

covid-19-analysis

February 1, 2024

```
[29]: !pip install numpy  
      !pip install pandas  
      !pip install seaborn  
      !pip install matplotlib  
      !pip install plotly
```

Requirement already satisfied: numpy in c:\python\lib\site-packages (1.26.3)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2

[notice] To update, run: python.exe -m pip install --upgrade pip

Requirement already satisfied: pandas in c:\python\lib\site-packages (2.2.0)

Requirement already satisfied: numpy<2,>=1.23.2 in c:\python\lib\site-packages (from pandas) (1.26.3)

Requirement already satisfied: python-dateutil>=2.8.2 in c:\python\lib\site-packages (from pandas) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in c:\python\lib\site-packages (from pandas) (2023.4)

Requirement already satisfied: tzdata>=2022.7 in c:\python\lib\site-packages (from pandas) (2023.4)

Requirement already satisfied: six>=1.5 in c:\python\lib\site-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2

[notice] To update, run: python.exe -m pip install --upgrade pip

Requirement already satisfied: seaborn in c:\python\lib\site-packages (0.13.2)

Requirement already satisfied: numpy!=1.24.0,>=1.20 in c:\python\lib\site-packages (from seaborn) (1.26.3)

Requirement already satisfied: pandas>=1.2 in c:\python\lib\site-packages (from seaborn) (2.2.0)

Requirement already satisfied: matplotlib!=3.6.1,>=3.4 in c:\python\lib\site-packages (from seaborn) (3.8.2)

Requirement already satisfied: contourpy>=1.0.1 in c:\python\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (1.2.0)

Requirement already satisfied: cycler>=0.10 in c:\python\lib\site-packages (from matplotlib!=3.6.1,>=3.4->seaborn) (0.12.1)

Requirement already satisfied: fonttools>=4.22.0 in c:\python\lib\site-packages

(from matplotlib!=3.6.1,>=3.4->seaborn) (4.47.2)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (1.4.5)
Requirement already satisfied: packaging>=20.0 in c:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (23.2)
Requirement already satisfied: pillow>=8 in c:\python\lib\site-packages (from
matplotlib!=3.6.1,>=3.4->seaborn) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\python\lib\site-packages
(from matplotlib!=3.6.1,>=3.4->seaborn) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\python\lib\site-
packages (from matplotlib!=3.6.1,>=3.4->seaborn) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\python\lib\site-packages (from
pandas>=1.2->seaborn) (2023.4)
Requirement already satisfied: tzdata>=2022.7 in c:\python\lib\site-packages
(from pandas>=1.2->seaborn) (2023.4)
Requirement already satisfied: six>=1.5 in c:\python\lib\site-packages (from
python-dateutil>=2.7->matplotlib!=3.6.1,>=3.4->seaborn) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2

[notice] To update, run: python.exe -m pip install --upgrade pip

Requirement already satisfied: matplotlib in c:\python\lib\site-packages (3.8.2)
Requirement already satisfied: contourpy>=1.0.1 in c:\python\lib\site-packages
(from matplotlib) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\python\lib\site-packages (from
matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in c:\python\lib\site-packages
(from matplotlib) (4.47.2)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\python\lib\site-packages
(from matplotlib) (1.4.5)
Requirement already satisfied: numpy<2,>=1.21 in c:\python\lib\site-packages
(from matplotlib) (1.26.3)
Requirement already satisfied: packaging>=20.0 in c:\python\lib\site-packages
(from matplotlib) (23.2)
Requirement already satisfied: pillow>=8 in c:\python\lib\site-packages (from
matplotlib) (10.2.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\python\lib\site-packages
(from matplotlib) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in c:\python\lib\site-
packages (from matplotlib) (2.8.2)
Requirement already satisfied: six>=1.5 in c:\python\lib\site-packages (from
python-dateutil>=2.7->matplotlib) (1.16.0)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2

[notice] To update, run: python.exe -m pip install --upgrade pip

Requirement already satisfied: plotly in c:\python\lib\site-packages (5.18.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\python\lib\site-packages

(from plotly) (8.2.3)

Requirement already satisfied: packaging in c:\python\lib\site-packages (from plotly) (23.2)

[notice] A new release of pip is available: 23.2.1 -> 23.3.2

[notice] To update, run: python.exe -m pip install --upgrade pip

```
[30]: # Import necessary libraries

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
[31]: # Load the data

df = pd.read_csv(r"C:\Users\Hitesh Karthik\OneDrive\Desktop\data science_\
projects\Covid-19 Analysis\country_wise_latest.csv")
df.head()
```

```
[31]: Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  \
0      Afghanistan    36263    1269    25198    9796      106      10
1           Albania     4880     144     2745    1991      117       6
2           Algeria    27973    1163    18837    7973      616       8
3           Andorra     907      52      803     52       10       0
4           Angola     950      41      242    667       18       1
```

```
      New recovered  Deaths / 100 Cases  Recovered / 100 Cases  \
0              18              3.50              69.49
1              63              2.95              56.25
2             749              4.16              67.34
3               0              5.73              88.53
4               0              4.32              25.47
```

```
      Deaths / 100 Recovered  Confirmed last week  1 week change  \
0              5.04              35526              737
1              5.25              4171              709
2              6.17             23691             4282
3              6.48              884              23
4             16.94              749              201
```

```
      1 week % increase  WHO Region
0              2.07  Eastern Mediterranean
1             17.00             Europe
2             18.07             Africa
3              2.60             Europe
4             26.84             Africa
```

```
[32]: df.shape
```

```
[32]: (187, 15)
```

```
[33]: df.isna().sum()
```

```
[33]: Country/Region      0
Confirmed              0
Deaths                0
Recovered              0
Active                0
New cases              0
New deaths             0
New recovered          0
Deaths / 100 Cases     0
Recovered / 100 Cases  0
Deaths / 100 Recovered 0
Confirmed last week    0
1 week change          0
1 week % increase      0
WHO Region             0
dtype: int64
```

```
[34]: df.info
```

```
[34]: <bound method DataFrame.info of
Recovered Active New cases \
0      Afghanistan      36263      1269      25198      9796      106
1      Albania          4880       144       2745      1991      117
2      Algeria          27973      1163      18837      7973      616
3      Andorra           907        52        803        52       10
4      Angola            950        41        242       667       18
..      ...
182  West Bank and Gaza    10621        78      3752      6791      152
183  Western Sahara        10         1         8         1         0
184      Yemen            1691       483        833       375       10
185      Zambia            4552       140       2815      1597       71
186      Zimbabwe          2704        36        542      2126      192

New deaths New recovered Deaths / 100 Cases Recovered / 100 Cases \
0          10           18           3.50           69.49
1           6           63           2.95           56.25
2           8          749           4.16           67.34
3           0           0           5.73           88.53
4           1           0           4.32           25.47
..      ...
182        2           0           0.73           35.33
```

183	0	0	10.00	80.00
184	4	36	28.56	49.26
185	1	465	3.08	61.84
186	2	24	1.33	20.04

	Deaths / 100 Recovered	Confirmed last week	1 week change \
0	5.04	35526	737
1	5.25	4171	709
2	6.17	23691	4282
3	6.48	884	23
4	16.94	749	201
..
182	2.08	8916	1705
183	12.50	10	0
184	57.98	1619	72
185	4.97	3326	1226
186	6.64	1713	991

	1 week % increase	WHO Region
0	2.07	Eastern Mediterranean
1	17.00	Europe
2	18.07	Africa
3	2.60	Europe
4	26.84	Africa
..
182	19.12	Eastern Mediterranean
183	0.00	Africa
184	4.45	Eastern Mediterranean
185	36.86	Africa
186	57.85	Africa

[187 rows x 15 columns]>

```
[35]: df.describe().T
```

C:\Python\Lib\site-packages\pandas\core\nanops.py:1016: RuntimeWarning:
invalid value encountered in subtract

```
[35]:
```

	count	mean	std	min	25% \
Confirmed	187.0	8.813094e+04	383318.663831	10.00	1114.000
Deaths	187.0	3.497519e+03	14100.002482	0.00	18.500
Recovered	187.0	5.063148e+04	190188.189643	0.00	626.500
Active	187.0	3.400194e+04	213326.173371	0.00	141.500
New cases	187.0	1.222957e+03	5710.374790	0.00	4.000
New deaths	187.0	2.895722e+01	120.037173	0.00	0.000

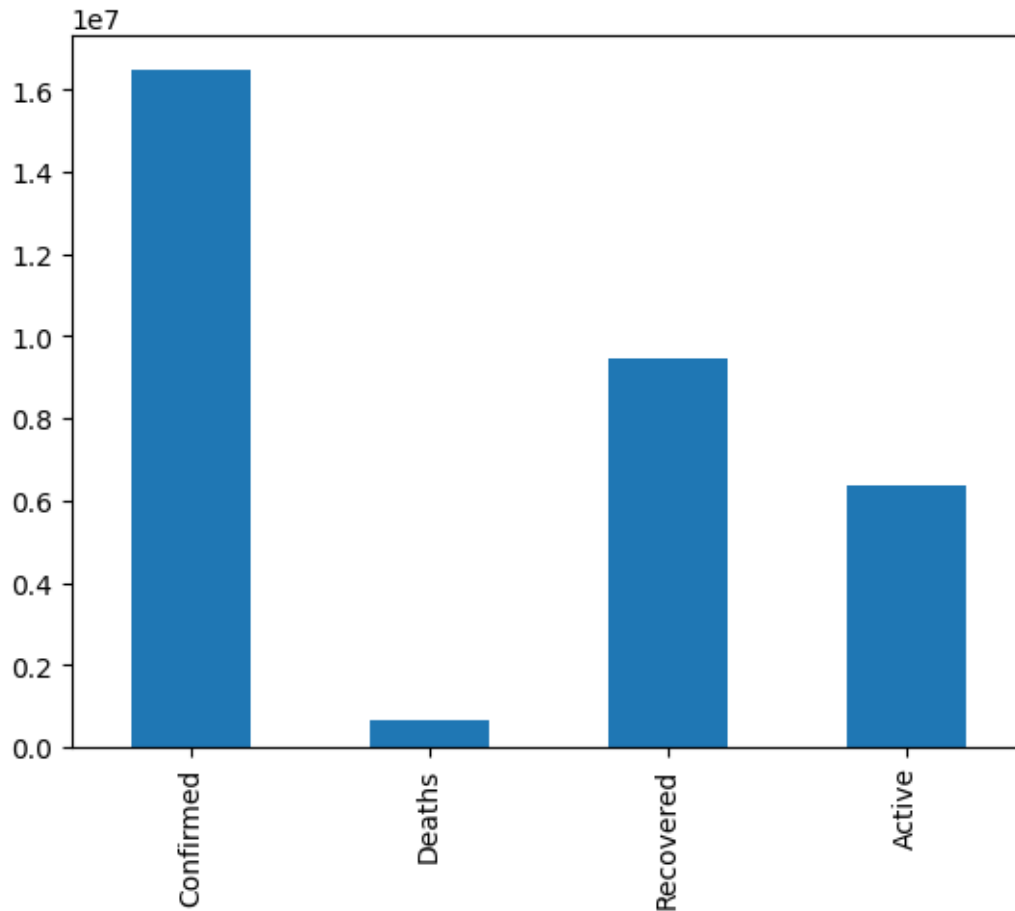
New recovered	187.0	9.338128e+02	4197.719635	0.00	0.000
Deaths / 100 Cases	187.0	3.019519e+00	3.454302	0.00	0.945
Recovered / 100 Cases	187.0	6.482053e+01	26.287694	0.00	48.770
Deaths / 100 Recovered	187.0	inf	NaN	0.00	1.450
Confirmed last week	187.0	7.868248e+04	338273.676567	10.00	1051.500
1 week change	187.0	9.448460e+03	47491.127684	-47.00	49.000
1 week % increase	187.0	1.360620e+01	24.509838	-3.84	2.775

	50%	75%	max
Confirmed	5059.00	40460.500	4290259.00
Deaths	108.00	734.000	148011.00
Recovered	2815.00	22606.000	1846641.00
Active	1600.00	9149.000	2816444.00
New cases	49.00	419.500	56336.00
New deaths	1.00	6.000	1076.00
New recovered	22.00	221.000	33728.00
Deaths / 100 Cases	2.15	3.875	28.56
Recovered / 100 Cases	71.32	86.885	100.00
Deaths / 100 Recovered	3.62	6.440	inf
Confirmed last week	5020.00	37080.500	3834677.00
1 week change	432.00	3172.000	455582.00
1 week % increase	6.89	16.855	226.32

```
[36]: column_names = ["Confirmed", "Deaths", "Recovered", "Active"]

df[column_names].sum().plot(kind="bar")
```

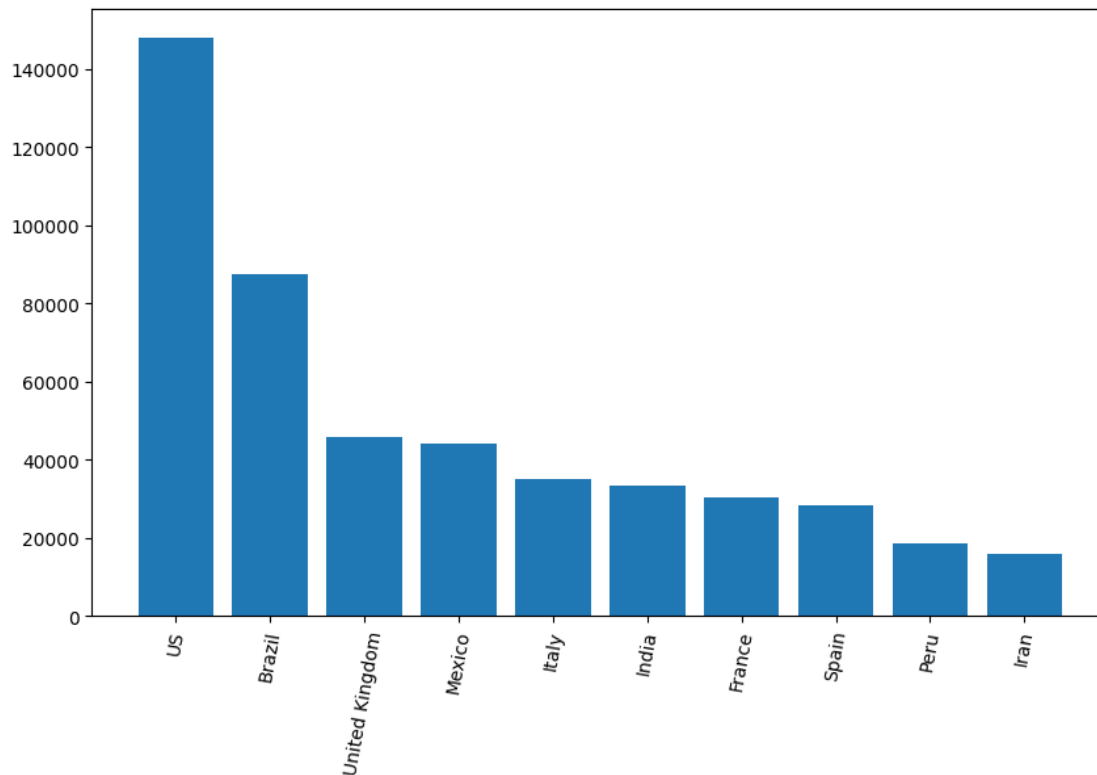
```
[36]: <Axes: >
```



```
[37]: # Top 10 countries with the highest number of death cases

country_deaths = df[['Country/Region', 'Deaths']]

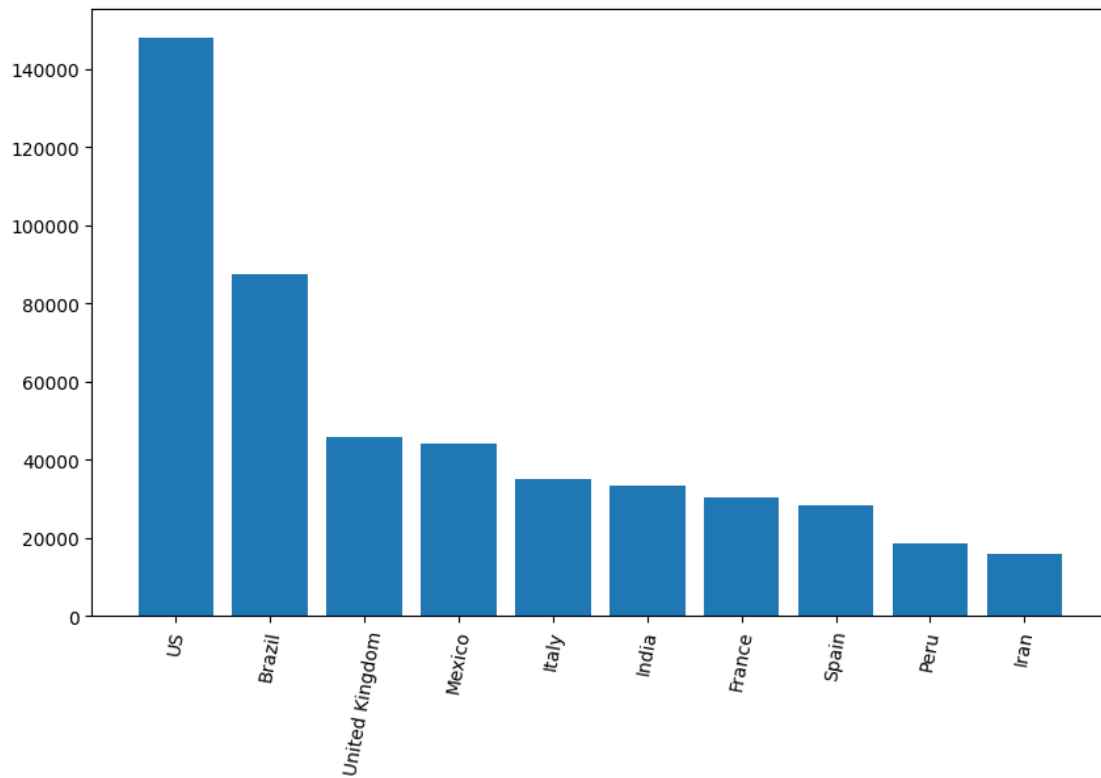
country_deaths = country_deaths.sort_values(by='Deaths', ascending=False)
plt.figure(figsize=(10, 6))
plt.bar(country_deaths['Country/Region'].head(10), country_deaths['Deaths'].
        ↪head(10))
plt.xticks(rotation=80)
plt.show()
```



```
[38]: # Top 10 countries with the highest number of confirmed cases

country_confirmed = df[['Country/Region', 'Deaths']]

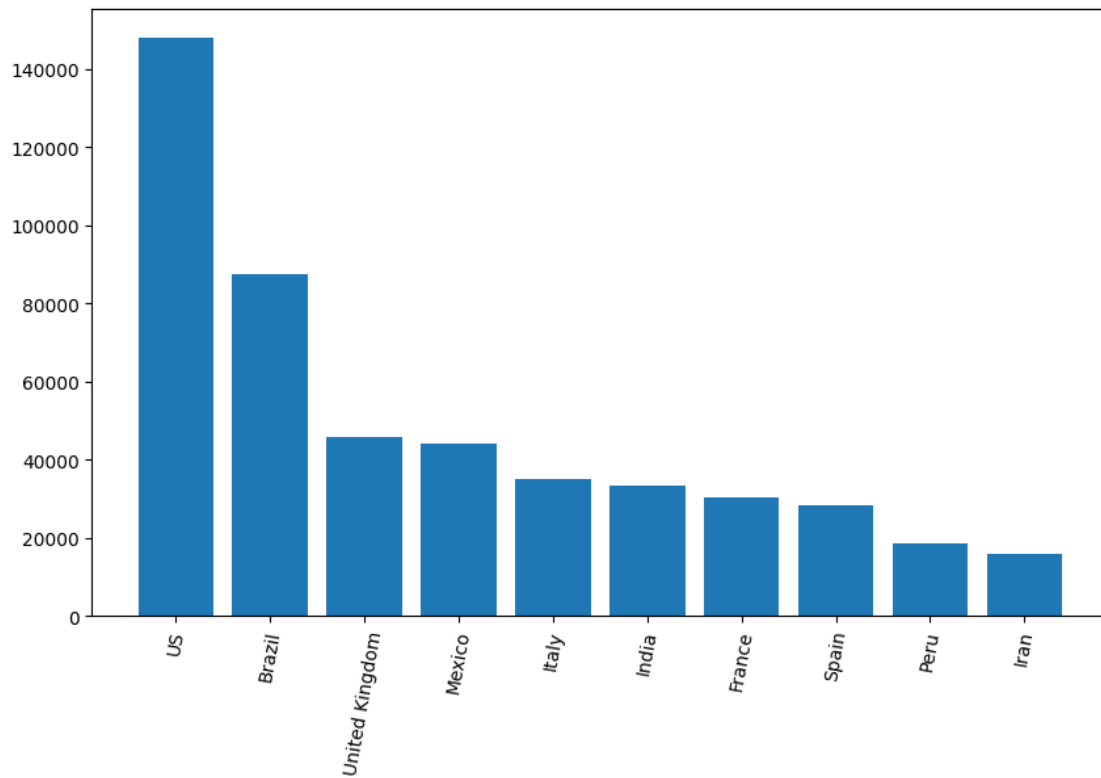
country_confirmed = country_confirmed.sort_values(by='Deaths', ascending=False)
plt.figure(figsize=(10, 6))
plt.bar(country_confirmed['Country/Region'].head(10),
        country_confirmed['Deaths'].head(10))
plt.xticks(rotation=80)
plt.show()
```

```
[39]: # Top 10 countries with the highest number of recovered cases

country_recovered = df[['Country/Region', 'Deaths']]

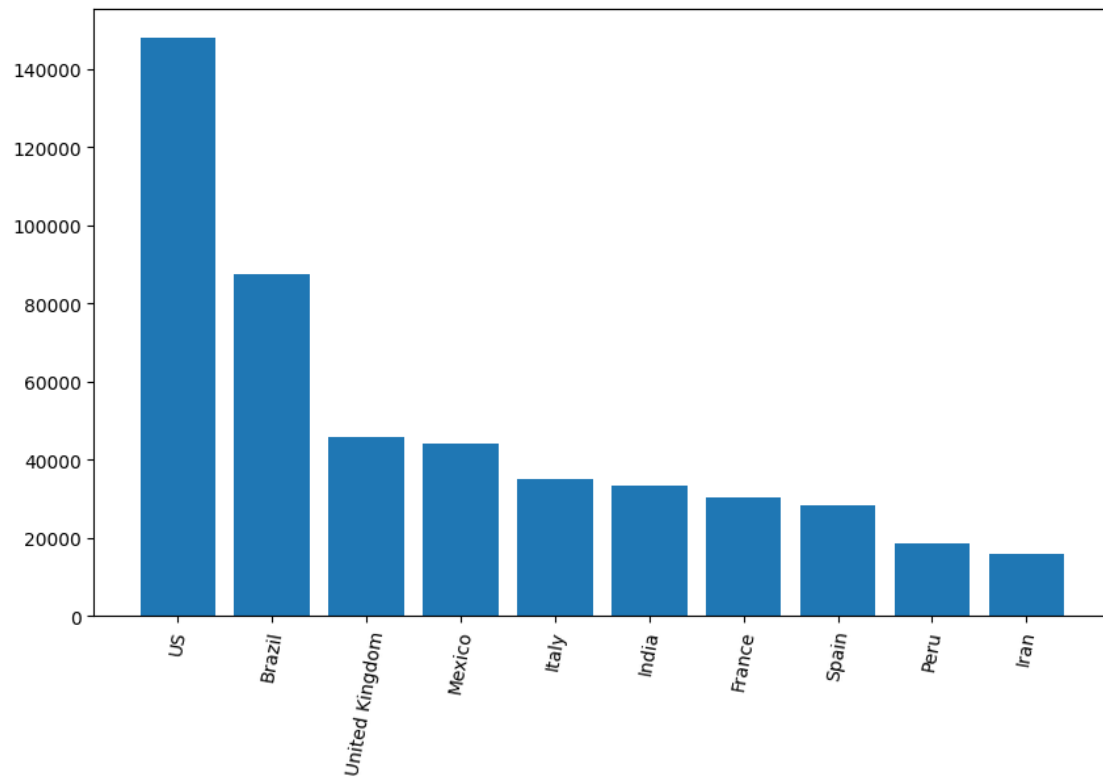
country_recovered = country_recovered.sort_values(by='Deaths', ascending=False)
plt.figure(figsize=(10, 6))
plt.bar(country_recovered['Country/Region'].head(10),
        country_recovered['Deaths'].head(10))
plt.xticks(rotation=80)
plt.show()
```



```
[40]: # Top 10 countries with the highest number of active cases

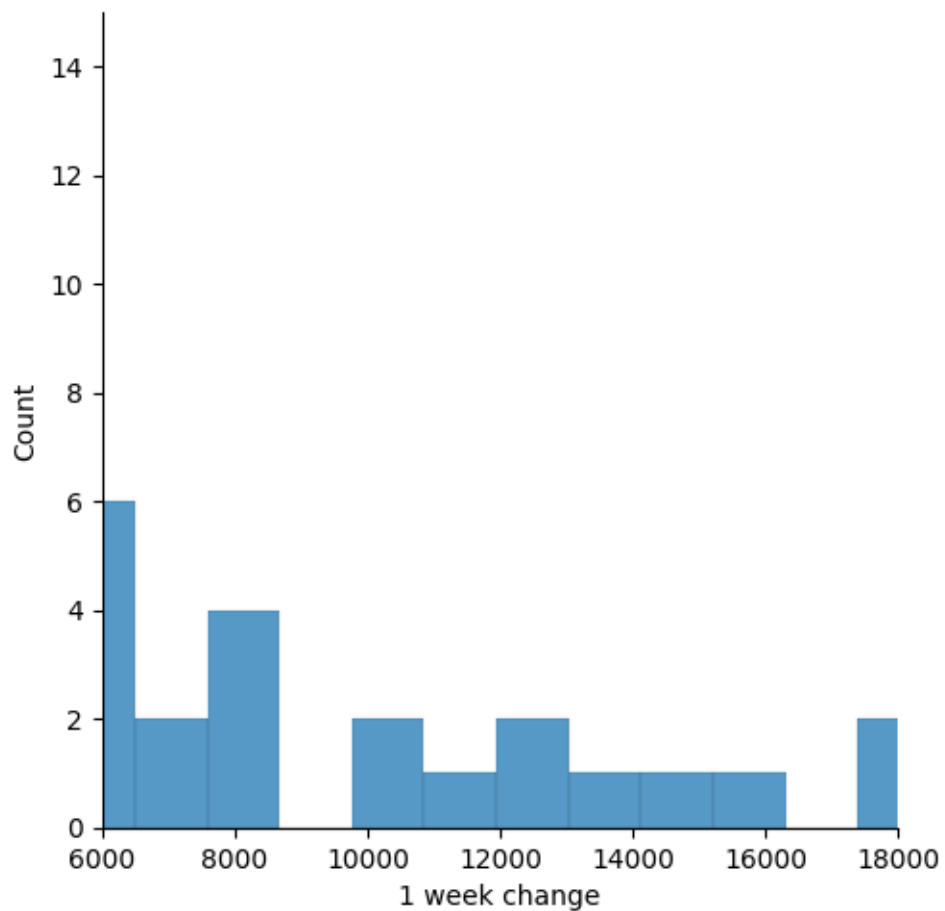
country_active = df[['Country/Region', 'Deaths']]

country_active = country_active.sort_values(by='Deaths', ascending=False)
plt.figure(figsize=(10, 6))
plt.bar(country_active['Country/Region'].head(10), country_active['Deaths'].
        ↪head(10))
plt.xticks(rotation=80)
plt.show()
```



```
[41]: # Seeing one week change  
  
sns.displot(df["1 week change"])  
plt.xlim(xmin=6000, xmax=18000)  
plt.ylim(ymin=0, ymax=15)
```

```
[41]: (0.0, 15.0)
```



```
[42]: # Import another dataset
```

```
df2 = pd.read_csv(r"C:\Users\Hitesh Karthik\OneDrive\Desktop\data science_
↳projects\Covid-19 Analysis\full_grouped.csv")
df2.head()
```

```
[42]:
```

	Date	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	\
0	2020-01-22	Afghanistan	0	0	0	0	0	
1	2020-01-22	Albania	0	0	0	0	0	
2	2020-01-22	Algeria	0	0	0	0	0	
3	2020-01-22	Andorra	0	0	0	0	0	
4	2020-01-22	Angola	0	0	0	0	0	

	New deaths	New recovered	WHO Region
0	0	0	Eastern Mediterranean
1	0	0	Europe
2	0	0	Africa
3	0	0	Europe

4	0	0	Africa
---	---	---	--------

```
[43]: df2.tail()
```

```
[43]:
```

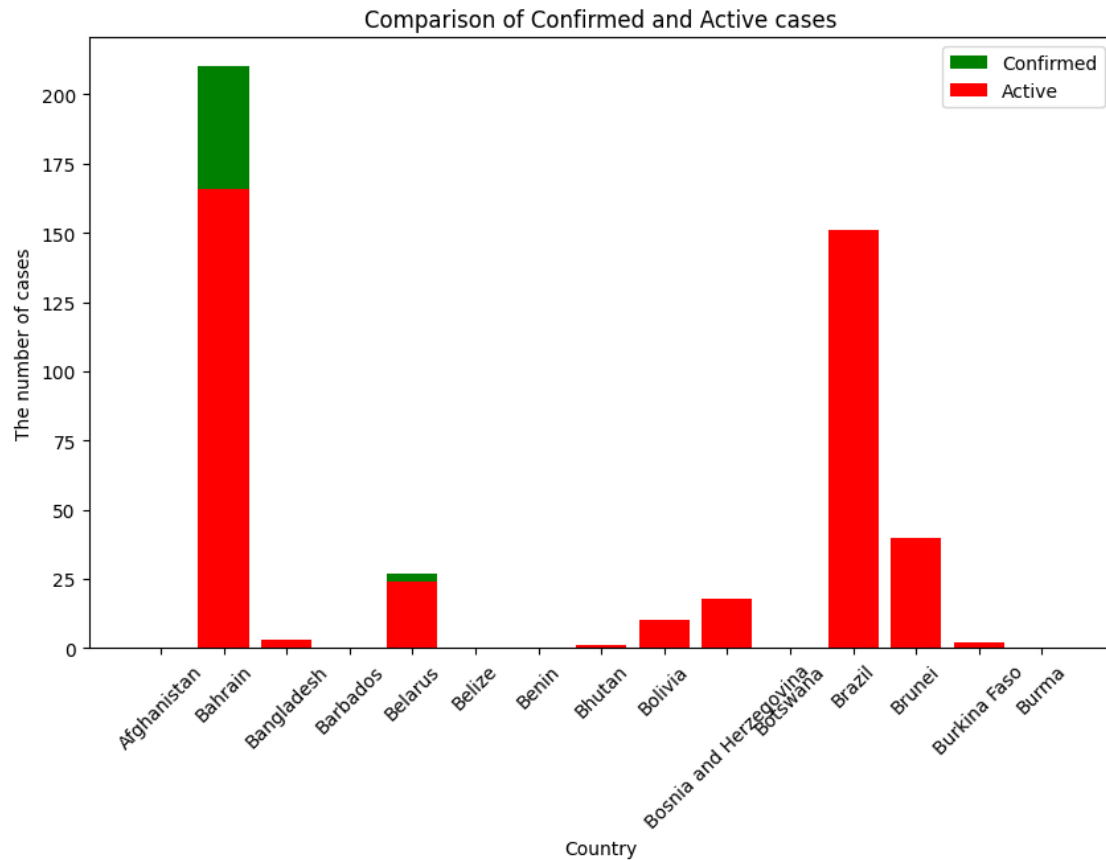
	Date	Country/Region	Confirmed	Deaths	Recovered	Active	\
35151	2020-07-27	West Bank and Gaza	10621	78	3752	6791	
35152	2020-07-27	Western Sahara	10	1	8	1	
35153	2020-07-27	Yemen	1691	483	833	375	
35154	2020-07-27	Zambia	4552	140	2815	1597	
35155	2020-07-27	Zimbabwe	2704	36	542	2126	

	New cases	New deaths	New recovered	WHO Region
35151	152	2	0	Eastern Mediterranean
35152	0	0	0	Africa
35153	10	4	36	Eastern Mediterranean
35154	71	1	465	Africa
35155	192	2	24	Africa

```
[44]: # Comparing death and recovered cases in countries with the least death cases

# Sort dataset which contains only top 15 countries
sorted_df2 = df2.sort_values(by="Deaths")
sorted_df2 = sorted_df2[:15]

# Plotting the comparison
plt.figure(figsize=(10, 6)) # Set the figure size
plt.bar(sorted_df2["Country/Region"], sorted_df2["Confirmed"],
        label='Confirmed', color='green') # Plot total recovered
plt.bar(sorted_df2["Country/Region"], sorted_df2["Active"], label='Active',
        color='red') # Plot total deaths
plt.legend() # Show legend
plt.xlabel('Country') # X-axis label
plt.ylabel('The number of cases') # Y-axis label
plt.xlim()
plt.ylim()
plt.title('Comparison of Confirmed and Active cases') # Plot title
plt.xticks(rotation=45) # Rotate x-axis labels for better visibility
plt.show() # Display the plot
```



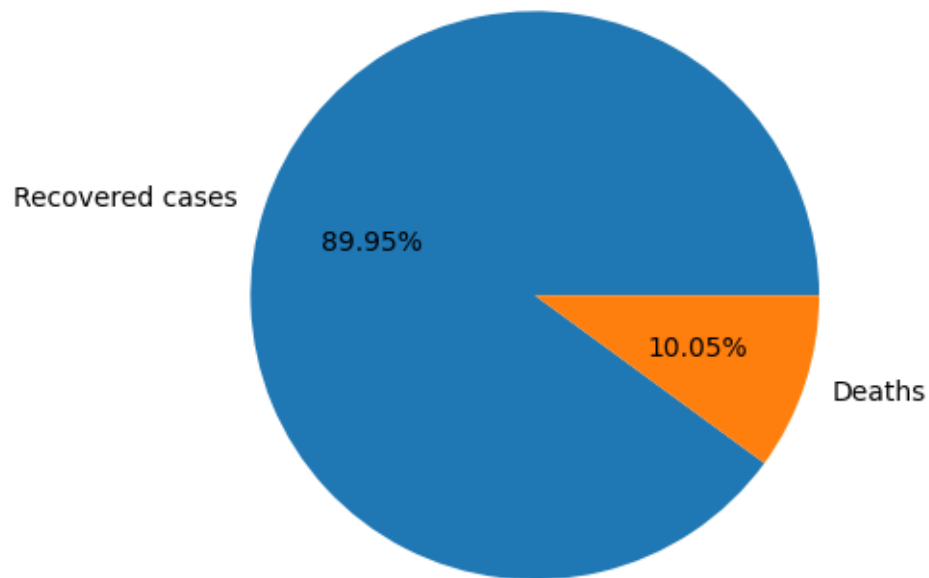
```
[45]: # Comparing recovered and death cases
```

```
recovered = df2["Recovered"].sum()
death = df2["Deaths"].sum()

data = [recovered, death]
labels = ["Recovered cases", "Deaths"]

plt.pie(data, labels=labels, autopct="%.2f%%")
```

```
[45]: ([<matplotlib.patches.Wedge at 0x2113155e810>,
<matplotlib.patches.Wedge at 0x211321f5a90>],
[Text(-1.045652528267805, 0.34148322085740523, 'Recovered cases'),
Text(1.045652520274813, -0.3414832453326693, 'Deaths')],
[Text(-0.5703559245097118, 0.1862635750131301, '89.95%'),
Text(0.570355920149898, -0.18626358836327414, '10.05%')])
```



```
[46]: # Import the worldometer dataset

df3 = pd.read_csv(r"C:\Users\Hitesh Karthik\OneDrive\Desktop\data science_
↳projects\Covid-19 Analysis\worldometer_data.csv")
df3.head()
```

```
[46]: Country/Region    Continent    Population    TotalCases    NewCases    \
0          USA    North America    3.311981e+08    5032179    NaN
1          Brazil    South America    2.127107e+08    2917562    NaN
2          India          Asia    1.381345e+09    2025409    NaN
3          Russia          Europe    1.459409e+08    871894    NaN
4    South Africa          Africa    5.938157e+07    538184    NaN

    TotalDeaths    NewDeaths    TotalRecovered    NewRecovered    ActiveCases    \
0    162804.0    NaN    2576668.0    NaN    2292707.0
1    98644.0    NaN    2047660.0    NaN    771258.0
2    41638.0    NaN    1377384.0    NaN    606387.0
3    14606.0    NaN    676357.0    NaN    180931.0
4     9604.0    NaN    387316.0    NaN    141264.0

    Serious,Critical    Tot Cases/1M pop    Deaths/1M pop    TotalTests    \
0    18296.0    15194.0    492.0    63139605.0
1     8318.0    13716.0    464.0    13206188.0
```

2	8944.0	1466.0	30.0	22149351.0
3	2300.0	5974.0	100.0	29716907.0
4	539.0	9063.0	162.0	3149807.0

	Tests/1M pop	WHO Region
0	190640.0	Americas
1	62085.0	Americas
2	16035.0	South-EastAsia
3	203623.0	Europe
4	53044.0	Africa

```
[47]: # Create a dataframe to identify ratio of Total deaths to total population
```

```
ratio = round(df3['TotalDeaths'] / df3["TotalCases"] * 100, 2)
data = {
    "Country": df3["Country/Region"],
    "Death percent": ratio
}

pd.DataFrame(data)
```

```
[47]:
```

	Country	Death percent
0	USA	3.24
1	Brazil	3.38
2	India	2.06
3	Russia	1.68
4	South Africa	1.78
..
204	Montserrat	7.69
205	Caribbean Netherlands	NaN
206	Falkland Islands	NaN
207	Vatican City	NaN
208	Western Sahara	10.00

[209 rows x 2 columns]

```
[48]: # Create pie chart to see serious/critical cases
```

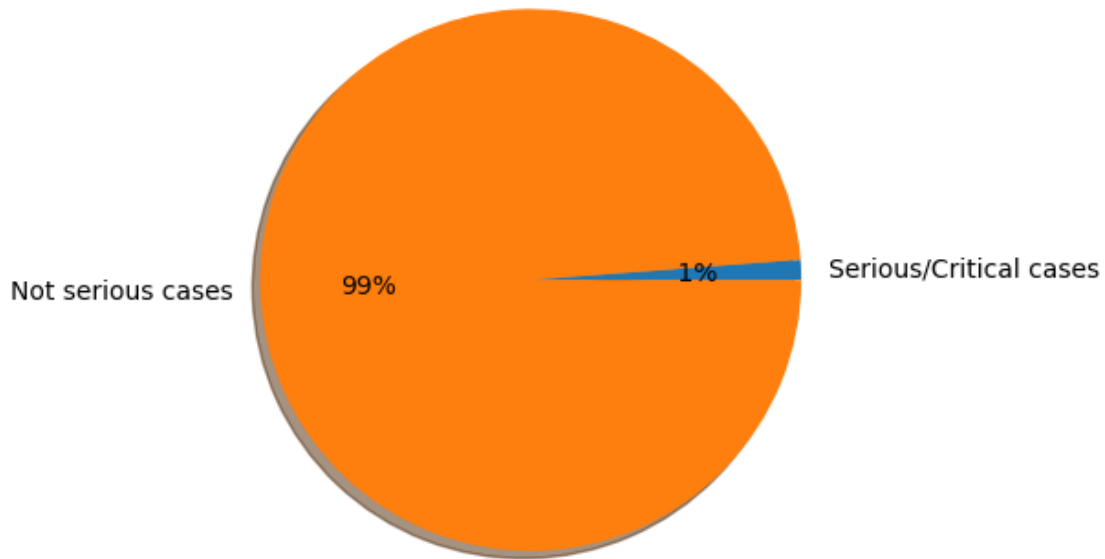
```
x = df3["Serious,Critical"].sum(), (df3["ActiveCases"] -
    df3["Serious,Critical"]).sum()
labels = ["Serious/Critical cases", "Not serious cases"]

plt.pie(x, labels=labels, autopct="%2.f%%", shadow=True)
```

```
[48]: ([<matplotlib.patches.Wedge at 0x2113155d610>,
    <matplotlib.patches.Wedge at 0x21131492550>],
    [Text(1.099223270522669, 0.04133039488617791, 'Serious/Critical cases'),
```



```
Text(-1.09922327197378, -0.04133035629241161, 'Not serious cases')],
[Text(0.5995763293760012, 0.02254385175609704, ' 1%'),
Text(-0.5995763301675162, -0.022543830704951782, '99%'))]
```



[49]: # Continents with the number of Active cases in the form of pie chart

```
continental_active = df3[["Continent", "ActiveCases"]]
continental_active = df3.groupby("Continent").sum()

continental_active
```

[49]:

Continent	Country/Region \
Africa	South AfricaEgyptNigeriaGhanaAlgeriaMoroccoKen...
Asia	IndiaIranSaudi ArabiaPakistanBangladeshTurkeyI...
Australia/Oceania	AustraliaNew ZealandPapua New GuineaFrench Pol...
Europe	RussiaSpainUKItalyGermanyFranceSwedenUkraineBe...
North America	USAMexicoCanadaDominican RepublicPanamaGuatema...
South America	BrazilPeruChileColombiaArgentinaEcuadorBolivia...

Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeaths \
Africa	1.343515e+09	1011867	0.0	22114.0	0.0
Asia	3.173656e+09	4689794	20.0	100627.0	1.0

Australia/Oceania	4.095791e+07	21735	0.0	281.0	0.0
Europe	7.476775e+08	2982576	0.0	205232.0	0.0
North America	5.895035e+08	5919209	6620.0	229855.0	819.0
South America	4.311105e+08	4543273	1282.0	154885.0	80.0

	TotalRecovered	NewRecovered	ActiveCases	\
Continent				
Africa	693620.0	0.0	296133.0	
Asia	3508170.0	42.0	1080997.0	
Australia/Oceania	12620.0	0.0	8834.0	
Europe	1587302.0	0.0	475261.0	
North America	3151678.0	4140.0	2537676.0	
South America	3116150.0	936.0	1272238.0	

	Serious,Critical	Tot Cases/1M pop	Deaths/1M pop	\
Continent				
Africa	1187.0	64456.0	1003.28	
Asia	18749.0	192429.0	1846.80	
Australia/Oceania	52.0	1446.0	15.30	
Europe	5200.0	209454.0	9673.00	
North America	25709.0	88547.0	3097.00	
South America	14295.0	108441.0	2818.00	

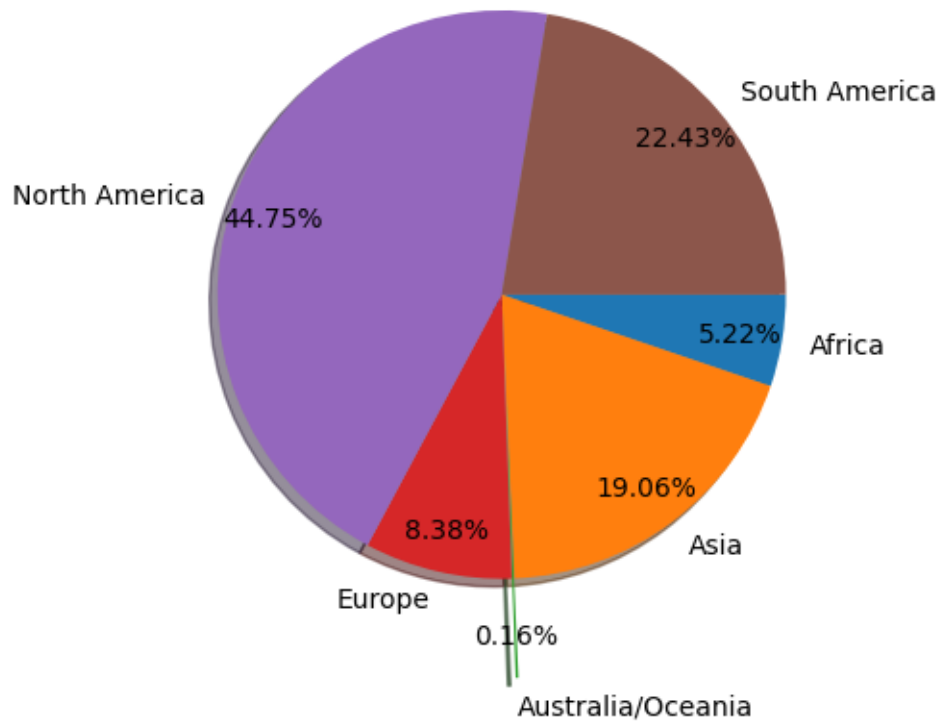
	TotalTests	Tests/1M pop	\
Continent			
Africa	8673853.0	806042.0	
Asia	65353821.0	3433453.0	
Australia/Oceania	5152811.0	347083.0	
Europe	96125611.0	8286140.0	
North America	70173584.0	2069875.0	
South America	22379618.0	1093646.0	

	WHO Region
Continent	
Africa	AfricaEasternMediterraneanAfricaAfricaAfricaEa...
Asia	South-EastAsiaEasternMediterraneanEasternMedit...
Australia/Oceania	WesternPacificWesternPacificWesternPacificWest...
Europe	EuropeEuropeEuropeEuropeEuropeEuropeEuropeEuro...
North America	AmericasAmericasAmericasAmericasAmericasAmeric...
South America	AmericasAmericasAmericasAmericasAmericasAmeric...

```
[50]: # Create a piechart

plt.pie(continental_active["ActiveCases"], labels=continental_active.index,
        autopct="%.2f%%",
        pctdistance=0.85, explode=[0, 0, 0.35, 0, 0, 0], shadow=True,
        counterclock=False)
```

```
[50]: ([<matplotlib.patches.Wedge at 0x21131507890>,
      <matplotlib.patches.Wedge at 0x2113150e650>,
      <matplotlib.patches.Wedge at 0x211314bb750>,
      <matplotlib.patches.Wedge at 0x21131512b50>,
      <matplotlib.patches.Wedge at 0x211315124d0>,
      <matplotlib.patches.Wedge at 0x2113139b550>],
      [Text(1.0852320122910852, -0.17964264387567336, 'Africa'),
       Text(0.6603272692179618, -0.8797544529737542, 'Asia'),
       Text(0.058199922031419486, -1.4488315185264078, 'Australia/Oceania'),
       Text(-0.24865555558549277, -1.0715271413624903, 'Europe'),
       Text(-1.041966751651943, 0.3525695512262767, 'North America'),
       Text(0.8379359495388681, 0.7126453146344229, 'South America')],
      [Text(0.8385883731340203, -0.13881477026756575, '5.22%'),
       Text(0.5102528898502432, -0.6798102591160827, '19.06%'),
       Text(0.048165452715657496, -1.1990329808494409, '0.16%'),
       Text(-0.19214292931606258, -0.8279982455982878, '8.38%'),
       Text(-0.8051561262765013, 0.2724401077657592, '44.75%'),
       Text(0.6474959610073071, 0.5506804703993268, '22.43%')])
```

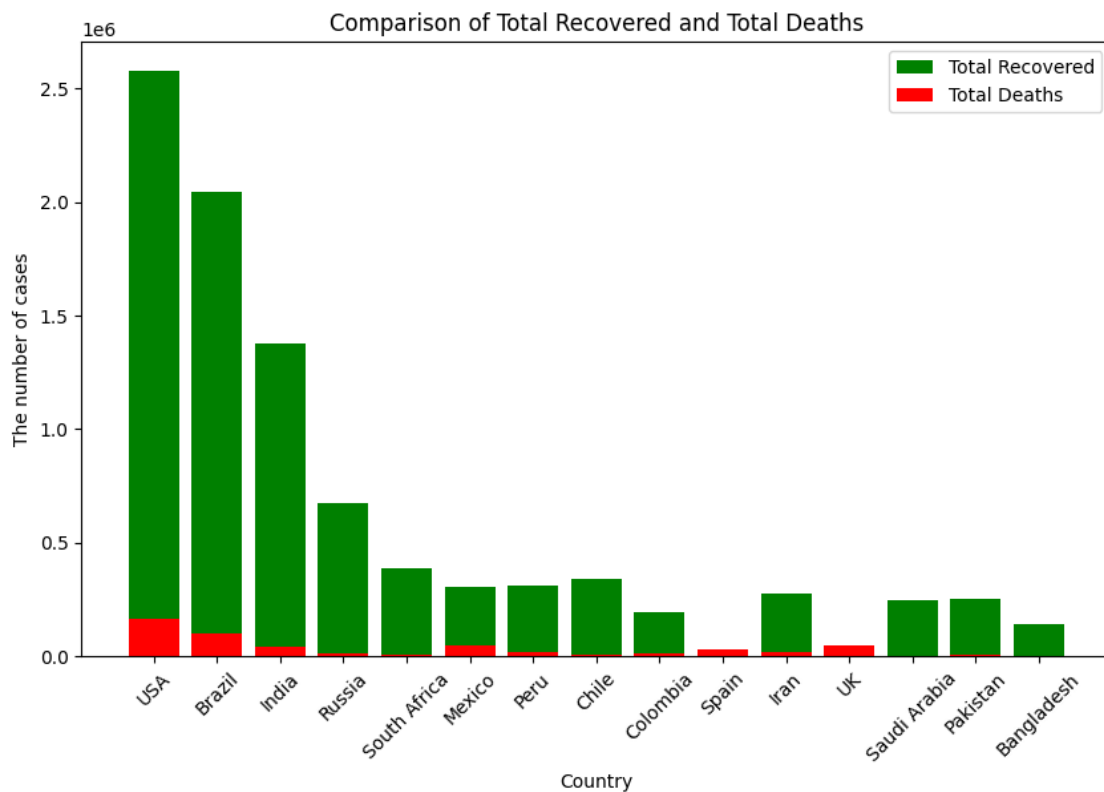


```
[51]: # Comparing total death and total recovered
```

```

# Plotting the comparison
plt.figure(figsize=(10, 6)) # Set the figure size
plt.bar(df3["Country/Region"][:15], df3["TotalRecovered"][:15], label='Total_
Recovered', color='green') # Plot total recovered
plt.bar(df3["Country/Region"][:15], df3["TotalDeaths"][:15], label='Total_
Deaths', color='red') # Plot total deaths
plt.legend() # Show legend
plt.xlabel('Country') # X-axis label
plt.ylabel('The number of cases') # Y-axis label
plt.title('Comparison of Total Recovered and Total Deaths') # Plot title
plt.xticks(rotation=45) # Rotate x-axis labels for better visibility
plt.show() # Display the plot

```



```

[52]: # Import another dataset

df4 = pd.read_csv(r"C:\Users\Hitesh Karthik\OneDrive\Desktop\data science_
projects\Covid-19 Analysis\country_wise_latest.csv")
df4.head()

```

```

[52]: Country/Region  Confirmed  Deaths  Recovered  Active  New cases  New deaths  \
0    Afghanistan    36263    1269    25198    9796    106    10

```

1	Albania	4880	144	2745	1991	117	6
2	Algeria	27973	1163	18837	7973	616	8
3	Andorra	907	52	803	52	10	0
4	Angola	950	41	242	667	18	1

	New recovered	Deaths / 100 Cases	Recovered / 100 Cases \
0	18	3.50	69.49
1	63	2.95	56.25
2	749	4.16	67.34
3	0	5.73	88.53
4	0	4.32	25.47

	Deaths / 100 Recovered	Confirmed last week	1 week change \
0	5.04	35526	737
1	5.25	4171	709
2	6.17	23691	4282
3	6.48	884	23
4	16.94	749	201

	1 week % increase	WHO Region
0	2.07	Eastern Mediterranean
1	17.00	Europe
2	18.07	Africa
3	2.60	Europe
4	26.84	Africa

```
[53]: # Death cases by countries

import pandas as pd
import plotly.express as px

# Example data
data = {
    "Country/Region": df4["Country/Region"],
    "NewDeaths": df4["New deaths"]
}

# Convert data to a DataFrame
df = pd.DataFrame(data)

# Read the shapefile for world countries
world = px.data.gapminder()

# Merge the shapefile with the data
merged = world.merge(df, left_on='country', right_on='Country/Region',
    ↪how='left')
```

```

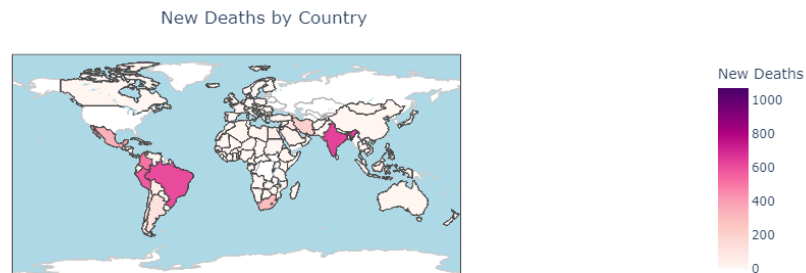
# Plotting the map
fig = px.choropleth(merged, locations='iso_alpha', color='NewDeaths',
                    color_continuous_scale="RdPu", range_color=(0,
↳df['NewDeaths'].max()),
                    labels={'NewDeaths': 'New Deaths'}, hover_name='country')

fig.update_geos(showcountries=True, countrycolor="darkgrey",
↳showcoastlines=True, coastlinecolor="lightgrey",
                    showland=True, landcolor="white", showocean=True,
↳oceancolor="lightblue", showlakes=True,
                    lakecolor="lightblue")

fig.update_layout(title_text='New Deaths by Country', title_x=0.5)

# Show the plot
fig.show()

```



```

[54]: # Active cases by countries

import pandas as pd
import plotly.express as px

# Example data
data = {
    "Country/Region": df4["Country/Region"],
    "Active": df4["Active"]
}

# Convert data to a DataFrame
df = pd.DataFrame(data)

# Read the shapefile for world countries
world = px.data.gapminder()

```

```

# Merge the shapefile with the data
merged = world.merge(df, left_on='country', right_on='Country/Region',
                    ↪how='left')

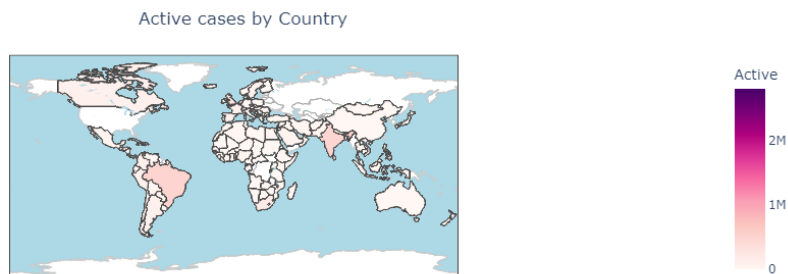
# Plotting the map
fig = px.choropleth(merged, locations='iso_alpha', color='Active',
                    color_continuous_scale="RdPu", range_color=(0, df['Active'].
                    ↪max()),
                    labels={'Active': 'Active'}, hover_name='country')

fig.update_geos(showcountries=True, countrycolor="darkgrey",
                ↪showcoastlines=True, coastlinecolor="lightgrey",
                    showland=True, landcolor="white", showocean=True,
                ↪oceancolor="lightblue", showlakes=True,
                    lakecolor="lightblue")

fig.update_layout(title_text='Active cases by Country', title_x=0.5)

# Show the plot
fig.show()

```



```

[55]: # One week change by countries

import pandas as pd
import plotly.express as px

# Example data
data = {
    "Country/Region": df4["Country/Region"],
    "1 week change": df4["1 week change"]
}

```

```

# Convert data to a DataFrame
df = pd.DataFrame(data)

# Read the shapefile for world countries
world = px.data.gapminder()

# Merge the shapefile with the data
merged = world.merge(df, left_on='country', right_on='Country/Region',
    ↪how='left')

# Plotting the map
fig = px.choropleth(merged, locations='iso_alpha', color='1 week change',
    color_continuous_scale="RdPu", range_color=(0, df['1 week_
    ↪change'].max()),
    labels={'1 week change': '1 week change'},
    ↪hover_name='country')

fig.update_geos(showcountries=True, countrycolor="darkgrey",
    ↪showcoastlines=True, coastlinecolor="lightgrey",
    showland=True, landcolor="white", showocean=True,
    ↪oceancolor="lightblue", showlakes=True,
    lakecolor="lightblue")

fig.update_layout(title_text='One week change by Country', title_x=0.5)

# Show the plot
fig.show()

```

