Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Webscrapping and Data Visualization

Name: Chong Mun Chen

IC Number: 960327-07-5097

Date : 7/7/2023

Introduction : Practising more on Webscraping, and data visualization with Matplotlib library, Seaborn package and TextTable module.

Conclusion : I know a little more about using Seaborn and TextTable in visualizing the data in addition to using Matplotlib graphs.

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]:

```
from bs4 import BeautifulSoup
from selenium import webdriver
```

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 [115]:

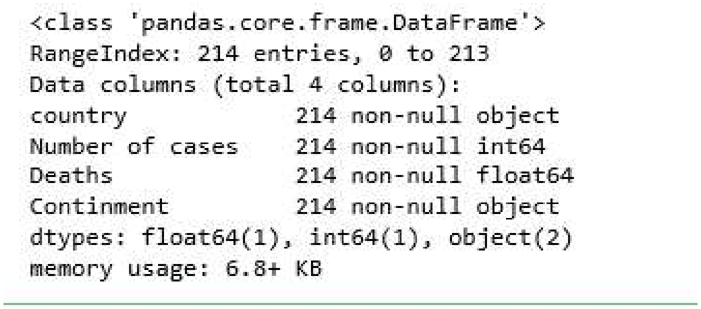
```
import pandas as pd

driver = webdriver.Chrome('C:\\Users\ACER\Desktop\ChromeDriver\chromedriver')
url = "https://www.worldometers.info/coronavirus/countries-where-coronavirus-has
driver.get(url)
data = []
soup = BeautifulSoup(driver.page_source,'html.parser')
for tbody in soup.find_all('tbody'):
    for tr in tbody.find_all('tr'):
        for td in tr.find_all('td'):
            data.append(td.text.rstrip())

driver.quit()

split_data = [data[i:i+4] for i in range(0, len(data), 4)]
df = pd.DataFrame(split_data, columns = ['Country', 'Number of Cases', 'Deaths',
df
```

Out[115]:

	Country	Number of Cases	Deaths	Continent
0	United States	107,355,576	1,168,501	North America
1	India	44,994,494	531,912	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	821	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 × 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 [116]:

```
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   Number of Cases  230 non-null    object
 2   Deaths           230 non-null    object
 3   Continent        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 [117]:

```
df['Deaths'] = df['Deaths'].str.replace(',', '')
df['Number of Cases'] = df['Number of Cases'].str.replace(',', '')

df['Deaths'] = pd.to_numeric(df['Deaths'])
df['Number of Cases'] = pd.to_numeric(df['Number of Cases'])

df['Death Rate'] = 100 * df['Deaths'] / df['Number of Cases']

new_df = df[df.Continent != '']
# or new_df = df[df['Continent'] != '']

new_df
```

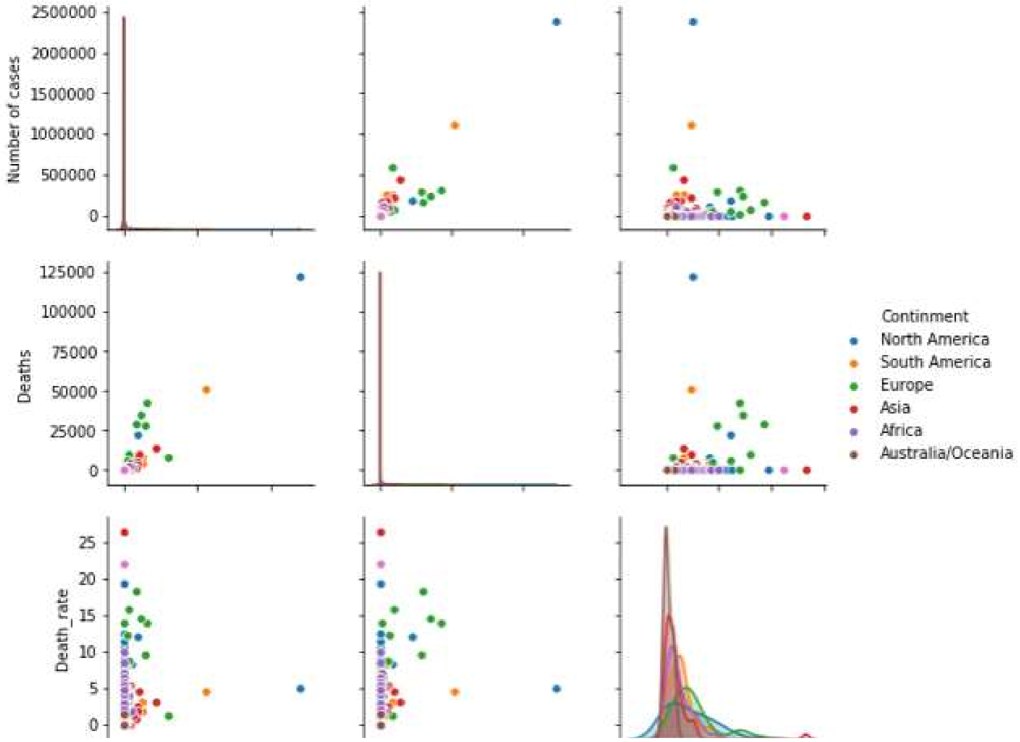
Out[117]:

	Country	Number of Cases	Deaths	Continent	Death Rate
0	United States	107355576	1168501	North America	1.088440
1	India	44994494	531912	Asia	1.182171
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	821	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

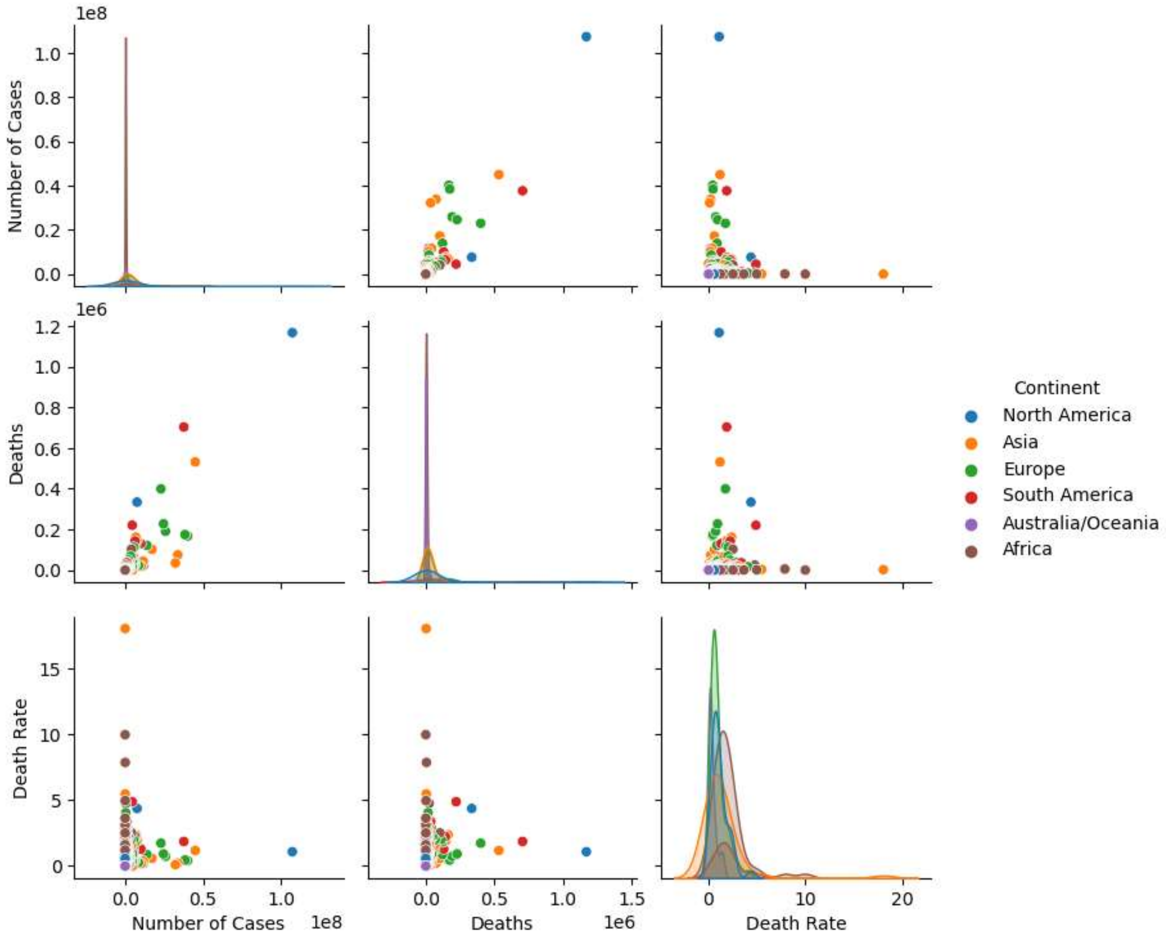
229 rows × 5 columns

5. Data Visualization - Pairplot

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

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


Number of cases
2500000
2000000
1500000
1000000
500000
0
Deaths
125000
100000
75000
50000
25000
0
Death_rate
25
20
15
10
5
0
Continment
North America
South America
Europe
Asia
Africa
Australia/Oceania

```
In [118]:  import seaborn as sns
           import matplotlib.pyplot as plt

           sns.pairplot(new_df, hue = 'Continent')

Out[118]: <seaborn.axisgrid.PairGrid at 0x26b0a0e72e0>
```

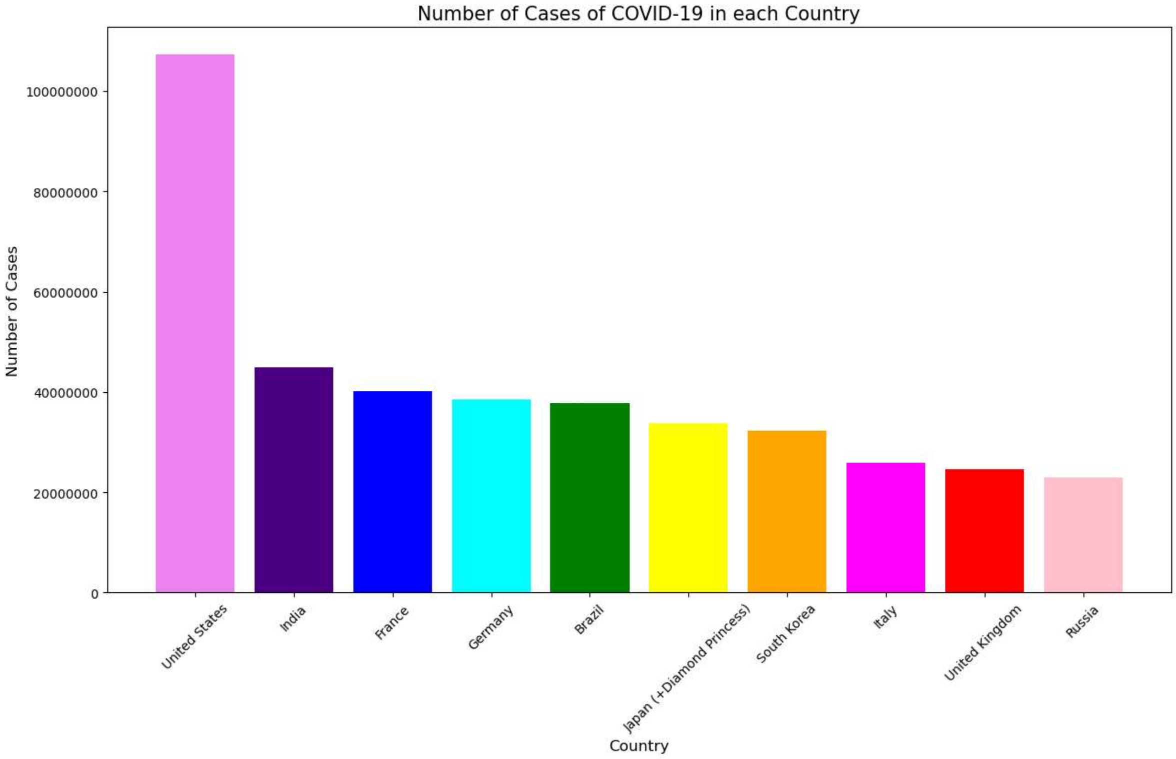


6. Data Visualization - barplot

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

In [119]:

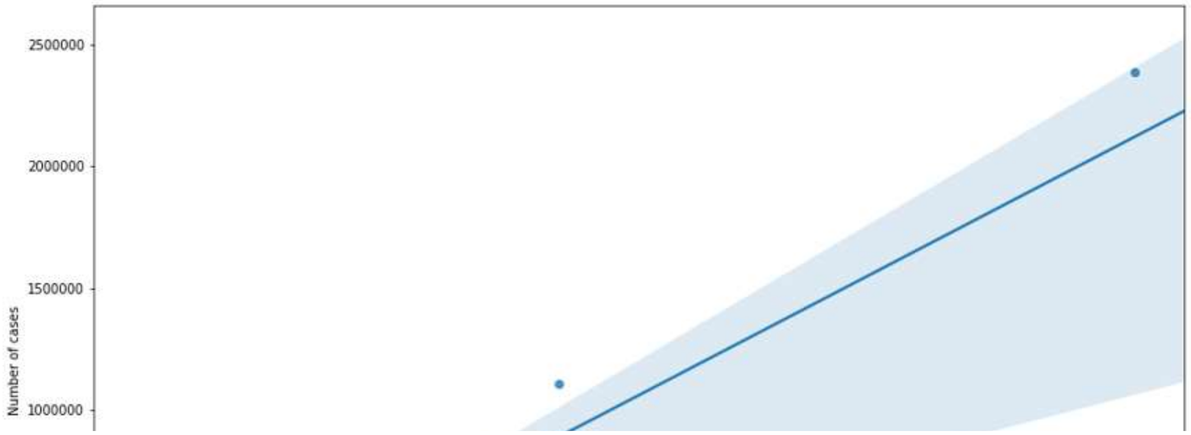
```
import matplotlib.pyplot as plt

plt.figure(figsize = (15,8))
plt.bar(new_df['Country'].head(10), new_df['Number of Cases'].head(10),
 color = ['violet', 'indigo', 'blue', 'cyan', 'green', 'yellow',
          'orange', 'magenta', 'red', 'pink'])
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.title('Number of Cases of COVID-19 in each Country', fontsize = 15)
plt.xlabel('Country', fontsize = 12)
plt.ylabel('Number of Cases', fontsize = 12)
plt.xticks(rotation=45)

plt.show()
plt.tight_layout()
```



```
<Figure size 640x480 with 0 Axes>
```

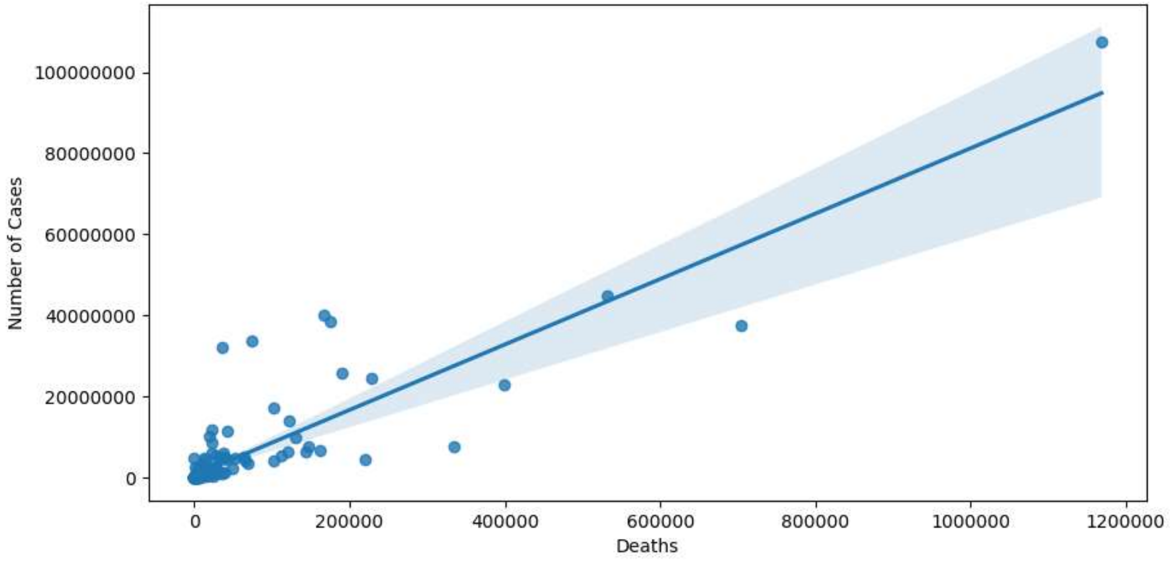
7. Data Visualization - regplot

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



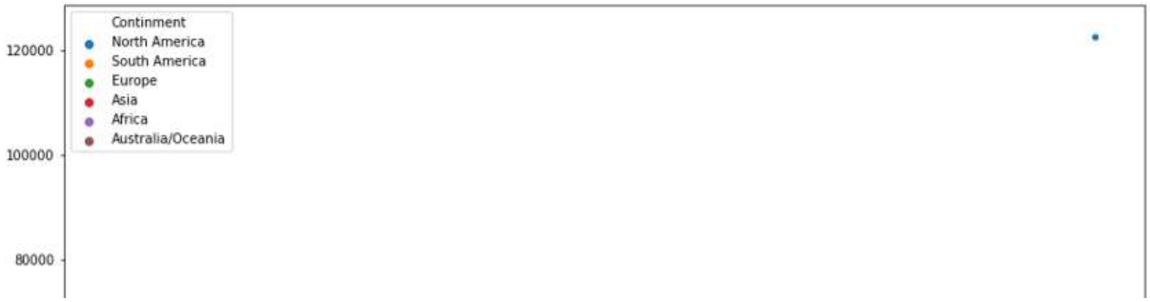
In [120]:

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

plt.figure(figsize = (10, 5))
sns.regplot(x = new_df['Deaths'], y = new_df['Number of Cases'])
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'x')
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.show()
```



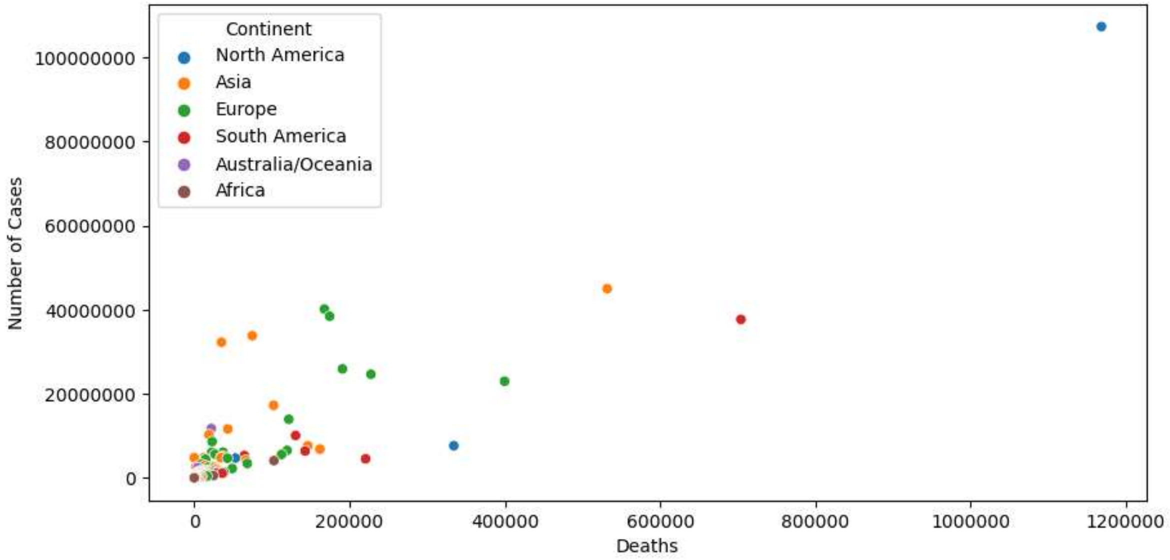
8. Data Visualization - scatterplot

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



In [121]:

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

plt.figure(figsize = (10, 5))
sns.scatterplot(x = new_df['Deaths'], y = new_df['Number of Cases'], hue = new_d
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'x')
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.show()
```

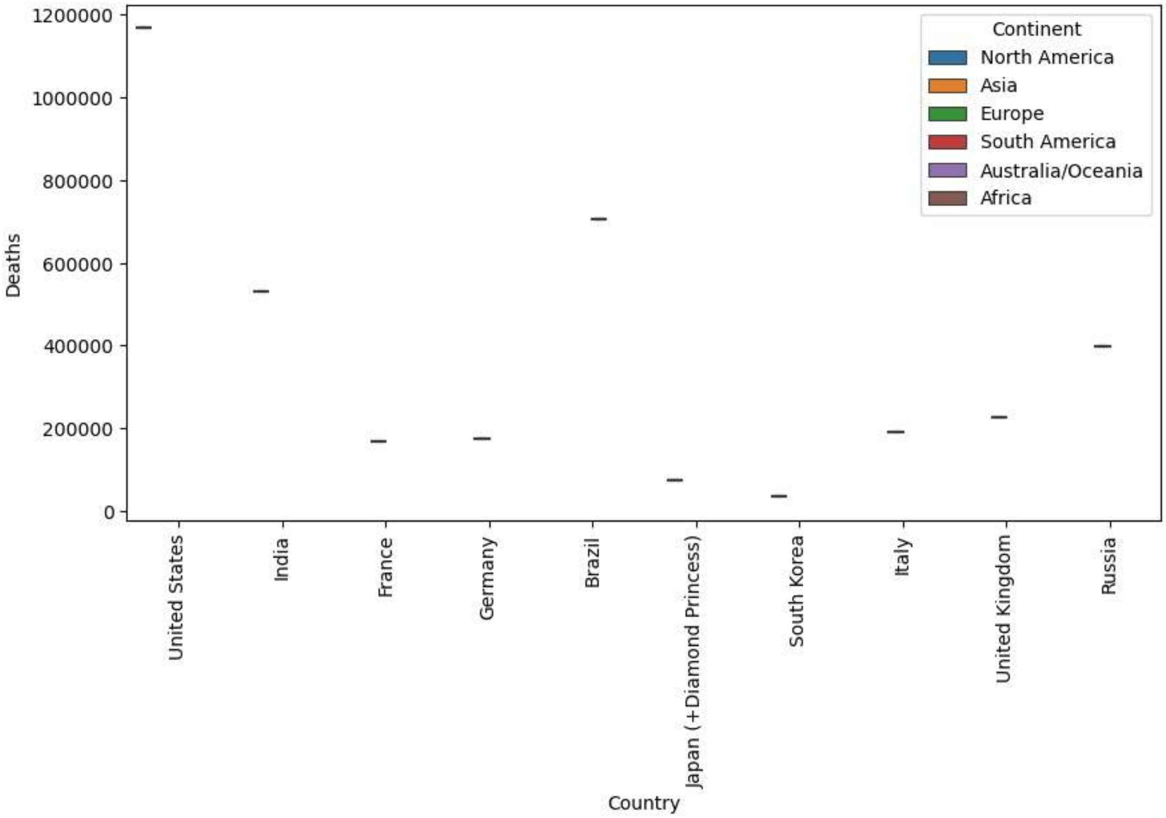


9. Data Visualization - boxplot

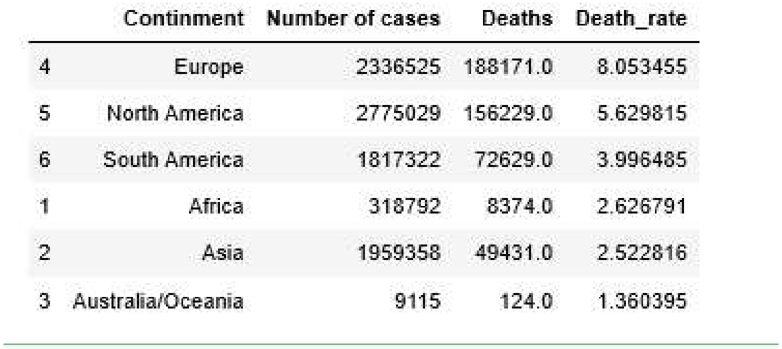
```
matplotlib.axes._subplots.AxesSubplot at 0x247da618a88>
```

In [122]:

```python
import seaborn as sns
import matplotlib.pyplot as plt

plt.figure(figsize = (10, 5))
sns.boxplot(x = new_df['Country'].head(10), y = new_df['Deaths'].head(10), hue =
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.xticks(rotation=90)
plt.show()
```



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 [124]:

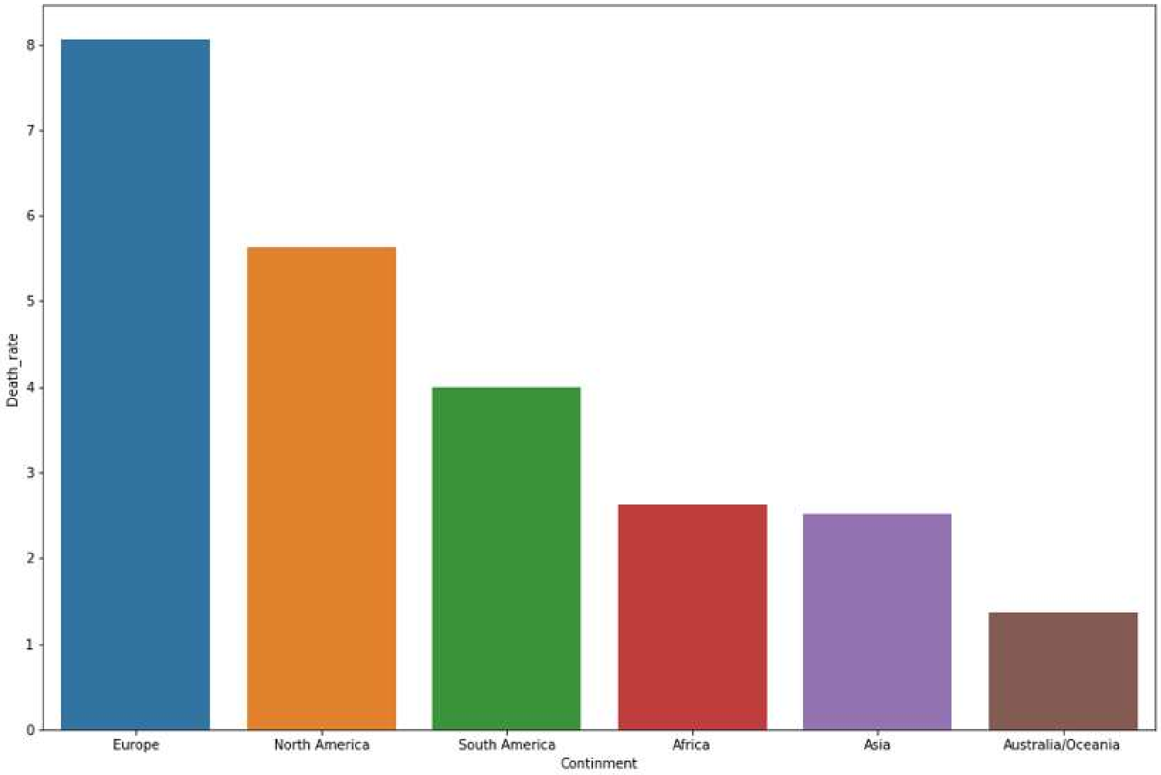
```
continent_df = new_df.groupby(['Continent'])[['Number of Cases', 'Deaths', 'Deat
continent_df = continent_df.sort_values('Death Rate', ascending=False)
continent_df
```

Out[124]:

	Continent	Number of Cases	Deaths	Death Rate
0	Africa	12831574	258806	110.757679
1	Asia	218289948	1547823	68.687709
3	Europe	249686971	2067126	43.896244
4	North America	127033942	1637656	41.865989
5	South America	68833395	1357698	24.933053
2	Australia/Oceania	14552582	29336	6.583591

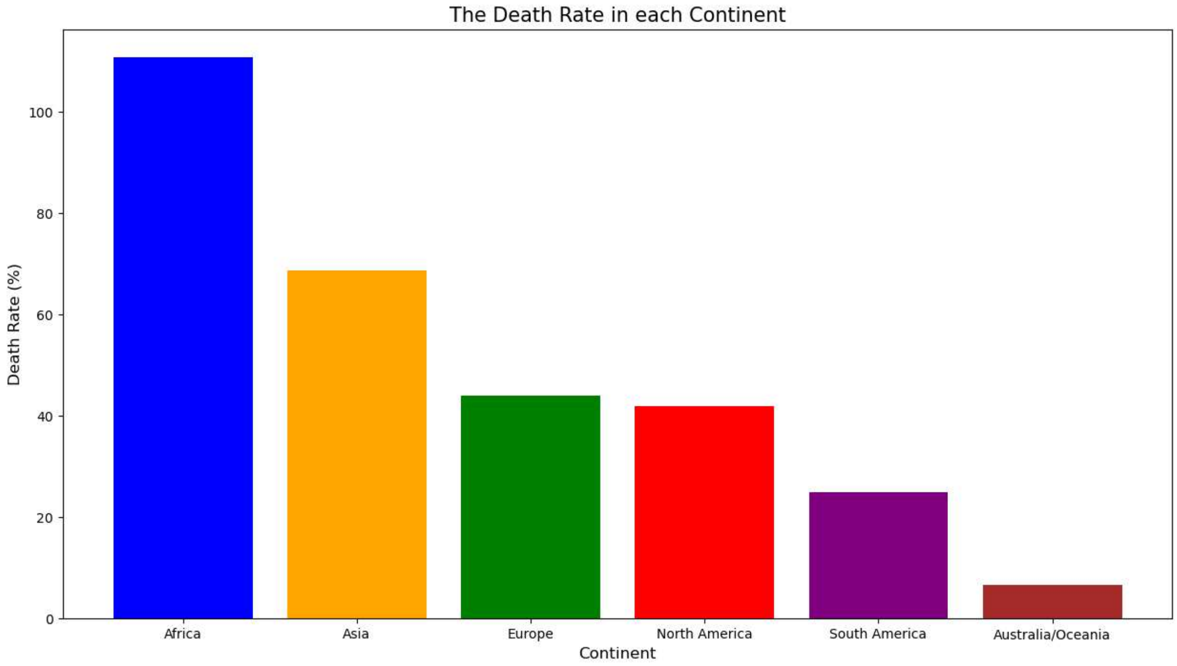
11. Data Visualization - barplot with death rate

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



In [125]:

```
import matplotlib.pyplot as plt

plt.figure(figsize = (15,8))
plt.bar(continent_df['Continent'], continent_df['Death Rate'],
 color = ['blue', 'orange', 'green', 'red', 'purple', 'brown'])
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.title('The Death Rate in each Continent', fontsize = 15)
plt.xlabel('Continent', fontsize = 12)
plt.ylabel('Death Rate (%)', fontsize = 12)

plt.show()
plt.tight_layout()
```



```
<Figure size 640x480 with 0 Axes>
```

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   |
+===========================+=================+=========+===============+
```

In [129]:

```
import texttable as tt

data = df.head(8)
table = tt.Texttable()
country = data['Country']
cases = data['Number of Cases']
deaths = data['Deaths']
continent = data['Continent']
rows = [['Country', 'Number of Cases', 'Deaths', 'Continent']]

for i in range(8):
    rows.append([country[i], cases[i], deaths[i], continent[i]])

table.add_rows(rows)
print(tb.draw())
```

```
+---------------------------+-----------------+---------+---------------+
|          Country          | Number of Cases | Deaths  |   Continent   |
+===========================+=================+=========+===============+
| United States             | 1.074e+08       | 1168501 | North America |
+---------------------------+-----------------+---------+---------------+
| India                     | 44994494        | 531912  | 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        |
+---------------------------+-----------------+---------+---------------+
```