Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Webscrapping and Data Visualization ¶

Name: Ooi Caaron

IC Number: 990701-07-5837

Date : 5/7/23

Introduction : Learning webscrapping and data visualization, displaying data through different type of graph

Conclusion : Still need to practice more and do revision

Mini Project 2

Webscraping and Data Visualization

Dataset: https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/ (https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/)

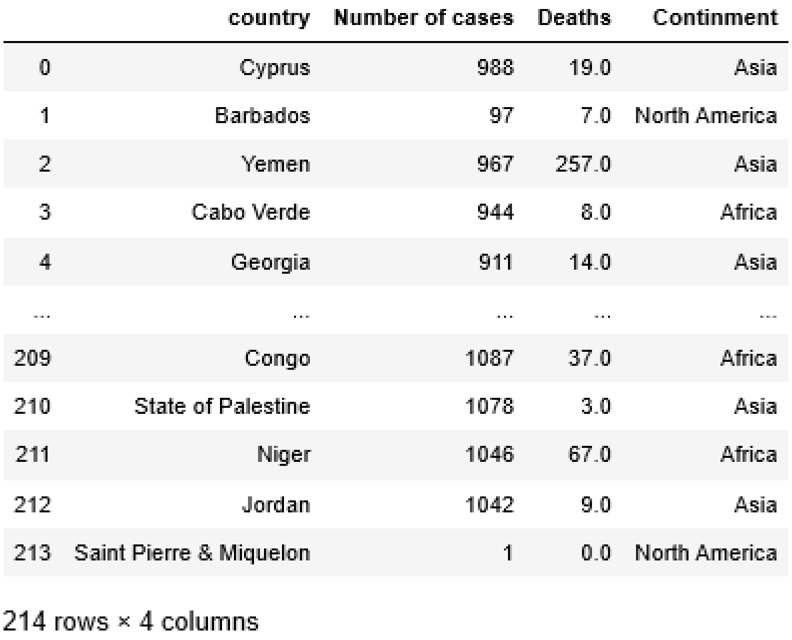
In this project, you are encouraged to use Worldometers to extract the number of COVID cases and then you will do data analysis and create some visualizations.

1. Import required libraries and write code to do webscraping

In [3]:

```
import pandas as pd
from selenium import webdriver
from bs4 import BeautifulSoup
import time
import texttable as tt
driver = webdriver.Chrome('C://GoogleDriver//chromedriver')
url='https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/'
driver.get(url)
soup = BeautifulSoup(driver.page_source, 'html.parser')
```

2. After running above code you are able to extract the data from the website, now we will be creating a pandas data frame for further analysis.



	country	Number of cases	Deaths	Continment
0	Cyprus	988	19.0	Asia
1	Barbados	97	7.0	North America
2	Yemen	967	257.0	Asia
3	Cabo Verde	944	8.0	Africa
4	Georgia	911	14.0	Asia
...	...	...	...	...
209	Congo	1087	37.0	Africa
210	State of Palestine	1078	3.0	Asia
211	Niger	1046	67.0	Africa
212	Jordan	1042	9.0	Asia
213	Saint Pierre & Miquelon	1	0.0	North America

214 rows × 4 columns

In [4]:

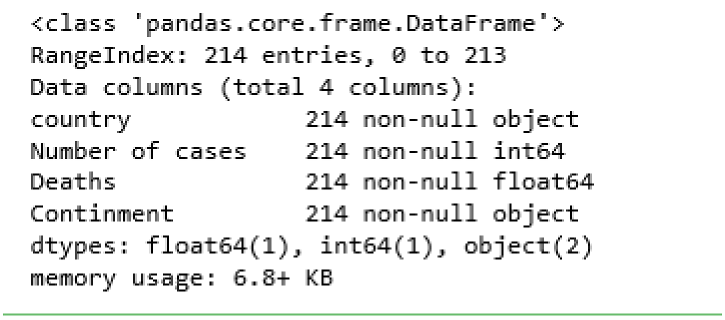
```
df_data = []
for tr in soup.find_all('tr', attrs={'role': 'row'}):
    row_data = [td.text.rstrip() for td in tr.find_all('td')]
    if len(row_data) == 4:
        country, cases, deaths, region = row_data
        df_data.append([country, cases, deaths, region])

df = pd.DataFrame(df_data, columns=['Country', 'Cases', 'Deaths', 'Region'])
df = df[df.Region != '']
df
```

Out[4]:

	Country	Cases	Deaths	Region
0	United States	107,346,013	1,168,414	North America
1	India	44,994,407	531,910	Asia
2	France	40,138,560	167,642	Europe
3	Germany	38,428,685	174,352	Europe
4	Brazil	37,682,660	704,159	South America
...	...	...	...	...
224	Montserrat	1,403	8	North America
225	Niue	820	0	Australia/Oceania
226	Holy See	29	0	Europe
227	Tokelau	23	0	Australia/Oceania
228	Western Sahara	10	1	Africa

229 rows × 4 columns

3. Data Type



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 214 entries, 0 to 213
Data columns (total 4 columns):
country              214 non-null object
Number of cases      214 non-null int64
Deaths               214 non-null float64
Continment           214 non-null object
dtypes: float64(1), int64(1), object(2)
memory usage: 6.8+ KB
```

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 229 entries, 0 to 228
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Country  229 non-null    object
 1   Cases    229 non-null    object
 2   Deaths   229 non-null    object
 3   Region   229 non-null    object
dtypes: object(4)
memory usage: 8.9+ KB
```

4. Creating a new column Death_rate

Hint: Death_rate = 100*(Death/Number of cases)

In [6]:

```
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

df['Deaths'] = df['Deaths'].str.replace(',', '')
df['Cases'] = df['Cases'].str.replace(',', '')
df['Deaths'] = pd.to_numeric(df['Deaths'])
df['Cases'] = pd.to_numeric(df['Cases'])

df['Death_rate'] = df['Deaths'] / df['Cases']
df['Death_rate'] = df['Death_rate'] * 100
df
```

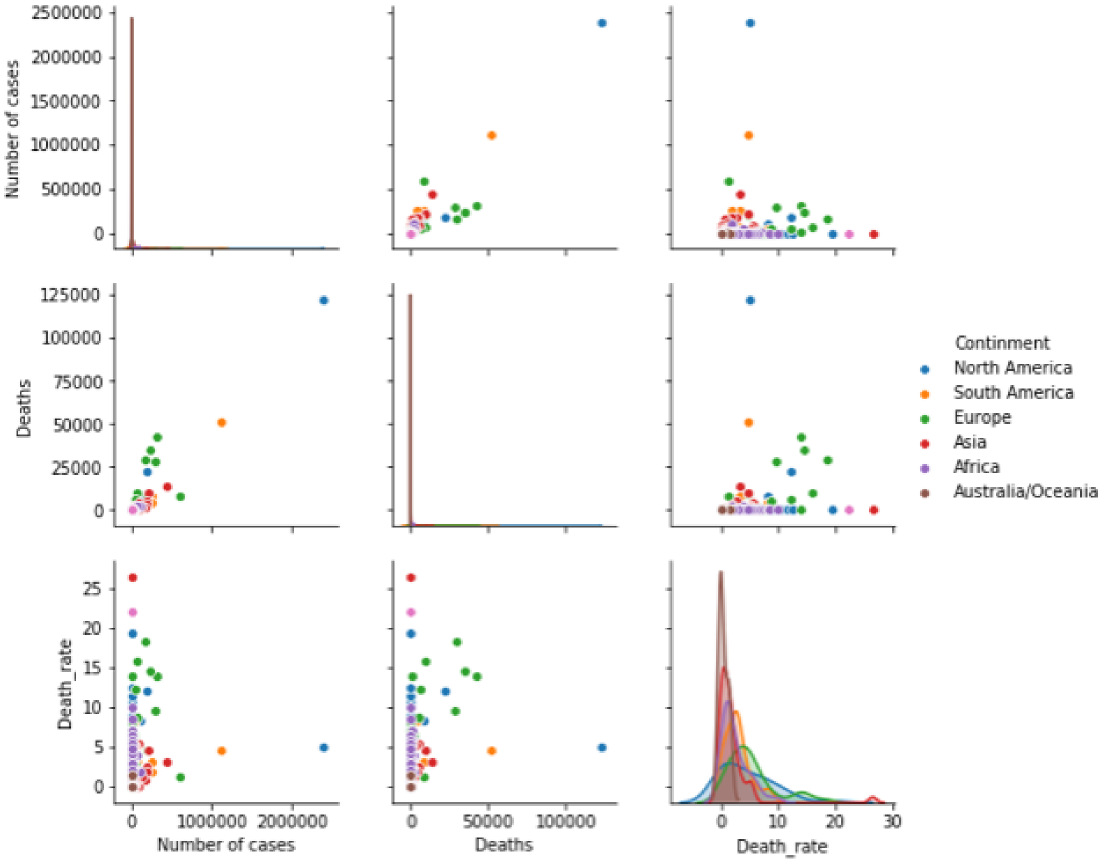
Out[6]:

	Country	Cases	Deaths	Region	Death_rate
0	United States	107346013	1168414	North America	1.088456
1	India	44994407	531910	Asia	1.182169
2	France	40138560	167642	Europe	0.417658
3	Germany	38428685	174352	Europe	0.453703
4	Brazil	37682660	704159	South America	1.868655
...	...	...	...	...	...
224	Montserrat	1403	8	North America	0.570207
225	Niue	820	0	Australia/Oceania	0.000000
226	Holy See	29	0	Europe	0.000000
227	Tokelau	23	0	Australia/Oceania	0.000000
228	Western Sahara	10	1	Africa	10.000000

229 rows × 5 columns

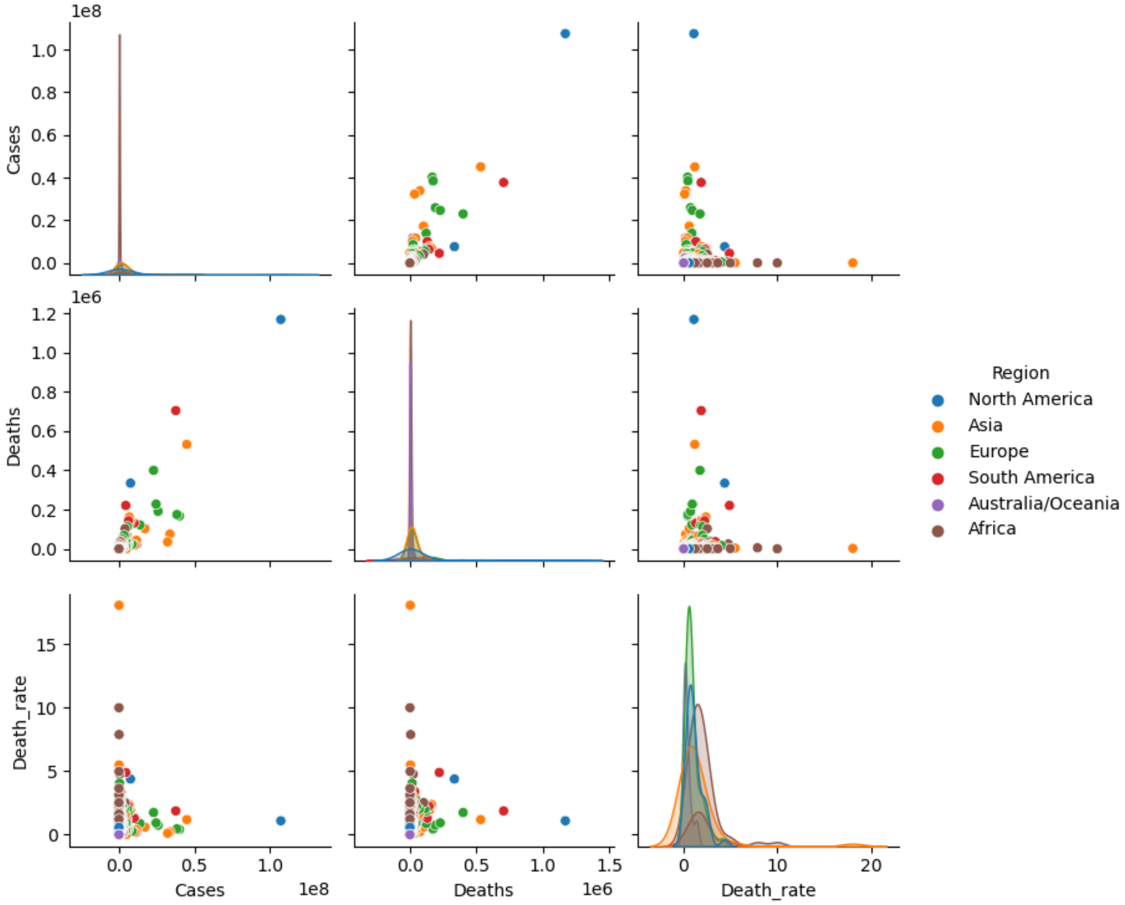
5. Data Visualization - Pairplot

```
<seaborn.axisgrid.PairGrid at 0x2[illegible]>

<Figure size 1600x480 with 0 Axes>
```



In [14]:

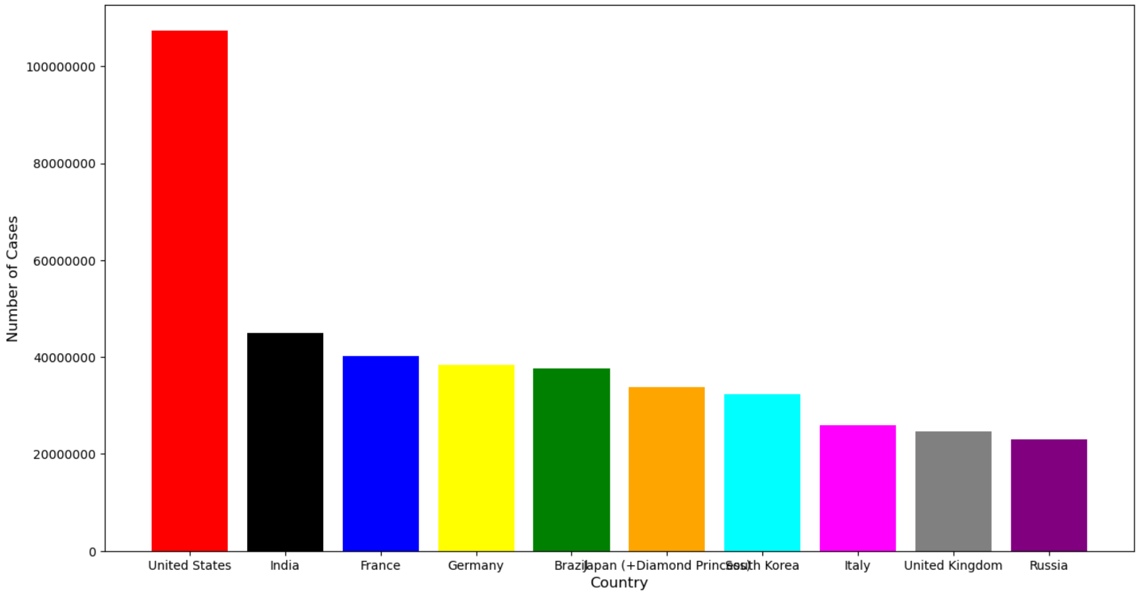
```
sns.pairplot(df, hue = 'Region')
plt.show()
```



6. Data Visualization - barplot

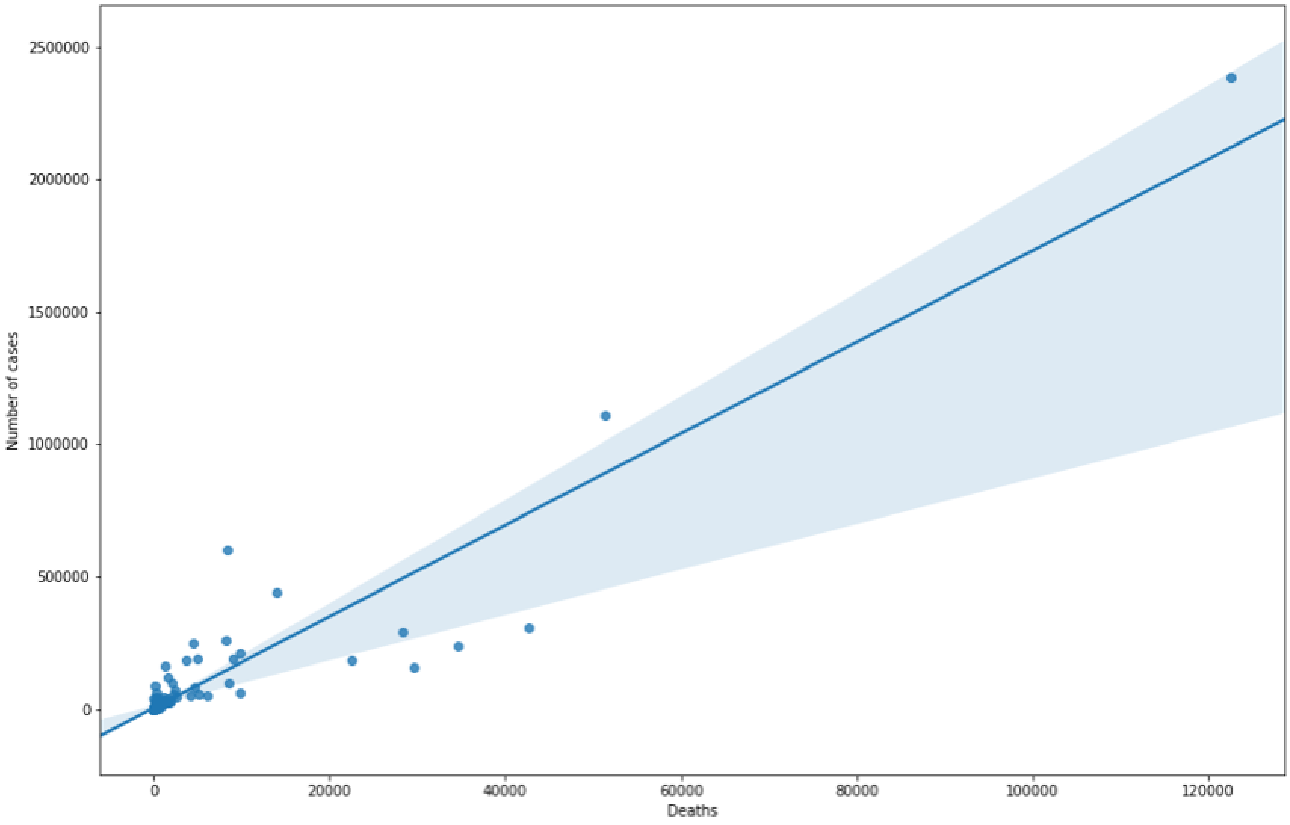
In [15]:

```
plt.figure(figsize = (15,8))
plt.bar(df['Country'].head(10),
        df['Cases'].head(10),
        color=['red','black','blue','yellow','green','orange','cyan','magenta','grey','p
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.xlabel('Country', fontsize = 12)
plt.ylabel('Number of Cases', fontsize = 12)
plt.show()
```



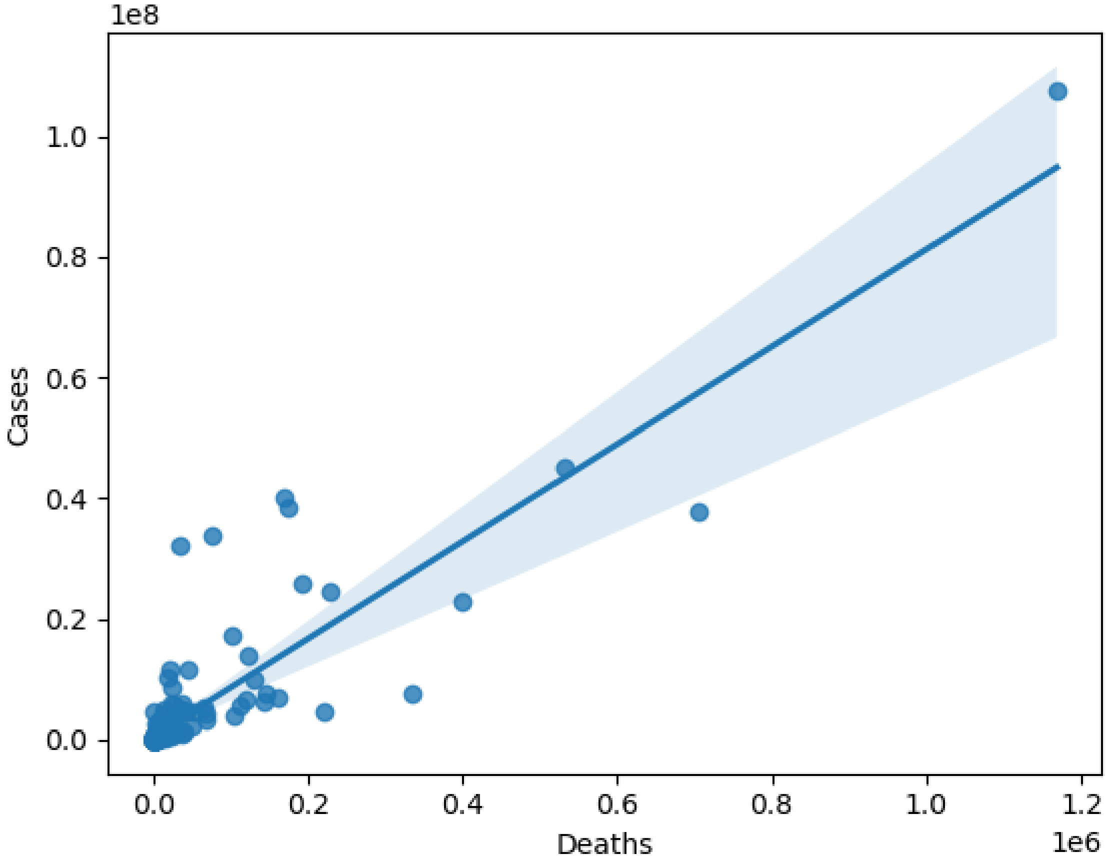
7. Data Visualization - regplot

```
<matplotlib.axes._subplots.AxesSubplot at 0x247da3f5bc8>
```



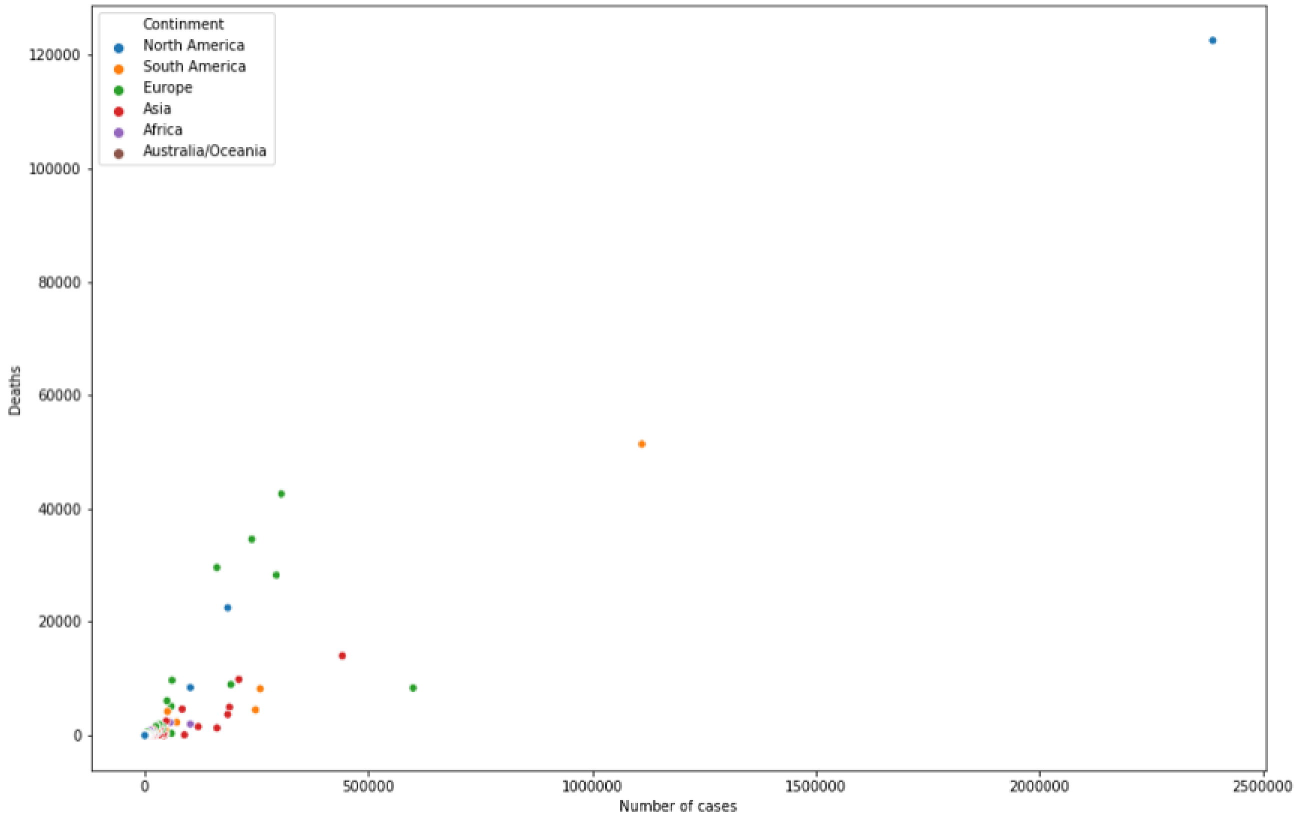
In [18]:

```
sns.regplot(x = df['Deaths'], y = df['Cases'], data = df)
plt.show()
```



8. Data Visualization - scatterplot

```
<matplotlib.axes._subplots.AxesSubplot at 0x247da544748>
```

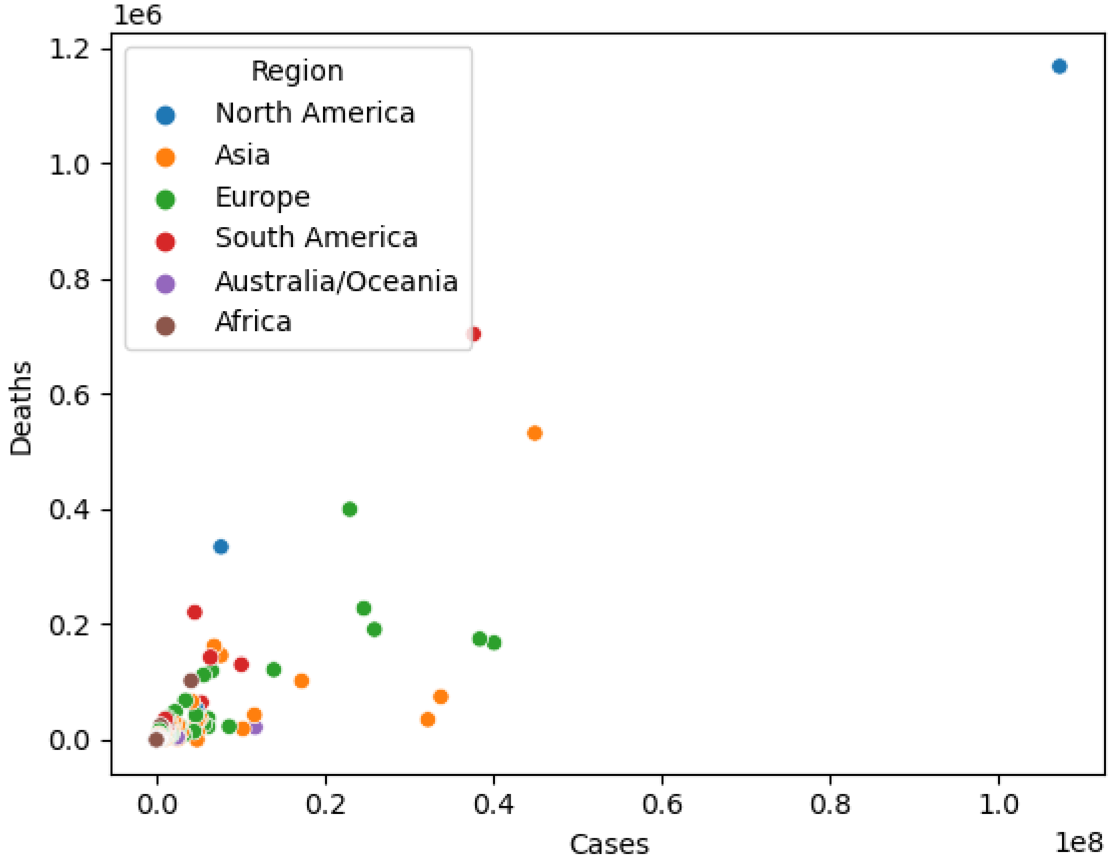


In [20]:

```
sns.scatterplot(x = df['Cases'], y = df['Deaths'], data = df , hue = 'Region')
```

Out[20]:

```
<Axes: xlabel='Cases', ylabel='Deaths'>
```



9. Data Visualization - boxplot

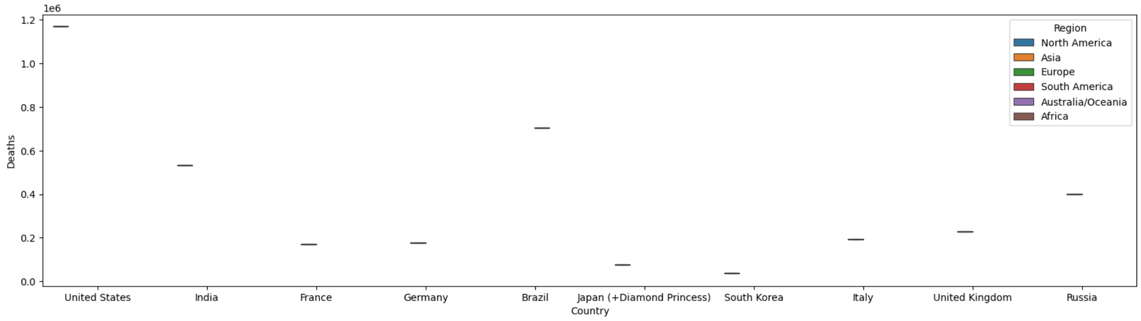
```
matplotlib.axes._subplots.AxesSubplot at 0x247da618a88>
```

In [23]:

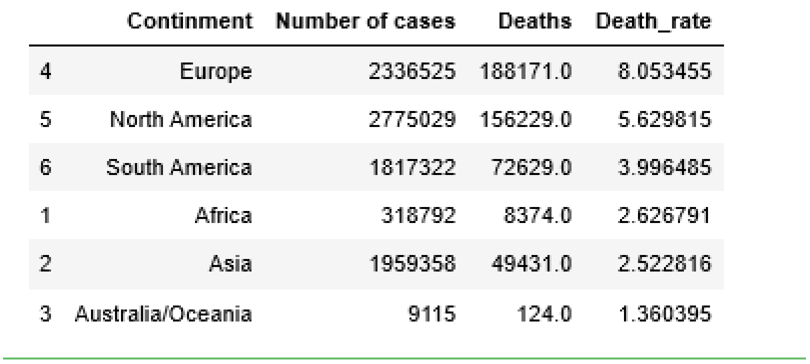
```
plt.figure(figsize = (20, 5))
sns.boxplot(x = df['Country'].head(10),
    y = df['Deaths'].head(10), data = df, hue = 'Region')
```

Out[23]:

```
<Axes: xlabel='Country', ylabel='Deaths'>
```



10. Write code to show the table as below



	Continment	Number of cases	Deaths	Death_rate
4	Europe	2336525	188171.0	8.053455
5	North America	2775029	156229.0	5.629815
6	South America	1817322	72629.0	3.996485
1	Africa	318792	8374.0	2.626791
2	Asia	1959358	49431.0	2.522816
3	Australia/Oceania	9115	124.0	1.360395

In [30]:

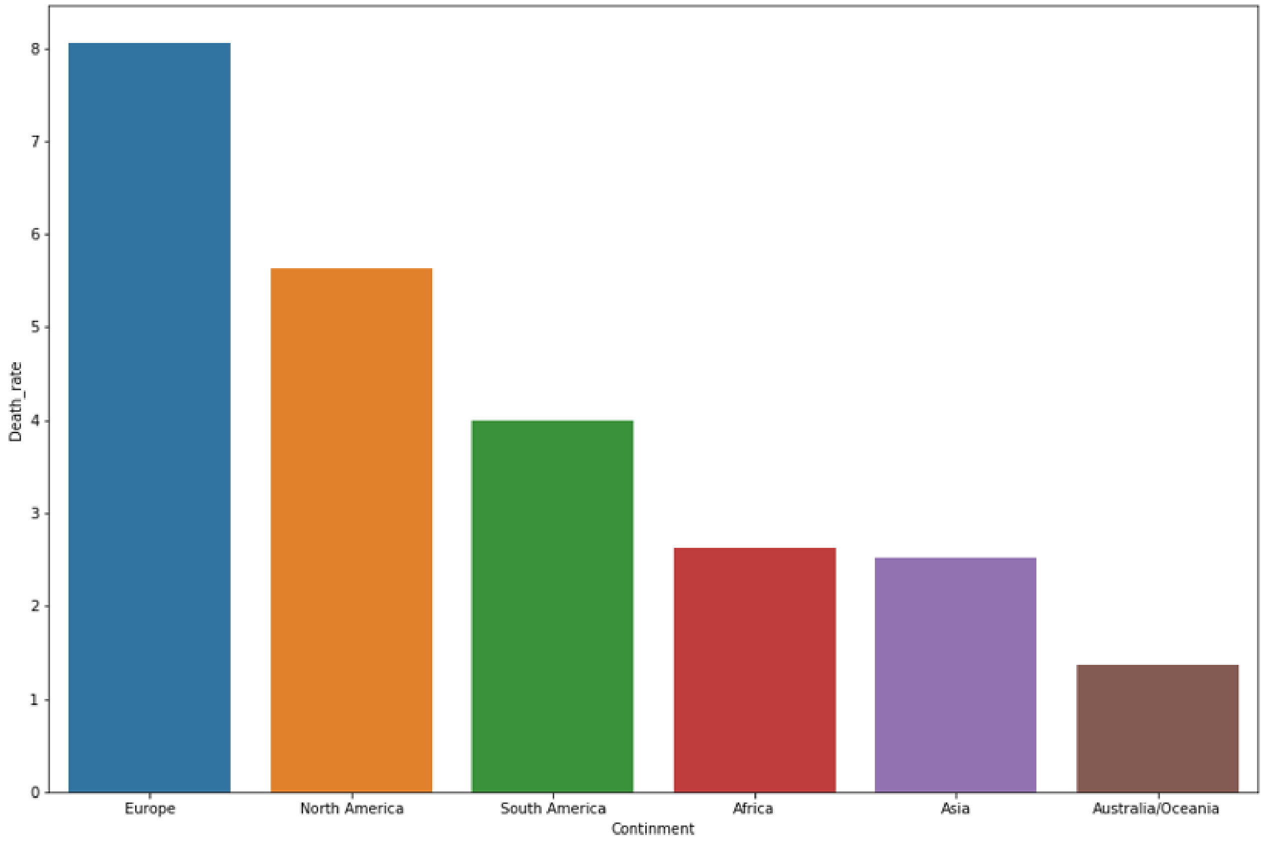
```
region = df.groupby('Region')[['Cases','Deaths','Death_rate']].sum().reset_index()
region = region.sort_values('Death_rate', ascending=False)
region
```

Out[30]:

	Region	Cases	Deaths	Death_rate
0	Africa	12831369	258806	110.763916
1	Asia	218285604	1547803	68.704256
3	Europe	249685794	2067060	43.892636
4	North America	127017548	1637506	41.869273
5	South America	68833115	1357694	24.933219
2	Australia/Oceania	14538582	29206	6.586907

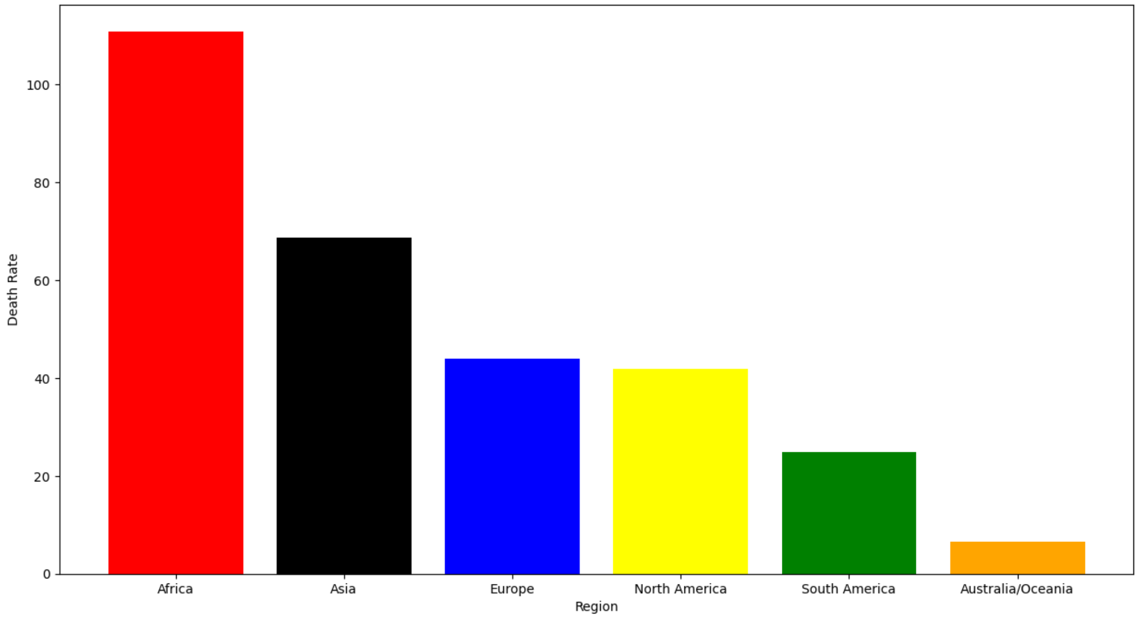
11. Data Visualization - barplot with death rate

```
<matplotlib.axes._subplots.AxesSubplot at 0x247da7bdb48>
```



In [32]:

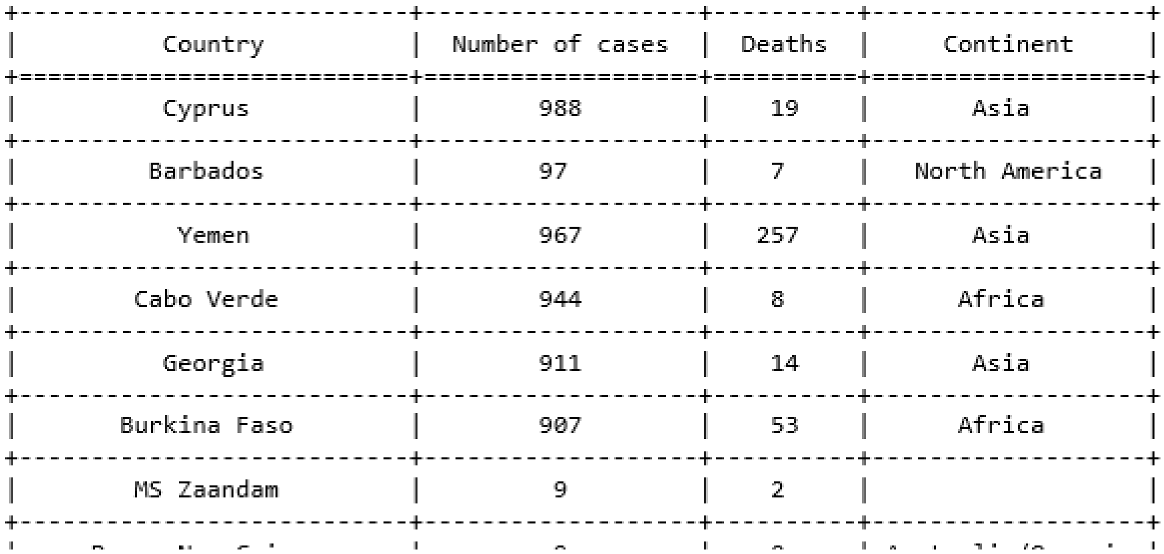
```
plt.figure(figsize = (15,8))
plt.bar(region['Region'], region['Death_rate'], color = ['red', 'black', 'blue', 'yellow
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.xlabel('Region')
plt.ylabel('Death Rate')
plt.show()
```



12. Create texttable

Hint: import texttable as tt

table = tt.Texttable() table.add_rows([(None, None, None, None)] + data) # Add an empty row at the beginning for the headers



```
+---------------------------+-------------------+----------+-------------------+
|          Country          |  Number of cases  |  Deaths  |     Continent     |
+===========================+===================+==========+===================+
|          Cyprus           |        988        |    19    |       Asia        |
+---------------------------+-------------------+----------+-------------------+
|         Barbados          |        97         |    7     |   North America   |
+---------------------------+-------------------+----------+-------------------+
|           Yemen           |        967        |   257    |       Asia        |
+---------------------------+-------------------+----------+-------------------+
|        Cabo Verde         |        944        |    8     |      Africa       |
+---------------------------+-------------------+----------+-------------------+
|          Georgia          |        911        |    14    |       Asia        |
+---------------------------+-------------------+----------+-------------------+
|       Burkina Faso        |        907        |    53    |      Africa       |
+---------------------------+-------------------+----------+-------------------+
|        MS Zaandam         |         9         |    2     |                   |
+---------------------------+-------------------+----------+-------------------+
|     Papua New Guinea      |         8         |    0     | Australia/Oceania |
```

In [8]:

```
df = df.head(10)
table = tt.Texttable()
table.set_cols_align(['a', 'a', 'a', 'a'])
table.set_cols_valign(['b', 'b', 'b', 'b'])
cases = df['Cases']
deaths = df['Deaths']
region = df['Region']
country = df['Country']
rows = [['Country', 'Cases', 'Deaths', 'Region']]

for x in range(10):
    rows.append([country[x], cases[x], deaths[x], region[x]])

table.add_rows(rows)

print(table.draw())
```

```
+---------------------------+-----------+---------+---------------+
|          Country          |   Cases   | Deaths  |    Region     |
+===========================+===========+=========+===============+
| United States             | 1.073e+08 | 1168414 | North America |
+---------------------------+-----------+---------+---------------+
| India                     | 44994407  | 531910  | Asia          |
+---------------------------+-----------+---------+---------------+
| France                    | 40138560  | 167642  | Europe        |
+---------------------------+-----------+---------+---------------+
| Germany                   | 38428685  | 174352  | Europe        |
+---------------------------+-----------+---------+---------------+
| Brazil                    | 37682660  | 704159  | South America |
+---------------------------+-----------+---------+---------------+
| Japan (+Diamond Princess) | 33804284  | 74707   | Asia          |
+---------------------------+-----------+---------+---------------+
| South Korea               | 32256154  | 35071   | Asia          |
+---------------------------+-----------+---------+---------------+
| Italy                     | 25897801  | 190868  | Europe        |
+---------------------------+-----------+---------+---------------+
| United Kingdom            | 24636637  | 227524  | Europe        |
+---------------------------+-----------+---------+---------------+
| Russia                    | 22963688  | 399649  | Europe        |
+---------------------------+-----------+---------+---------------+
```

In [ ]: