In [1]:
```python
import requests as r
```

In [2]:
```python
url = '''https://en.wikipedia.org/wiki/COVID-19_pandemic_by_country_and_territory'''

print(url)
```

https://en.wikipedia.org/wiki/COVID-19_pandemic_by_country_and_territory (https://en.wikipedia.org/wiki/COVID-19_pandemic_by_country_and_territory)

In [4]:
```python
response = r.get(url)
```

In [5]:
```python
response
```

Out[5]: <Response [200]>

In [7]:
```python
data = response.content
```

In [9]:
```python
data
```

Out[9]: b'<!DOCTYPE html>\n<html class="client-nojs vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-feature-sticky-header-disabled vector-feature-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled vector-feature-custom-font-size-clientpref-0 vector-feature-client-preferences-disabled vector-feature-client-prefs-pinned-disabled vector-feature-night-mode-disabled skin-night-mode-clientpref-0 vector-toc-available" lang="en" dir="ltr">\n<head>\n<meta charset="UTF-8">\n<title>COVID-19 pandemic by country and territory - Wikipedia</title>\n<script>(function(){var className="client-js vector-feature-language-in-header-enabled vector-feature-language-in-main-page-header-disabled vector-feature-sticky-header-disabled vector-feature-page-tools-pinned-disabled vector-feature-toc-pinned-clientpref-1 vector-feature-main-menu-pinned-disabled vector-feature-limited-width-clientpref-1 vector-feature-limited-width-content-enabled vector-feature-custom-font-size-clientpref-0 vector-feature-client-preferences-disabled vector-feature-client-prefs-pinned-disabled vector-feature-night-mode-disabled skin-night-mode-clientpref-0 vector-toc-available";var cookie=document.cookie.match(/(?:^|;)enwikimwclientpreferences=([^;]+)/);if(cookie){cookie[1].split(\'%2C\').forEach(function(pref){className=className.replace(new RegExp(\'(^|)\'+pref.replace(/-clientpref-\\w+$|[^\\w-]+/g,\'\')+\'-clientpref-\\\\w+(|$)\'),\'$1\'+pref+\'$2\');});}document.documentElement.className=className;}());RLCONF={"wgBreakFrames":false,"wgSeparatorTransformTable":["",""],\n"wgDigitTransformTable":["",""],"wgDefaultDateFormat":"dmy","wgMonthNames":["","January","February","March","April","May","June","July","August","September","October","November","December"],"wgRequestId":"3acf0700-3bdf-479b-8ff1-86c4abf6e76d","wgCanonicalNamespace":"","wgCanonicalSpecialPageName":false,"wgNamespaceNumber":0,"wgPageName":"COVID-19_pandemic_by_country_and_territory","wgTitle":"COVID-

In [10]:
```python
# pip install beautifulsoup4
```

In [11]:
```python
from bs4 import BeautifulSoup
```

In [12]:
```python
soup = BeautifulSoup(data,'html.parser')
```

In [13]:
```
1  soup
```

```
<ul class="vector-toc-list" id="toc-Total_vaccinations-sublist">
</ul>
</li>
</ul>
</li>
<li class="vector-toc-list-item vector-toc-level-1" id="toc-Statistics">
<a class="vector-toc-link" href="#Statistics">
<div class="vector-toc-text">
<span class="vector-toc-numb">2</span>Statistics</div>
</a>
<button aria-controls="toc-Statistics-sublist" class="cdx-button cdx-button--weight-quiet cdx-button--icon-only v
ector-toc-toggle">
<span class="vector-icon vector-icon--x-small mw-ui-icon-wikimedia-expand"></span>
<span>Toggle Statistics subsection</span>
</button>
<ul class="vector-toc-list" id="toc-Statistics-sublist">
<li class="vector-toc-list-item vector-toc-level-2" id="toc-Total_cases,_deaths,_and_death_rates_by_country">
<a class="vector-toc-link" href="#Total_cases,_deaths,_and_death_rates_by_country">
<div class="vector-toc-text">
<span class="vector-toc-numb">2.1</span>Total cases, deaths, and death rates by country</div>
```

In [23]:
```
1   all_row = []
2
3   for index,value in enumerate(soup.find_all('tr')):
4
5       if 46<=index<=284:
6           temp_data = value.text.strip().split('\n')
7
8           while '' in temp_data:
9               temp_data.remove('')
10  #         print(temp_data)
11           all_row.append(temp_data)
```

In [25]:
```
1   print(len(all_row))
```

239

In [26]:
```python
col = ['Country','Deaths/Million','Deaths','Cases']
```

In [27]:
```python
import pandas as pd
```

In [29]:
```python
table = pd.DataFrame(all_row,columns=col)
```

In [30]:
```python
table
```

Out[30]:

	Country	Deaths/Million	Deaths	Cases
0	World[a]	882	7,037,007	774,834,237
1	Peru	6,485	220,831	4,524,748
2	Bulgaria	5,704	38,691	1,328,885
3	Bosnia and Herzegovina	5,067	16,386	403,601
4	Hungary	4,920	49,046	2,230,202
...	...	...	...	...
234	Niue	—	—	993
235	Turkmenistan	—	0	0
236	Vatican City	—	0	26
237	Tokelau	—	0	80
238	Saint Helena, Ascension and Tristan da Cunha	—	—	2,166

239 rows × 4 columns

In [32]:
```python
table.to_csv(r"C:\Users\Lenovo\Desktop\covid_data.csv")
print('done')
```

done

In [ ]:   1