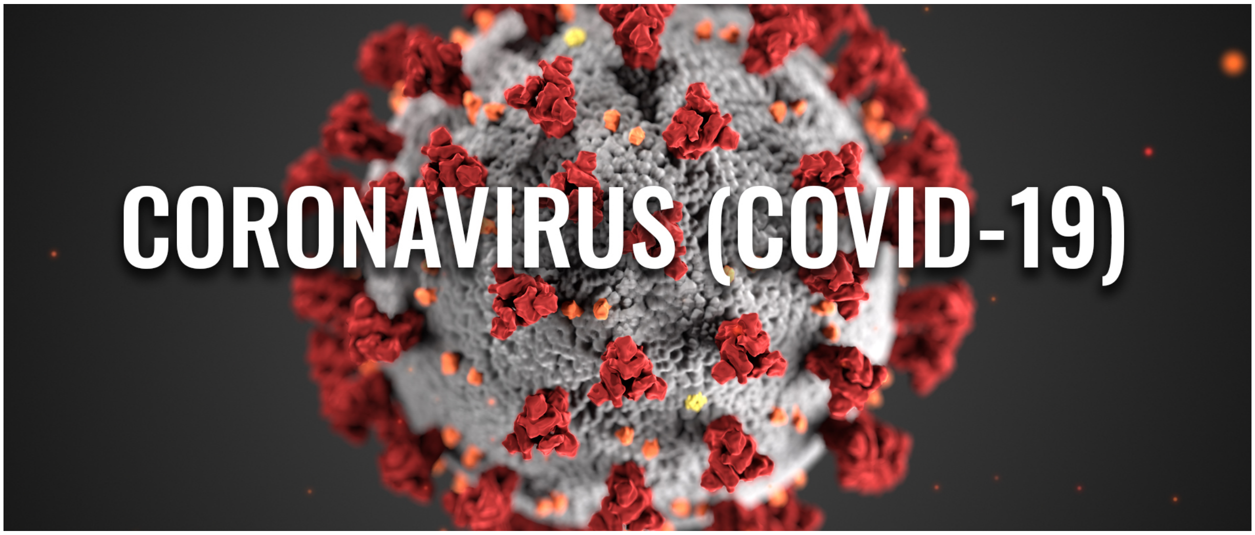
Report on Web Scraping COVID Data

Introduction

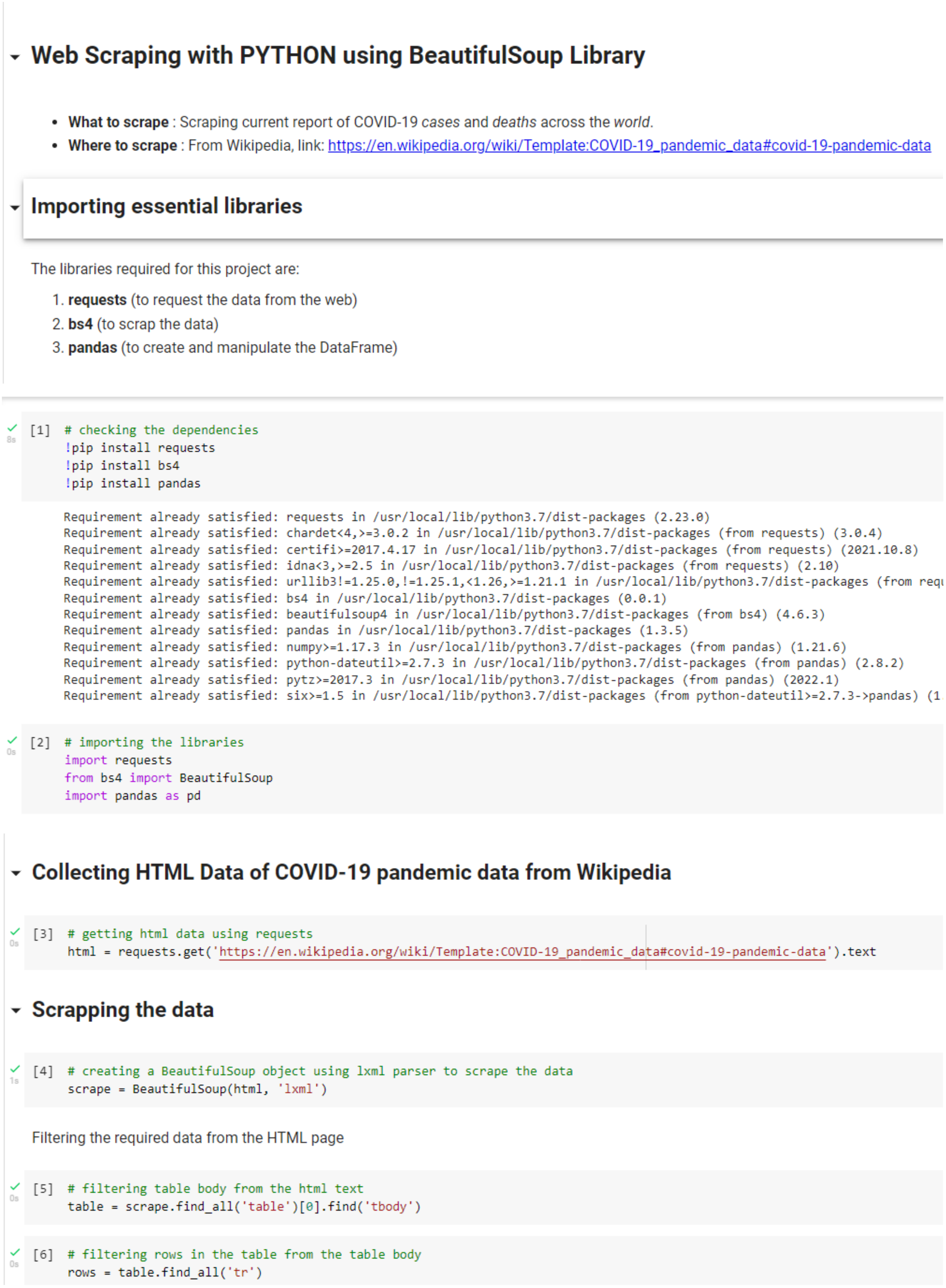
Coronavirus disease 2019 (**COVID-19**) is a contagious disease caused by a virus, the severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The first known case was identified in Wuhan, China, in December 2019. The disease spread worldwide, leading to the COVID-19 pandemic.



Work

I have scraped the data of COVID-19 pandemic cases and deaths that happened across the world using the BeautifulSoup web scraping library. And then converted that data into a Pandas Dataframe. Also, I have done Data Preprocessing on the data I gathered.

Code



Web Scraping with PYTHON using BeautifulSoup Library

- **What to scrape** : Scraping current report of COVID-19 *cases* and *deaths* across the *world*.
- **Where to scrape** : From Wikipedia, link: https://en.wikipedia.org/wiki/Template:COVID-19_pandemic_data#covid-19-pandemic-data

Importing essential libraries

The libraries required for this project are:

1. **requests** (to request the data from the web)
2. **bs4** (to scrap the data)
3. **pandas** (to create and manipulate the DataFrame)

```
[1]  # checking the dependencies
     !pip install requests
     !pip install bs4
     !pip install pandas
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (2.23.0)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests) (3.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests) (2021.10.8)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requ
Requirement already satisfied: bs4 in /usr/local/lib/python3.7/dist-packages (0.0.1)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.7/dist-packages (from bs4) (4.6.3)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (1.3.5)
Requirement already satisfied: numpy>=1.17.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (1.21.6)
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-packages (from pandas) (2022.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/dist-packages (from python-dateutil>=2.7.3->pandas) (1
```

```
[2]  # importing the libraries
     import requests
     from bs4 import BeautifulSoup
     import pandas as pd
```

Collecting HTML Data of COVID-19 pandemic data from Wikipedia

```
[3]  # getting html data using requests
     html = requests.get('https://en.wikipedia.org/wiki/Template:COVID-19_pandemic_data#covid-19-pandemic-data').text
```

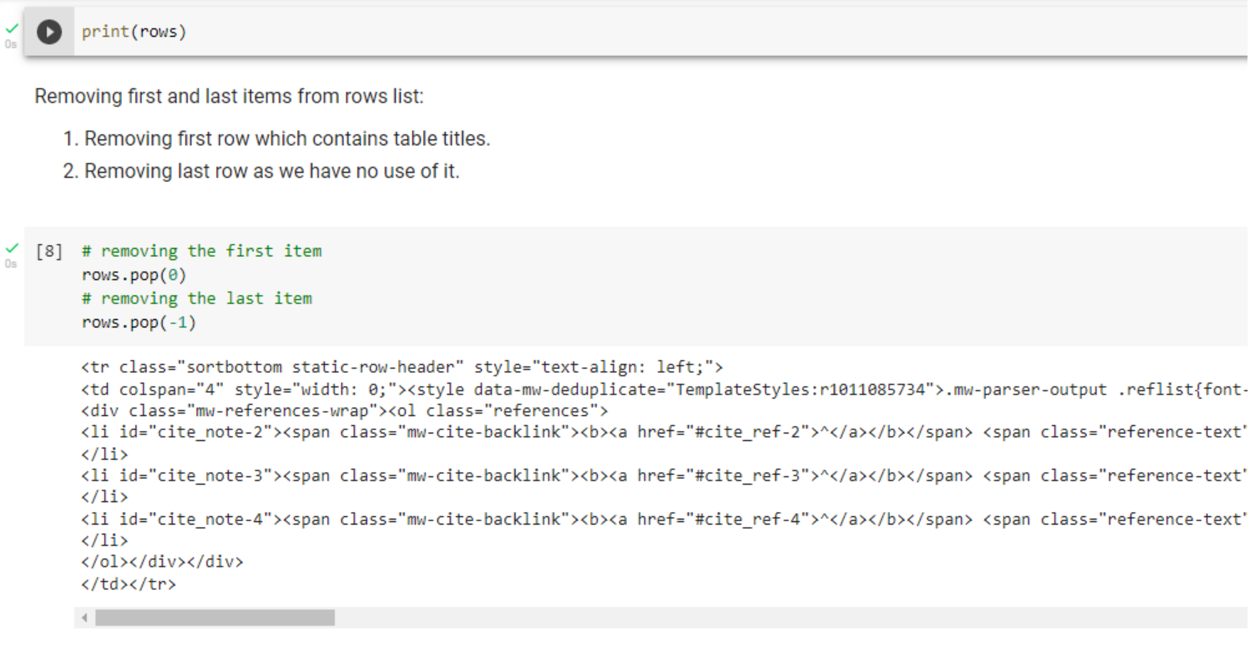
Scrapping the data

```
[4]  # creating a BeautifulSoup object using lxml parser to scrape the data
     scrape = BeautifulSoup(html, 'lxml')
```

Filtering the required data from the HTML page

```
[5]  # filtering table body from the html text
     table = scrape.find_all('table')[0].find('tbody')
```

```
[6]  # filtering rows in the table from the table body
     rows = table.find_all('tr')
```



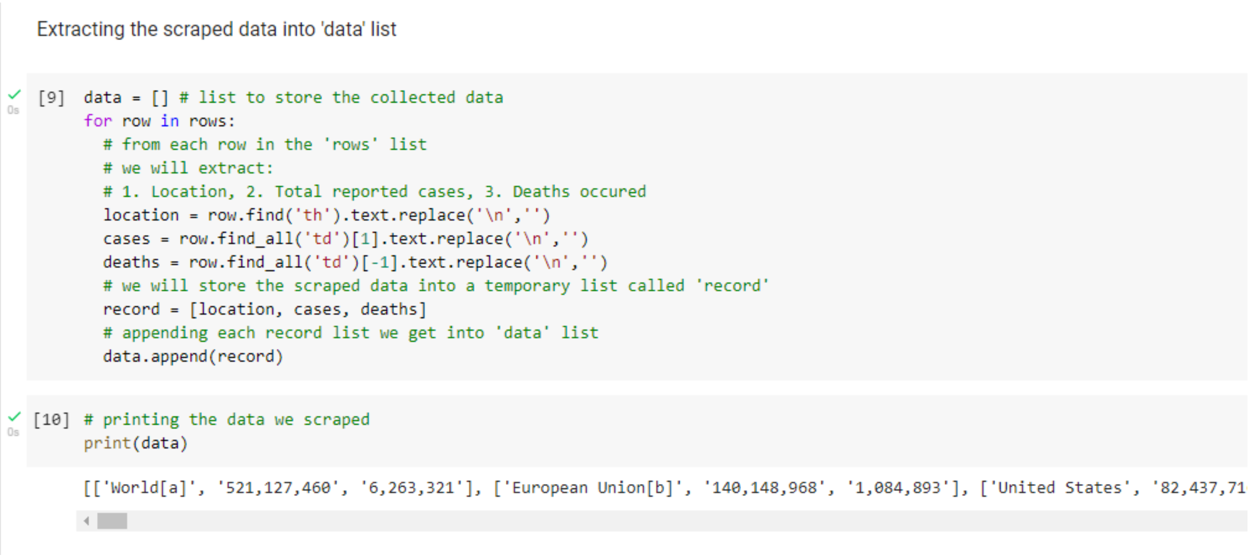
```
print(rows)
```

Removing first and last items from rows list:

1. Removing first row which contains table titles.
2. Removing last row as we have no use of it.

```
[8] # removing the first item
    rows.pop(0)
    # removing the last item
    rows.pop(-1)
```

```
<tr class="sortbottom static-row-header" style="text-align: left;">
<td colspan="4" style="width: 0;"><style data-mw-deduplicate="TemplateStyles:r1011085734">.mw-parser-output .reflist{font-
<div class="mw-references-wrap"><ol class="references">
<li id="cite_note-2"><span class="mw-cite-backlink"><b><a href="#cite_ref-2">^</a></b></span> <span class="reference-text'
</li>
<li id="cite_note-3"><span class="mw-cite-backlink"><b><a href="#cite_ref-3">^</a></b></span> <span class="reference-text'
</li>
<li id="cite_note-4"><span class="mw-cite-backlink"><b><a href="#cite_ref-4">^</a></b></span> <span class="reference-text'
</li>
</ol></div></div>
</td></tr>
```



Extracting the scraped data into 'data' list

```
[9] data = [] # list to store the collected data
    for row in rows:
      # from each row in the 'rows' list
      # we will extract:
      # 1. Location, 2. Total reported cases, 3. Deaths occured
      location = row.find('th').text.replace('\n','')
      cases = row.find_all('td')[1].text.replace('\n','')
      deaths = row.find_all('td')[-1].text.replace('\n','')
      # we will store the scraped data into a temporary list called 'record'
      record = [location, cases, deaths]
      # appending each record list we get into 'data' list
      data.append(record)
```

```
[10] # printing the data we scraped
     print(data)
```

```
[['World[a]', '521,127,460', '6,263,321'], ['European Union[b]', '140,148,968', '1,084,893'], ['United States', '82,437,71
```

Creating DataFrame

```
[11] # creating a DataFrame named as 'covid_data' using the 'data' list (of lists) we scraped
     covid_data = pd.DataFrame(data, columns = ['Location', 'Cases', 'Deaths'])
```

```
[12] # first five rows of the DataFrame
     covid_data.head()
```

	Location	Cases	Deaths
0	World[a]	521,127,460	6,263,321
1	European Union[b]	140,148,968	1,084,893
2	United States	82,437,716	999,570
3	India	43,121,599	524,214
4	Brazil	30,682,094	665,104

```
[13] # last five rows of the DataFrame
     covid_data.tail()
```

	Location	Cases	Deaths
212	Macau	82	—
213	Vatican City	29	0
214	Marshall Islands	17	—
215	Federated States of Micronesia	7	0
216	Saint Helena, Ascension and Tristan da Cunha	4	—

Data Preprocessing

```
[14] # shape of the DataFrame
     covid_data.shape
```

```
(217, 3)
```

```
[15] # Info of the DataFrame
     covid_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 217 entries, 0 to 216
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   Location  217 non-null    object
 1   Cases     217 non-null    object
 2   Deaths    217 non-null    object
dtypes: object(3)
memory usage: 5.2+ KB
```

```
[16] # Data types of the columns in the DataFrame
     covid_data.dtypes
```

```
Location    object
Cases       object
Deaths      object
dtype: object
```

```
[17] # checking if there is any null value
     covid_data.isnull().sum()
```

```
     Location    0
     Cases       0
     Deaths      0
     dtype: int64
```

In the DataFrame we have created:

1. There are 217 rows and 3 columns.
2. There are no null values.

However the data type of the columns 'Cases' and 'Deaths' is inappropriate and also the values are not in the right format.

Changing the values into right format for both 'Cases' and 'Deaths' columns.

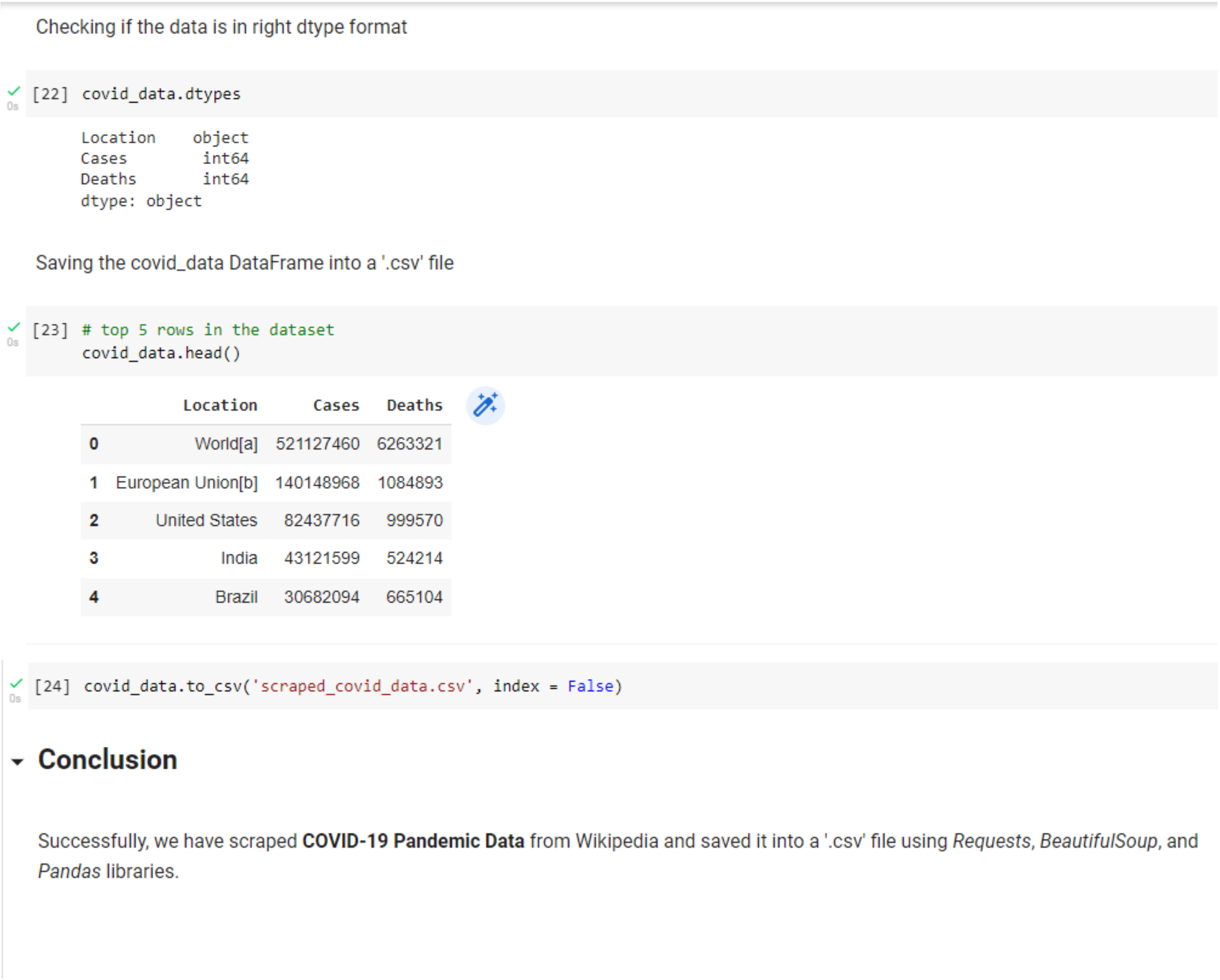
```
[18] # creating a function to change the format of the values.
     def valToNum(val):
       # Our objective is to
       # 1. Remove the commas and
       # 2. Replace the value to 0 if '—' is the value.
       val = val.replace(',','')
       val = val.replace('—','0')
       return val
```

Applying this function to every value in '*Cases*' and '*Deaths*' column

```
[19] # chaning the data format of 'Cases' using apply() function in pandas
     covid_data['Cases'] = covid_data['Cases'].apply(valToNum)
```

```
[20] # chaning the data format of 'Deaths' using apply() function in pandas
     covid_data['Deaths'] = covid_data['Deaths'].apply(valToNum)
```

```
[21] # changing the dtype of both the columns to pandas int64 type
     covid_data['Cases'] = covid_data['Cases'].astype('int64')
     covid_data['Deaths'] = covid_data['Deaths'].astype('int64')
```



Checking if the data is in right dtype format

```
[22] covid_data.dtypes
```

```
Location    object
Cases        int64
Deaths       int64
dtype: object
```

Saving the covid_data DataFrame into a '.csv' file

```
[23] # top 5 rows in the dataset
     covid_data.head()
```

	Location	Cases	Deaths
0	World[a]	521127460	6263321
1	European Union[b]	140148968	1084893
2	United States	82437716	999570
3	India	43121599	524214
4	Brazil	30682094	665104

```
[24] covid_data.to_csv('scraped_covid_data.csv', index = False)
```

Conclusion

Successfully, we have scraped **COVID-19 Pandemic Data** from Wikipedia and saved it into a '.csv' file using *Requests*, *BeautifulSoup*, and *Pandas* libraries.

Data Source

- ➔ https://en.wikipedia.org/wiki/Template:COVID-19_pandemic_data#covid-19-pandemic-data

References

- ➔ https://www.youtube.com/watch?v=XVv6mJpFOb0