Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Webscrapping and Data Visualization

Name: Phua Yan Han

IC Number: 050824070059

Date : 5/7/23

Introduction : learning webscraping

Conclusion : learn how to do webscraping

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 [2]:
```
import time
from selenium import webdriver
from bs4 import BeautifulSoup
import time
import pandas as pd
driver = webdriver.Chrome('C:\\Users\Asus\Documents\ChromeDriver\chromedriver\
```

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.

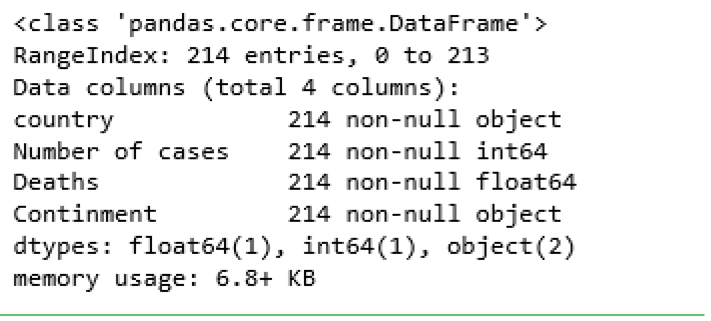
	country	Number of cases	Deaths	Continment
0	Cyprus	988	19.0	Asia
1	Barbados	97	7.0	North America
2	Yemen	967	257.0	Asia
3	Cabo Verde	944	8.0	Africa
4	Georgia	911	14.0	Asia
...	...	...	...	...
209	Congo	1087	37.0	Africa
210	State of Palestine	1078	3.0	Asia
211	Niger	1046	67.0	Africa
212	Jordan	1042	9.0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

214 rows × 4 columns

In [3]:
```python
driver.get("https://www.worldometers.info/coronavirus/countries-where-coronavir
data = []
soup = BeautifulSoup(driver.page_source, "html.parser")
for tr in soup.find_all('tr', attrs={'role': 'row'}):
        for td in tr.find_all('td'):
            data.append(td.text.rstrip())
data

num_columns = 4
num_rows = len(data) // num_columns

data_2d = [data[i*num_columns : (i+1)*num_columns] for i in range(num_rows)]

df = pd.DataFrame(data_2d, columns=['Country', 'Cases', 'Deaths', 'Region'])
print(df)
```

```
             Country        Cases     Deaths             Region
0      United States  107,331,578  1,168,278      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
..               ...          ...        ...                ...
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       MS Zaandam            9          2

[230 rows x 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 [3]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 230 entries, 0 to 229
Data columns (total 4 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Country  230 non-null    object
 1   Cases    230 non-null    object
 2   Deaths   230 non-null    object
 3   Region   230 non-null    object
dtypes: object(4)
memory usage: 7.3+ KB
```

4. Creating a new column Death_rate

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

In [4]:
```
calculation = []
death=[]
cases=[]
for index, row in df.iterrows():
    # Access row data using row[column_name]
    death.append((int(row['Deaths'].replace(",", ""))))
    cases.append(int(row['Cases'].replace(",", "")))
    calculation.append((int(row['Deaths'].replace(",", ""))/int(row['Cases'].re
df["DeathRate"]=calculation
df["Cases"]=cases
df["Deaths"]=death
df = df[df['Region'] != '']
df
```

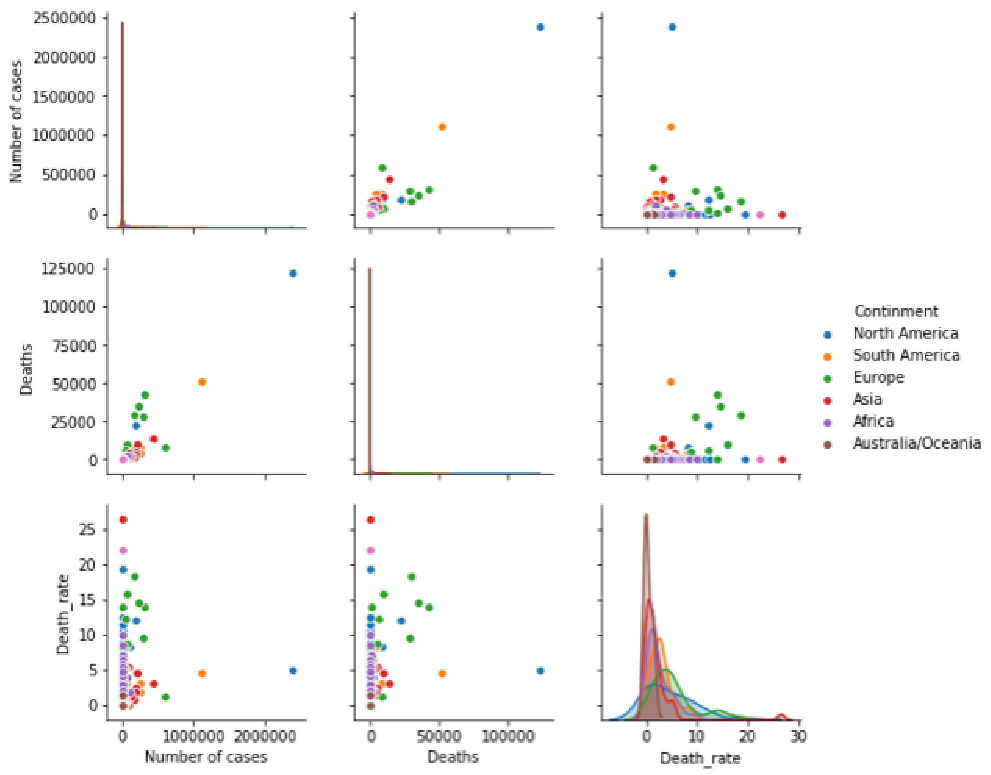
Out[4]:

	Country	Cases	Deaths	Region	DeathRate
0	United States	107331578	1168278	North America	1.088476
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

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

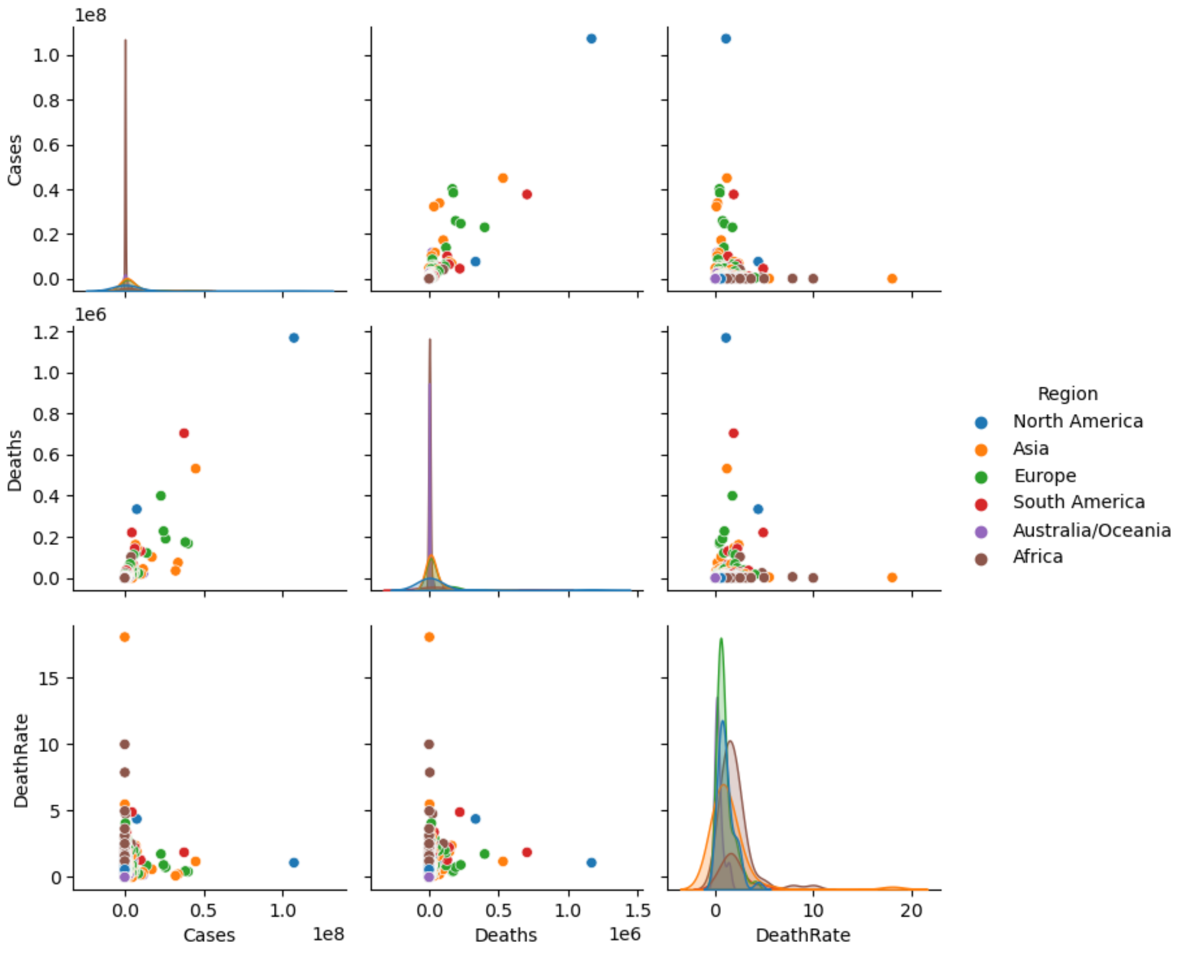


In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 229 entries, 0 to 228
Data columns (total 5 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 
 4   DeathRate  229 non-null    float64
dtypes: float64(1), object(4)
memory usage: 10.7+ KB
```

```
In [10]: import seaborn as sns
         sns.pairplot(df, hue = 'Region')
```

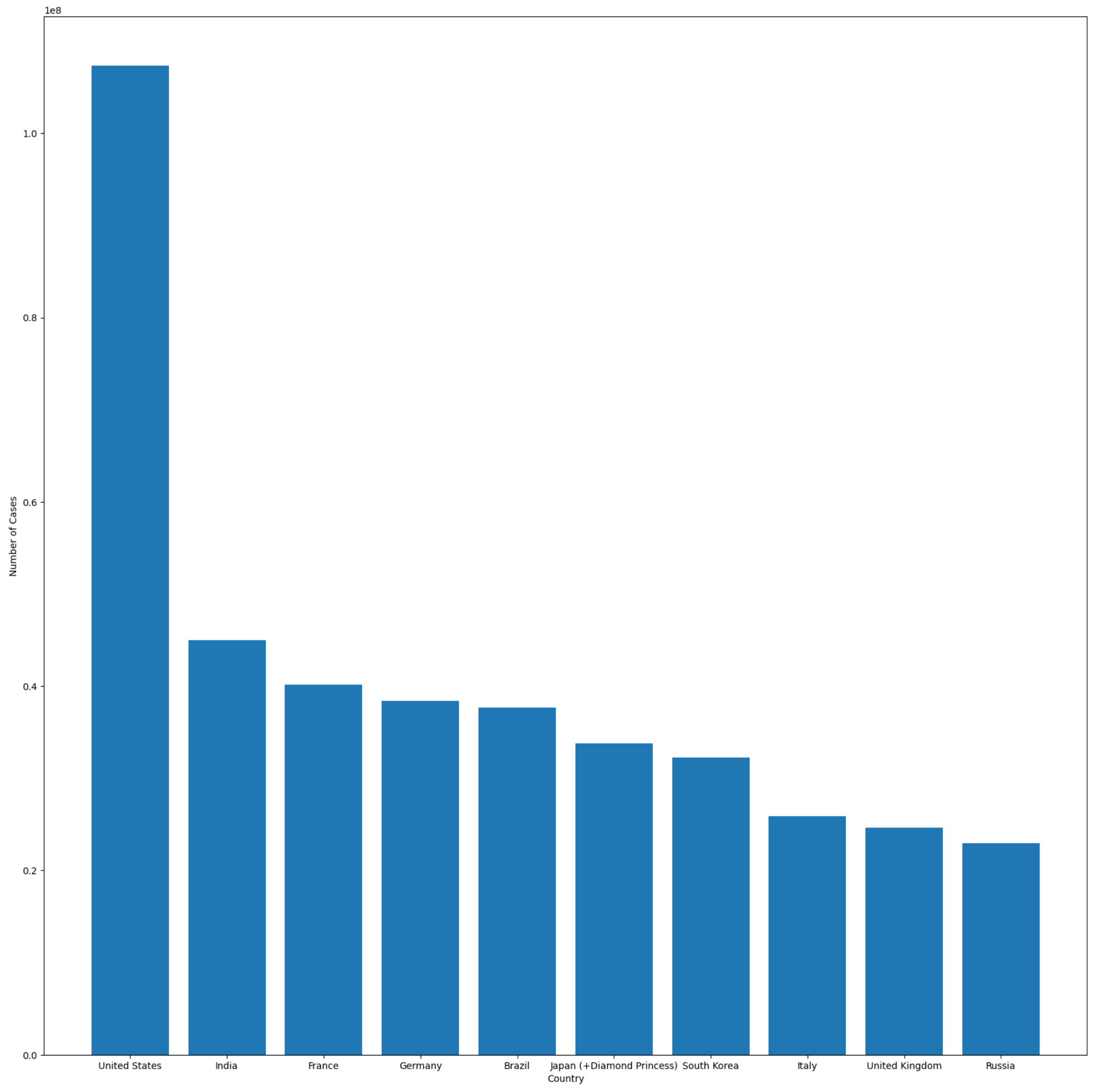
```
Out[10]: <seaborn.axisgrid.PairGrid at 0x22a57e34550>
```



6. Data Visualization - barplot

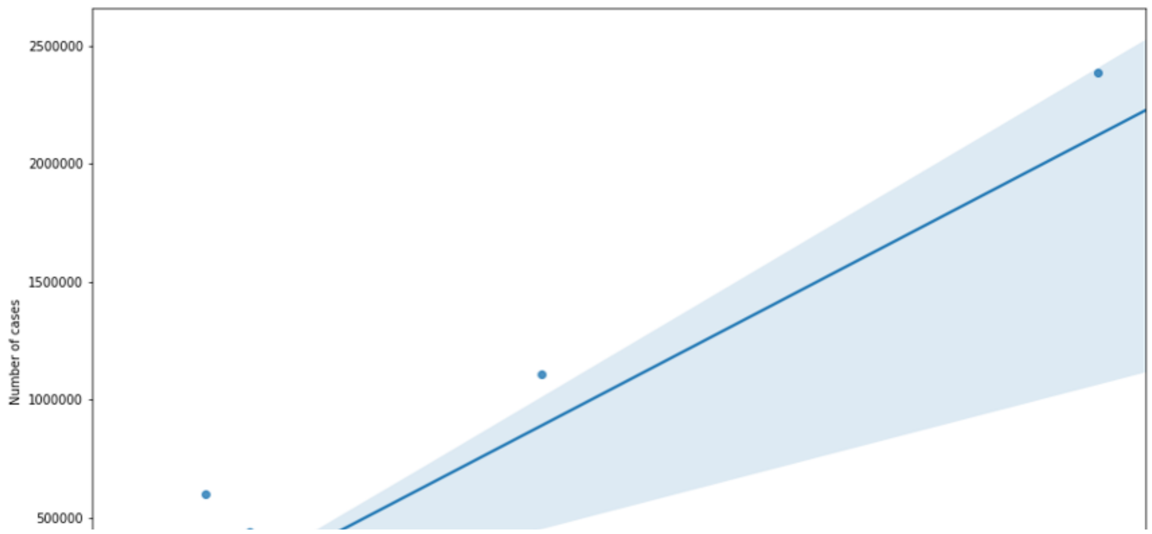
```
<matplotlib.axes._subplots.AxesSubplot at 0x247da3f8b48>
```

In [14]:
```
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(20,20))
ax.bar(df.Country[:10], df.Cases[:10])

ax.set_xlabel('Country')
ax.set_ylabel('Number of Cases')

plt.show()
```



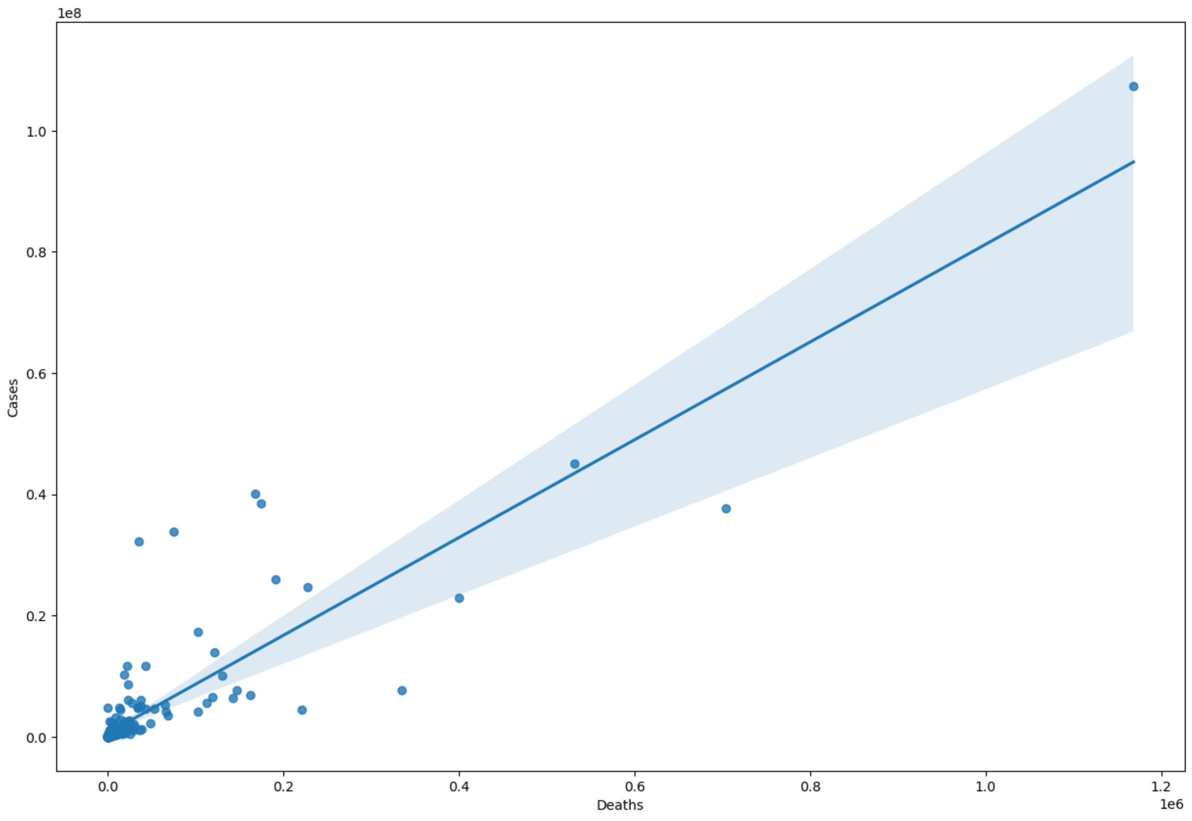
7. Data Visualization - regplot

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



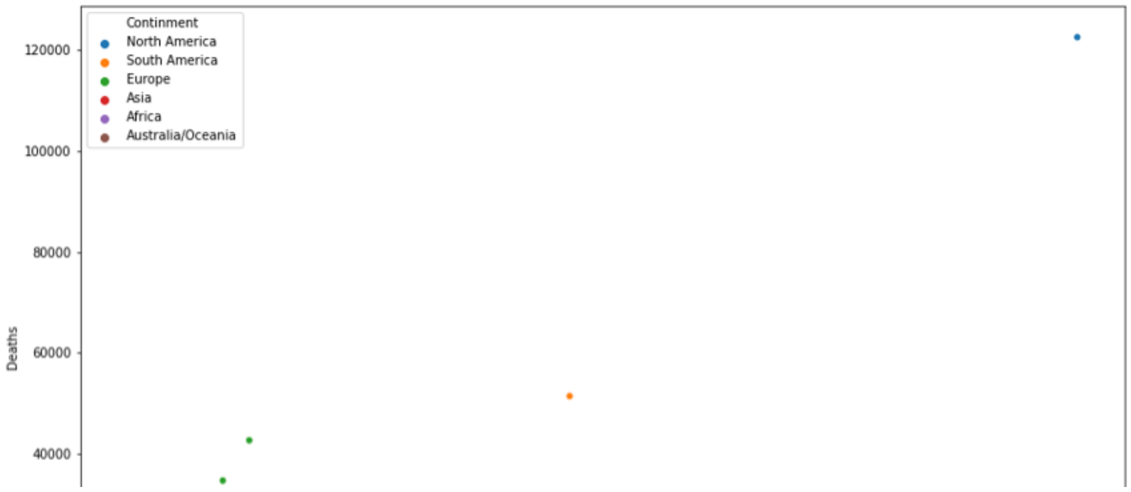
In [24]:
```
plt.figure(figsize = (15, 10))
sns.regplot(x = df['Deaths'], y = df['Cases'], data = df)
```

Out[24]:
```
<Axes: xlabel='Deaths', ylabel='Cases'>
```



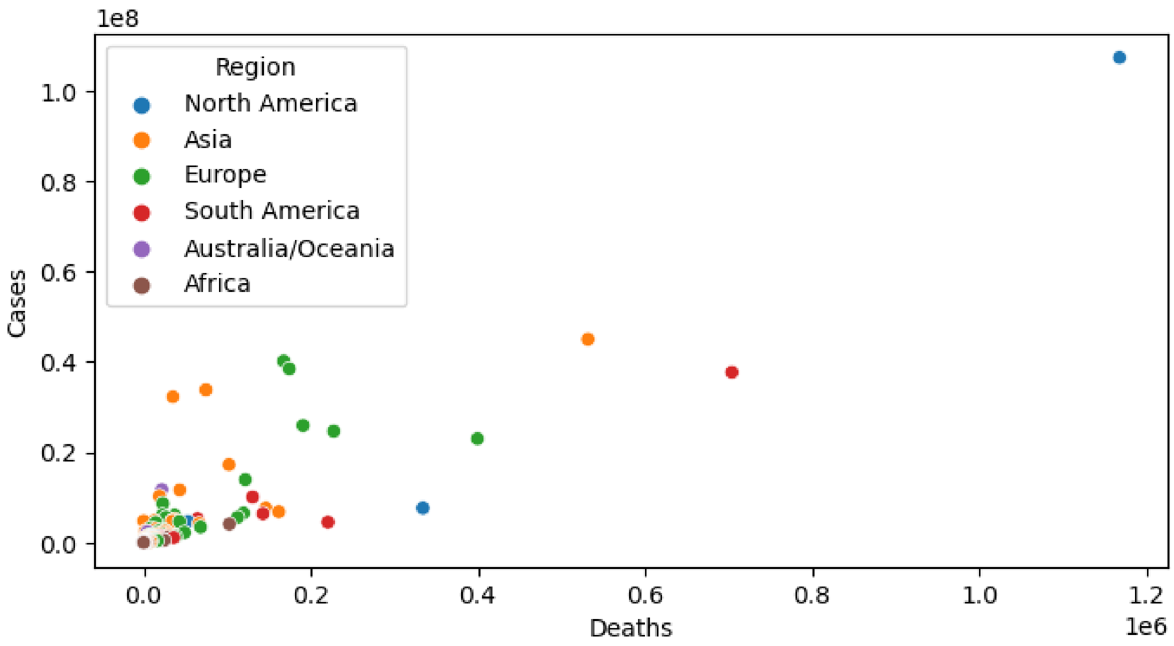
8. Data Visualization - scatterplot

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



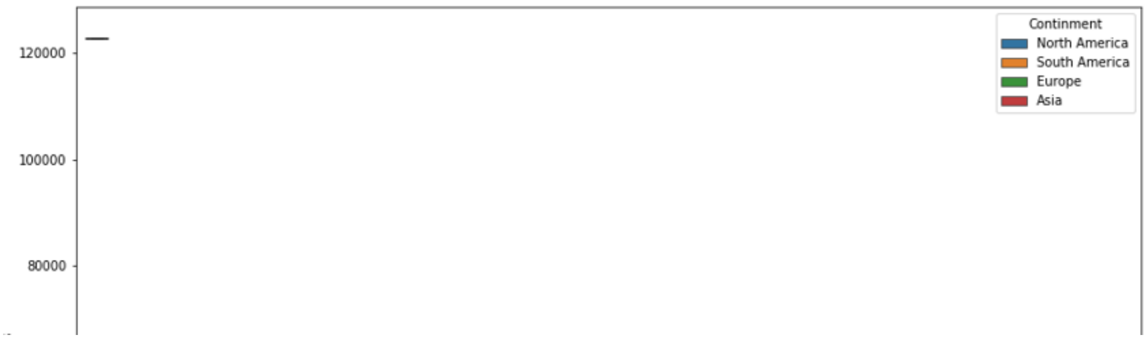
In [25]:
```
plt.figure(figsize = (8, 4))
sns.scatterplot(x = df['Deaths'], y = df['Cases'],
 data = df, hue = 'Region')
```

Out[25]:
```
<Axes: xlabel='Deaths', ylabel='Cases'>
```



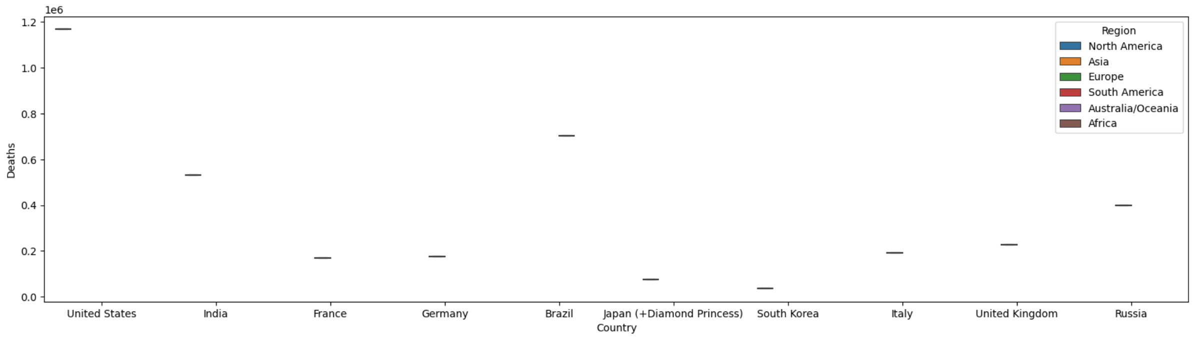
9. Data Visualization - boxplot

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



In [30]:
```
plt.figure(figsize = (20,5))
sns.boxplot(x = df['Country'].head(10),
 y = df['Deaths'].head(10), data = df, hue = 'Region')
```

Out[30]:
```
<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 [36]:
```
groupedData=df.groupby(['Region'])['Cases','Deaths','DeathRate'].sum()
groupedData
```

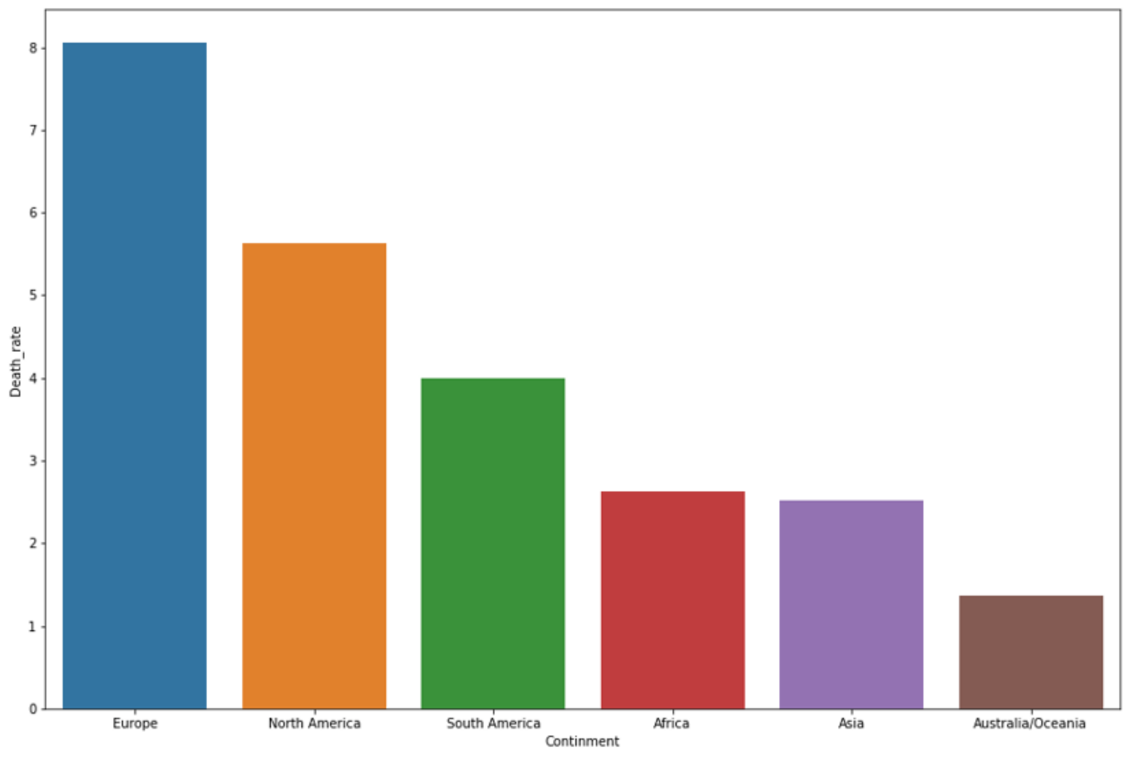
```
C:\Users\Asus\AppData\Local\Temp\ipykernel_24324\671437134.py:1: FutureWarnin
g: Indexing with multiple keys (implicitly converted to a tuple of keys) will
be deprecated, use a list instead.
  groupedData=df.groupby(['Region'])['Cases','Deaths','DeathRate'].sum()
```

Out[36]:

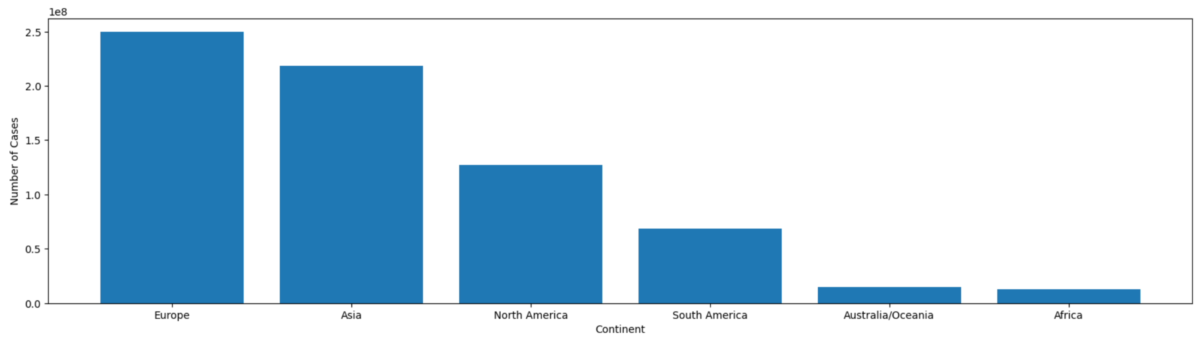
Region	Cases	Deaths	DeathRate
Africa	12830615	258804	110.769851
Asia	218283918	1547796	68.717365
Australia/Oceania	14538582	29206	6.586907
Europe	249684134	2067034	43.892580
North America	127002143	1637367	41.856031
South America	68831885	1357665	24.933194

11. Data Visualization - barplot with death rate

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



In [41]:
```python
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(20,5))
groupedData=groupedData.sort_values(by='Cases',ascending=False)
ax.bar(groupedData.index, groupedData.Cases)

ax.set_xlabel('Continent')
ax.set_ylabel('Number of Cases')

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      |         9         |    0     | Australia/Oceania |
+---------------------------+-------------------+----------+-------------------+
```

In [7]:
```
import texttable as tt
data = df.head(10)
table = tt.Texttable()
cases = df['Cases']
deaths = df['Deaths']
region = df['Region']
country = df['Country']
rows = [['Country', 'Number of 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           | Number of Cases | Deaths  |    Region     |
+============================+=================+=========+===============+
| United States              | 1.073e+08       | 1168278 | 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        |
+----------------------------+-----------------+---------+---------------+
```