kuwait	86478.0	535.0	621616.0	Asia	4270563.0	75.49	2.00	65530.537
150	Peru	663437.0	29259.0	584232.0	South America	32971846.0	76.74	1.60	12236.706
27	Brazil	3997865.0	123780.0	4797948.0	South America	212559409.0	75.88	2.20	14103.452
199	United States	6114406.0	185744.0	83898416.0	North America	331002647.0	78.86	2.77	54225.446

Q6: Create a dataframe with 10 countires that have highest number of deaths cases per million people?

In [52]:

```
highest_deaths = combined_df.sort_values('deaths_per_million', ascending=False).head(10)
highest_deaths
```

Out[52]:

	location	total_cases	total_deaths	total_tests	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita
162	San Marino	735.0	42.0	NaN	Europe	33938.0	84.97	3.80	56861.470
150	Peru	663437.0	29259.0	584232.0	South America	32971846.0	76.74	1.60	12236.706
18	Belgium	85817.0	9898.0	2281853.0	Europe	11589616.0	81.63	5.64	42658.576
3	Andorra	1199.0	53.0	NaN	Europe	77265.0	83.73	NaN	NaN
177	Spain	479554.0	29194.0	6416533.0	Europe	46754783.0	83.56	2.97	34272.360
198	United Kingdom	338676.0	41514.0	13447568.0	Europe	67886004.0	81.32	2.54	39753.244
40	Chile	414739.0	11344.0	2458762.0	South America	19116209.0	80.18	2.11	22767.037
97	Italy	271515.0	35497.0	5214766.0	Europe	60461828.0	83.51	3.18	35220.084
27	Brazil	3997865.0	123780.0	4797948.0	South America	212559409.0	75.88	2.20	14103.452
182	Sweden	84532.0	5820.0	NaN	Europe	10099270.0	82.80	2.22	46949.283

Q7: Count number of countries that feature in both the lists of "highest number of tests per million" and "highest number of cases per million".

In [46]:

```
merged_df_cases_tests_highest = highest_tests.merge(highest_positive, on = 'location')
count = merged_df_cases_tests_highest.location.count()
print(count)
```

```
2
```

Q8: Count number of countries that feature in both the lists "20 countries with lowest GDP per capita" and "20 countries with

Q5. 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.

In [43]:

```
lowest_gdp = countries_df[countries_df['population']> 10**7].sort_values(by = ['gdp_per_capita']).head(20)
lowest_gdp
```

Out[43]:

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita
32	Burundi	Africa	11890781.0	61.58	0.8	702.225
52	Democratic Republic of Congo	Africa	89561404.0	60.68	NaN	808.133
140	Niger	Africa	24206636.0	62.42	0.3	926.000
118	Malawi	Africa	19129955.0	64.26	1.3	1095.042
132	Mozambique	Africa	31255435.0	60.85	0.7	1136.103
117	Madagascar	Africa	27691019.0	67.04	0.2	1416.440
207	Yemen	Asia	29825968.0	66.12	0.7	1479.147
176	South Sudan	Africa	11193729.0	57.85	NaN	1569.888
85	Haiti	North America	11402533.0	64.00	0.7	1653.173
195	Uganda	Africa	45741000.0	63.37	0.5	1697.707
31	Burkina Faso	Africa	20903278.0	61.58	0.4	1703.102
63	Ethiopia	Africa	114963583.0	66.60	0.3	1729.927
39	Chad	Africa	16425859.0	54.24	NaN	1768.153
0	Afghanistan	Asia	38928341.0	64.83	0.5	1803.987
158	Rwanda	Africa	12952209.0	69.02	NaN	1854.211
209	Zimbabwe	Africa	14862927.0	61.49	1.7	1899.775
82	Guinea	Africa	13132792.0	61.60	0.3	1998.926
121	Mali	Africa	20250834.0	59.31	0.1	2014.306
20	Benin	Africa	12123198.0	61.77	0.5	2064.236
135	Nepal	Asia	29136808.0	70.78	0.3	2442.804

In [44]:

```
lowest_beds= countries_df[countries_df['population']> 10**7].sort_values(by = ['hospital_beds_per_thousand']).he
lowest_beds
```

Out[44]:

	location	continent	population	life_expectancy	hospital_beds_per_thousand	gdp_per_capita
121	Mali	Africa	2.025083e+07	59.31	0.10	2014.306
117	Madagascar	Africa	2.769102e+07	67.04	0.20	1416.440
140	Niger	Africa	2.420664e+07	62.42	0.30	926.000
135	Nepal	Asia	2.913681e+07	70.78	0.30	2442.804
82	Guinea	Africa	1.313279e+07	61.60	0.30	1998.926
63	Ethiopia	Africa	1.149636e+08	66.60	0.30	1729.927
31	Burkina Faso	Africa	2.090328e+07	61.58	0.40	1703.102
195	Uganda	Africa	4.574100e+07	63.37	0.50	1697.707
0	Afghanistan	Asia	3.892834e+07	64.83	0.50	1803.987
20	Benin	Africa	1.212320e+07	61.77	0.50	2064.236
90	India	Asia	1.380004e+09	69.66	0.53	6426.674
80	Guatemala	North America	1.791557e+07	74.30	0.60	7423.808
145	Pakistan	Asia	2.208923e+08	67.27	0.60	5034.708
85	Haiti	North America	1.140253e+07	64.00	0.70	1653.173
207	Yemen	Asia	2.982597e+07	66.12	0.70	1479.147
187	Tanzania	Africa	5.973421e+07	65.46	0.70	2683.304
132	Mozambique	Africa	3.125544e+07	60.85	0.70	1136.103
32	Burundi	Africa	1.189078e+07	61.58	0.80	702.225
33	Cambodia	Asia	1.671897e+07	69.82	0.80	3645.070
204	Venezuela	South America	2.843594e+07	72.06	0.80	16745.022

In [53]:

```
merged_df =  lowest_gdp.merge(lowest_beds, on ='location')
count = merged_df.location.count()
print(count)
```

```
14
```