Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Webscrapping and Data Visualization

Name: Chuay Xiang Ze

IC Number: 021224070255

Date : 6/7/2023

Introduction : Learning how to webscrape and visualize data using seaborn

Conclusion : Managed to complete tasks related to this topic

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]:

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

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]:

```
import pandas as pd
driver = webdriver.Chrome('C:\\Users\Xiang Ze\Downloads\chromedriver_win32\chromedriver.

url='https://www.worldometers.info/coronavirus/countries-where-coronavirus-has-spread/'

driver.get(url)

soup = BeautifulSoup(driver.page_source, 'html.parser')

data = []
for tr in soup.find_all('tr', attrs={'role': 'row'}):
    for td in tr.find_all('td'):
        data.append(td.text.rstrip())
data

df_data = []
for i in range(0, len(data), 4):
    country = data[i]
    total_cases = data[i+1]
    total_deaths = data[i+2]
    continent = data[i+3]
    df_data.append([country, total_cases, total_deaths, continent])

df = pd.DataFrame(df_data, columns=['Country', 'Total Cases', 'Total Deaths', 'Continent
df
```

Out[4]:

	Country	Total Cases	Total 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):
```

In [27]:

```
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   Total Cases   230 non-null    object
 2   Total 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 [9]:

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

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

df['Death Rate'] = df['Total Deaths'] / df['Total Cases']
df['Death Rate'] = df['Death Rate'] * 100
df= df[df.Continent != '']
df
```

```
C:\Users\Xiang Ze\AppData\Local\Temp\ipykernel_7316\556933275.py:1: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df['Total Deaths'] = df['Total Deaths'].str.replace(',', '')
C:\Users\Xiang Ze\AppData\Local\Temp\ipykernel_7316\556933275.py:2: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df['Total Cases'] = df['Total Cases'].str.replace(',', '')
C:\Users\Xiang Ze\AppData\Local\Temp\ipykernel_7316\556933275.py:4: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df['Total Deaths'] = pd.to_numeric(df['Total Deaths'])
C:\Users\Xiang Ze\AppData\Local\Temp\ipykernel_7316\556933275.py:5: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df['Total Cases'] = pd.to_numeric(df['Total Cases'])
C:\Users\Xiang Ze\AppData\Local\Temp\ipykernel_7316\556933275.py:7: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df['Death Rate'] = df['Total Deaths'] / df['Total Cases']
C:\Users\Xiang Ze\AppData\Local\Temp\ipykernel_7316\556933275.py:8: Settin
gWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://
pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-
view-versus-a-copy)
  df['Death Rate'] = df['Death Rate'] * 100
```

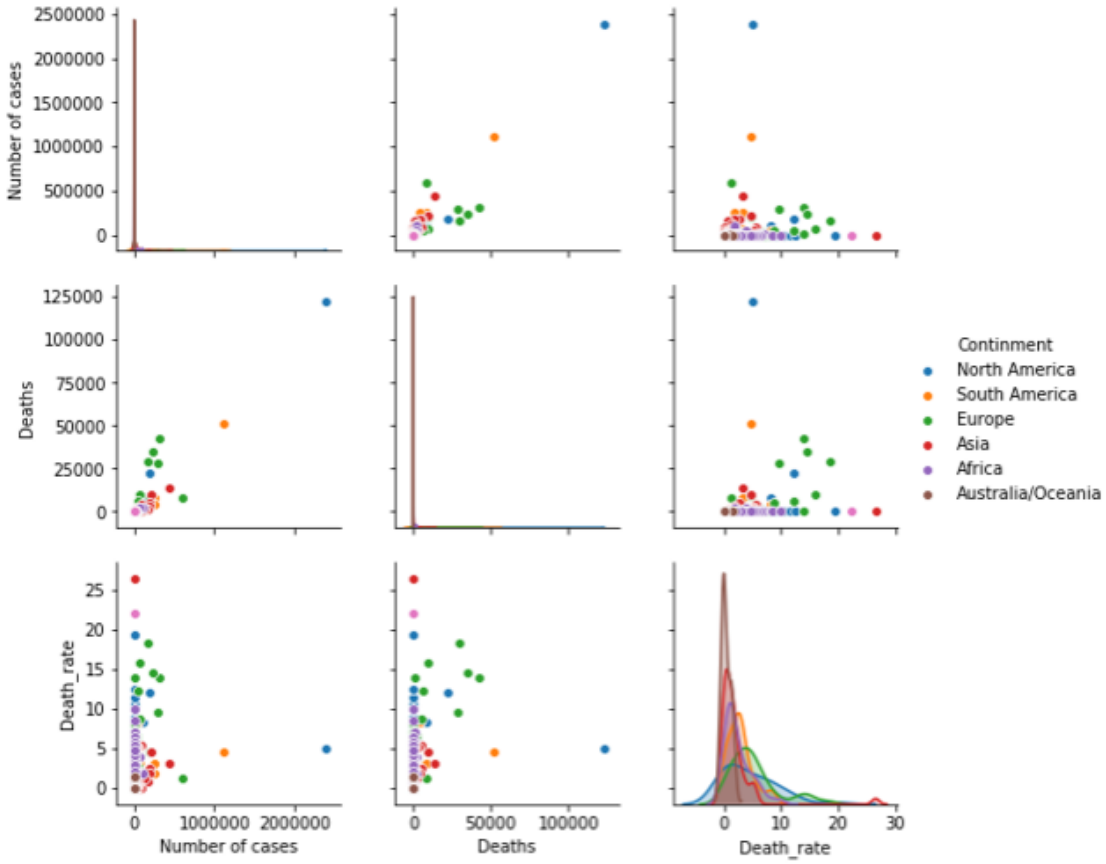
Out[9]:

	Country	Total Cases	Total Deaths	Continent	Death Rate
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>

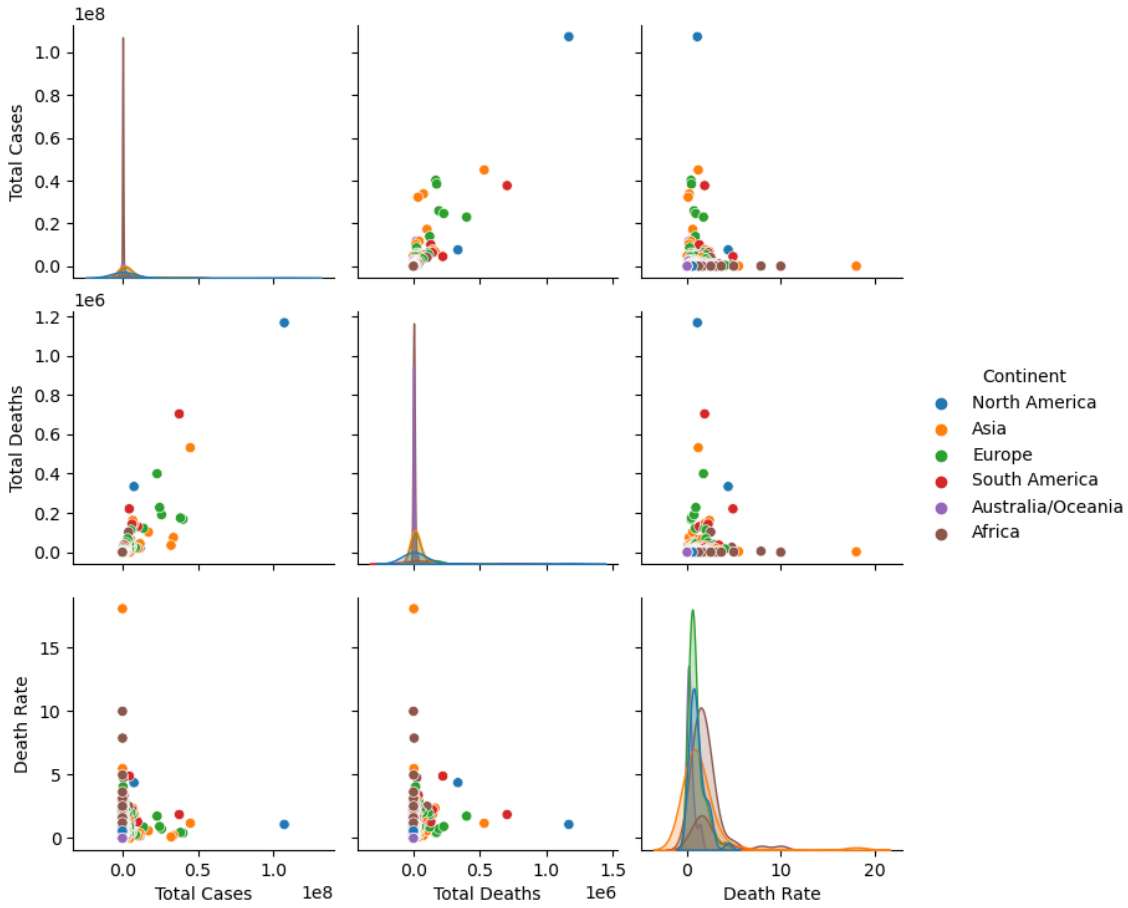


In [10]:

```
import seaborn as sns

sns.pairplot(df, hue = 'Continent')
```

Out[10]:

```
<seaborn.axisgrid.PairGrid at 0x218d7d8aa90>
```

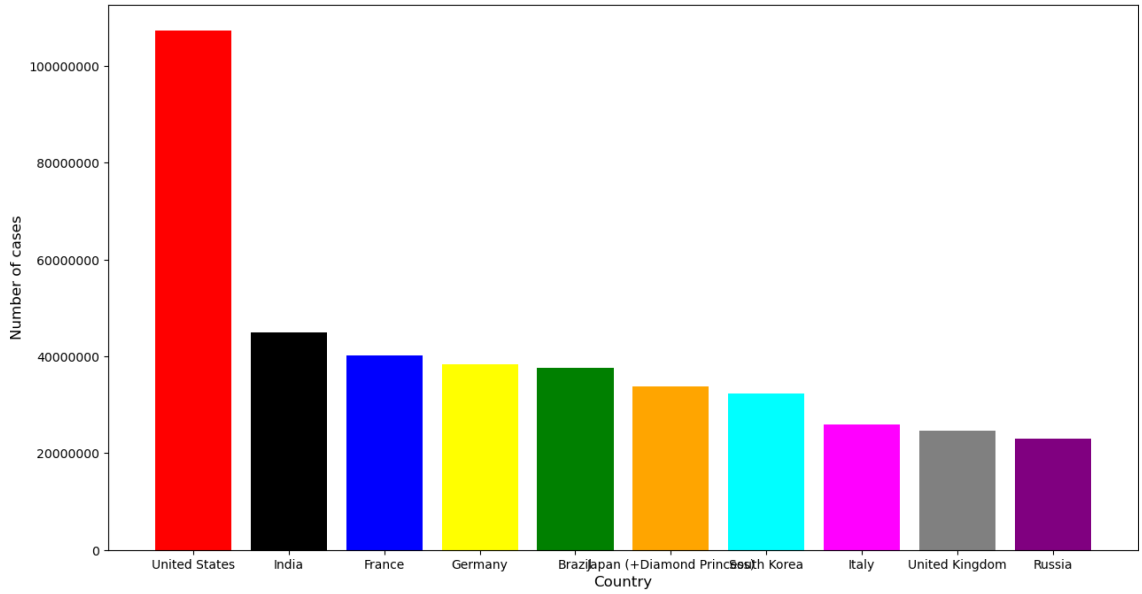


6. Data Visualization - barplot

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

In [19]:

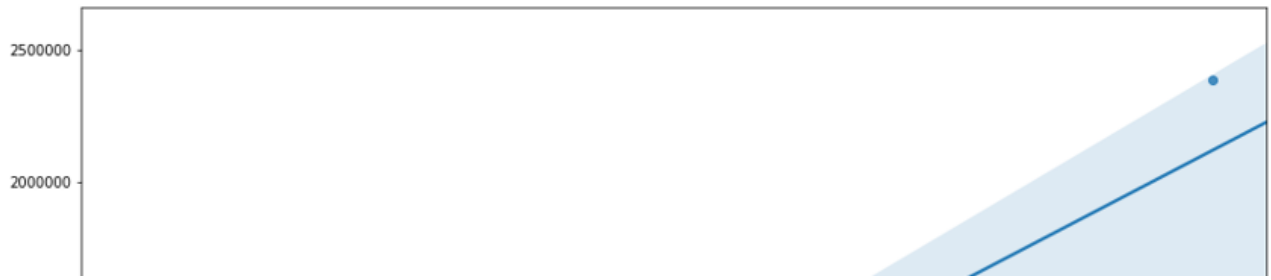
```
import matplotlib.pyplot as plt

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

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




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

7. Data Visualization - regplot

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

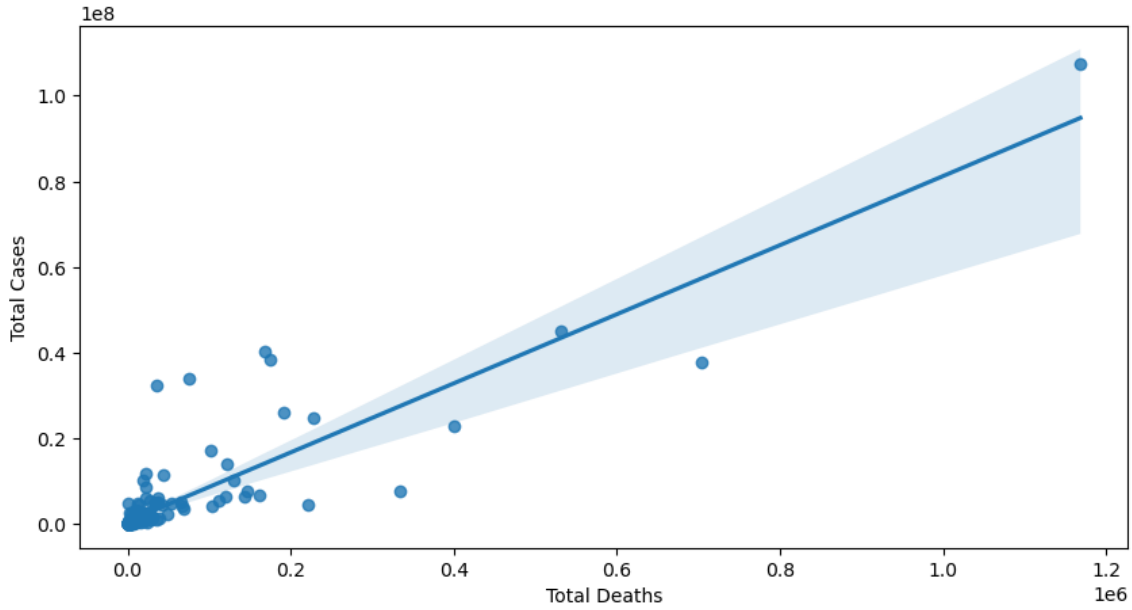


In [22]:

```
plt.figure(figsize = (10, 5))
sns.regplot(x = df['Total Deaths'], y = df['Total Cases'])
```

Out[22]:

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



8. Data Visualization - scatterplot

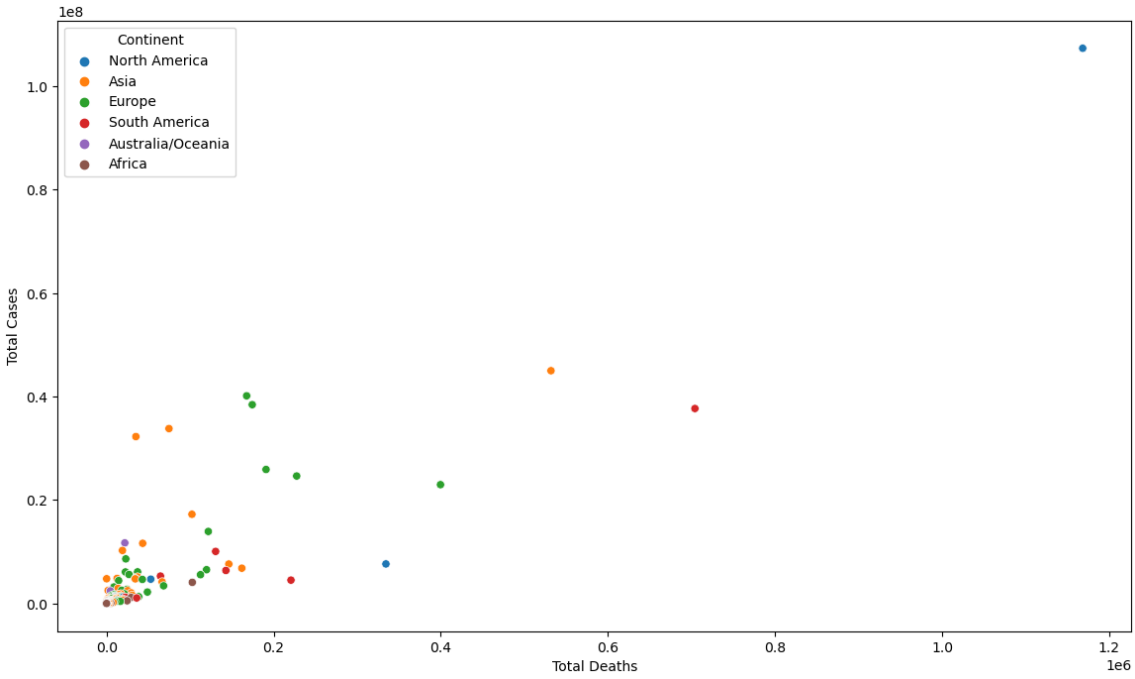
```
<matplotlib.axes._subplots.AxesSubplot at 0x247da544748>
```

In [31]:

```
plt.figure(figsize = (14, 8))
sns.scatterplot(x = df['Total Deaths'], y = df['Total Cases'], hue = df['Continent'])
```

Out[31]:

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



9. Data Visualization - boxplot

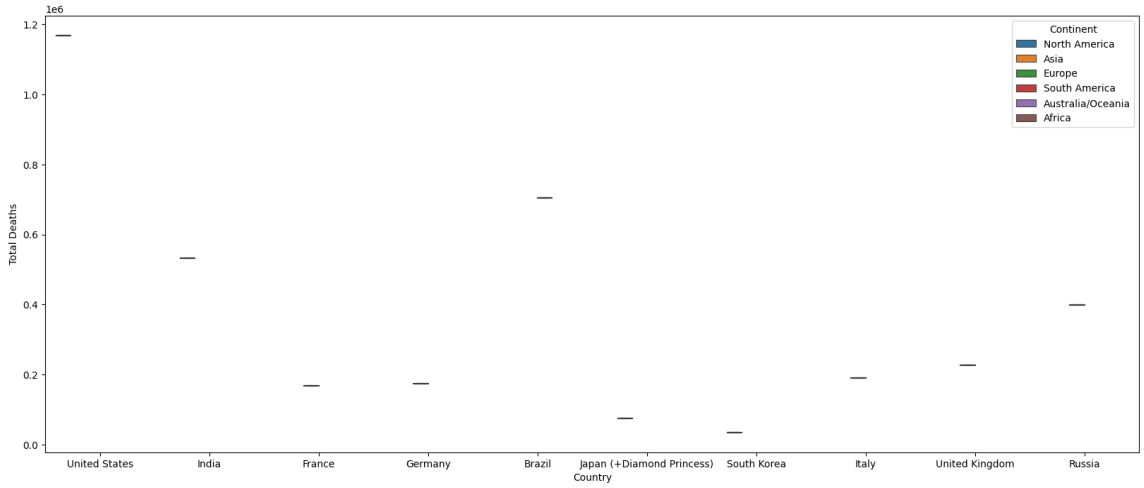
```
matplotlib.axes._subplots.AxesSubplot at 0x247da618a88>
```

In [35]:

```
plt.figure(figsize = (20, 8))
sns.boxplot(x = df['Country'].head(10),y = df['Total Deaths'].head(10), hue = df['Contin
```

Out[35]:

```
<Axes: xlabel='Country', ylabel='Total 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 [44]:

```
df_grouped = df.groupby('Continent')[('Total Cases', 'Total Deaths', 'Death Rate')].sum(
df_grouped = df_grouped.reset_index().sort_values('Death Rate', ascending=False)
df_grouped
```

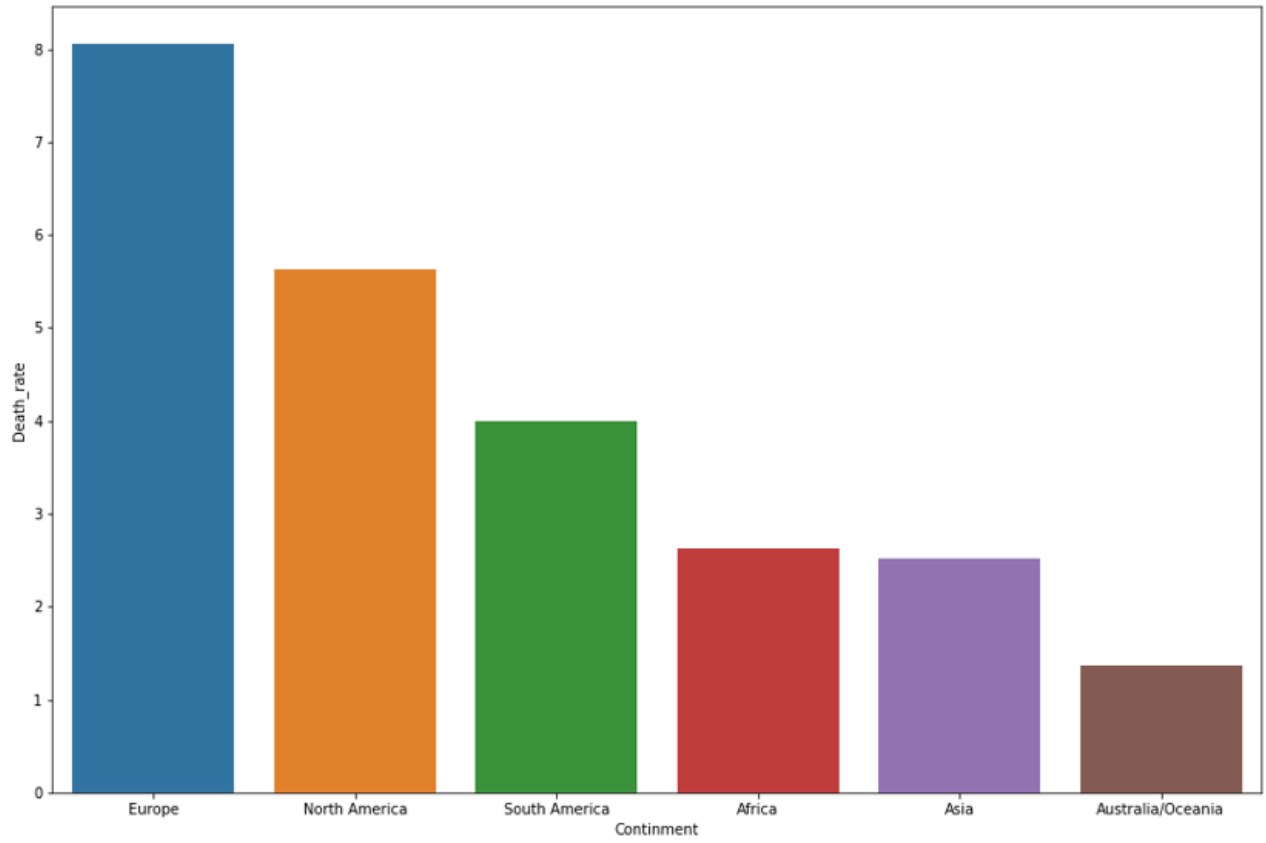
```
C:\Users\Xiang Ze\AppData\Local\Temp\ipykernel_7316\2267533703.py:1: Futur
eWarning: Indexing with multiple keys (implicitly converted to a tuple of
keys) will be deprecated, use a list instead.
  df_grouped = df.groupby('Continent')[('Total Cases', 'Total Deaths', 'De
ath Rate')].sum()
```

Out[44]:

	Continent	Total Cases	Total Deaths	Death Rate
0	Africa	12830615	258804	110.769851
1	Asia	218283918	1547796	68.717365
3	Europe	249684134	2067034	43.892580
4	North America	127002143	1637367	41.856031
5	South America	68831885	1357665	24.933194
2	Australia/Oceania	14538582	29206	6.586907

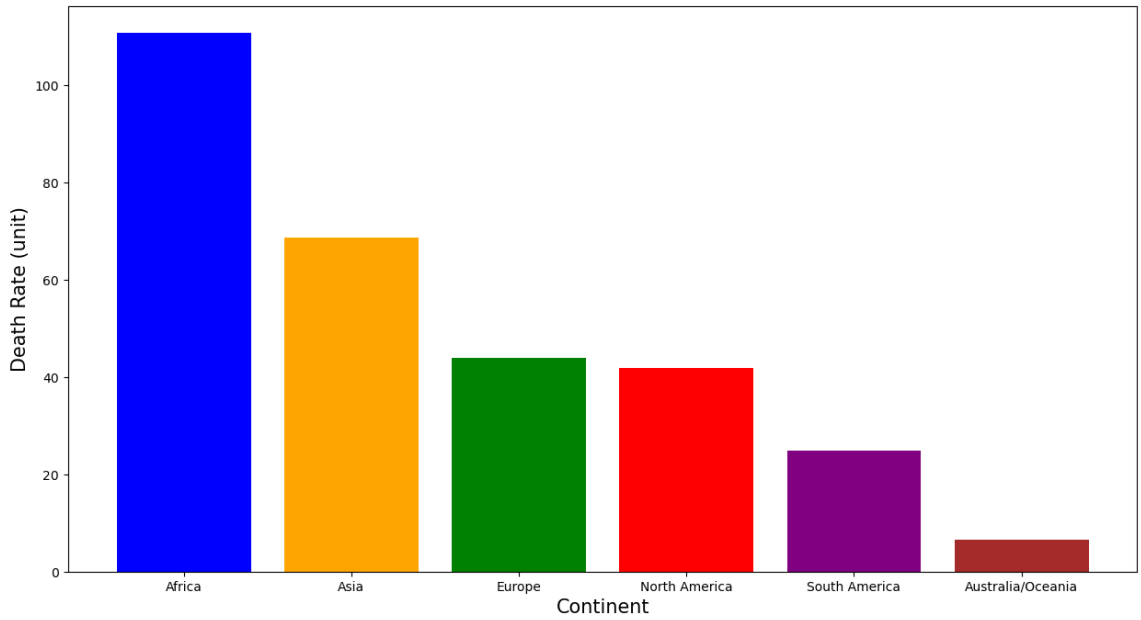
11. Data Visualization - barplot with death rate

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



In [49]:

```
plt.figure(figsize = (15,8))
plt.bar(df_grouped['Continent'], df_grouped['Death Rate'],
 color = ['blue', 'orange', 'green', 'red', 'purple',
 'brown'])
plt.ticklabel_format(useOffset = False, style = 'plain', axis = 'y')
plt.xlabel('Continent', fontsize = 15)
plt.ylabel('Death Rate (unit)', fontsize = 15)
plt.show()

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

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



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

In [18]:

```
import texttable as tt
data = df.head(8)
tb = tt.Texttable()
cases = data['Total Cases']
deaths = data['Total Deaths']
continent = data['Continent']
country = data['Country']
rows = [['Country', 'Number of Cases', 'Deaths', 'Continent']]
for i in range(8):
     rows.append([country[i], cases[i], deaths[i], continent[i]])
tb.add_rows(rows)
print(tb.draw())
```

```
+--------------------------+-----------------+-----------+---------------
+
|          Country         | Number of Cases |  Deaths   |   Continent
|
+==========================+=================+===========+===============
+
| United States            | 107,355,576     | 1,168,501 | North America
|
+--------------------------+-----------------+-----------+---------------
+
| India                    | 44,994,494      | 531,912   | Asia
|
+--------------------------+-----------------+-----------+---------------
+
| France                   | 40,138,560      | 167,642   | Europe
|
+--------------------------+-----------------+-----------+---------------
+
| Germany                  | 38,428,685      | 174,352   | Europe
|
+--------------------------+-----------------+-----------+---------------
+
| Brazil                   | 37,682,660      | 704,159   | South America
|
+--------------------------+-----------------+-----------+---------------
+
| Japan (+Diamond Princess) | 33,804,284      | 74,707    | Asia
|
+--------------------------+-----------------+-----------+---------------
+
| South Korea              | 32,256,154      | 35,071    | Asia
|
+--------------------------+-----------------+-----------+---------------
+
| Italy                    | 25,897,801      | 190,868   | Europe
|
+--------------------------+-----------------+-----------+---------------
+
```

In [ ]:

```

```