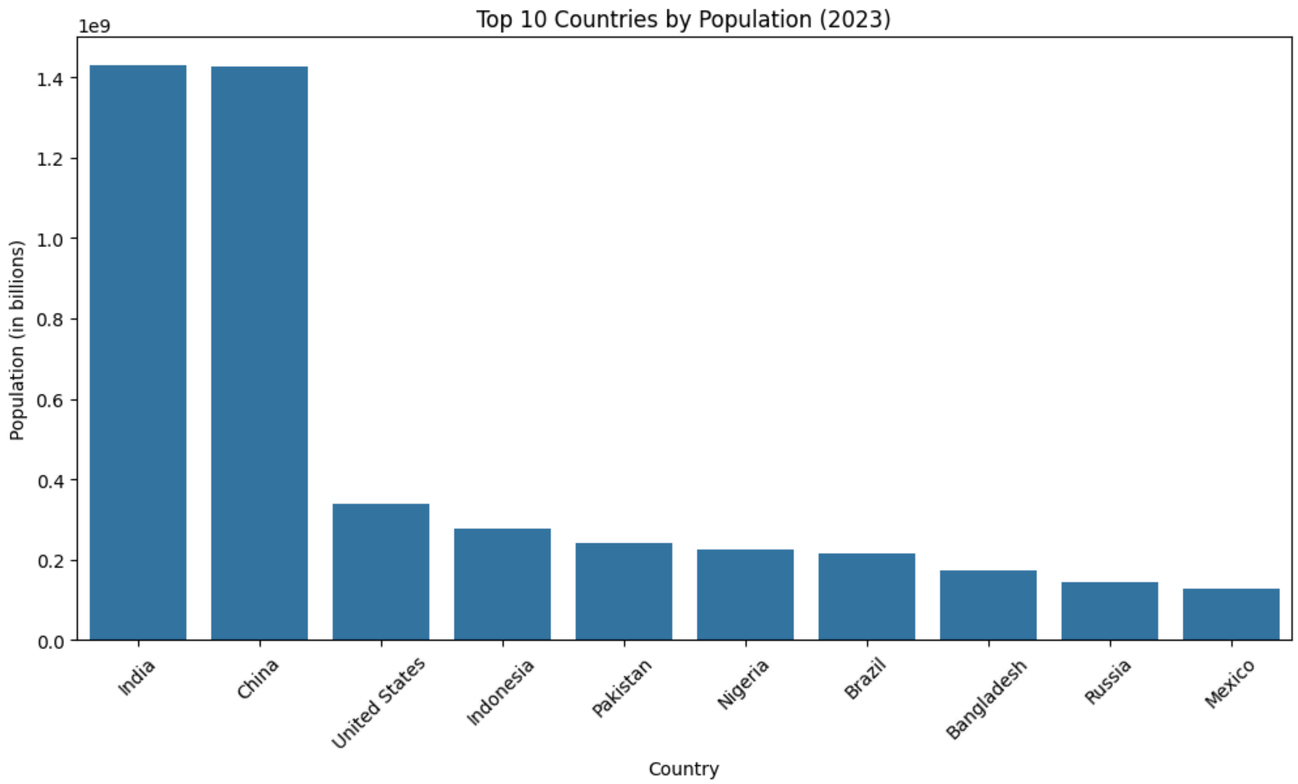
```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import seaborn as sns
```

```
data = pd.read_csv("transformed_data.csv")
data2 = pd.read_csv("raw_data.csv")
data3 = pd.read_csv("countries-table.csv")
data4 = pd.read_csv("owid-covid-data.csv")
```

```
pd.plotting.register_matplotlib_converters()
%matplotlib inline
data3 = data3.dropna()
data3.head()
```

	country	rank	area	landAreaKm	cca2	cca3	netChange	growthRate	worldPercentage	density	densityMi	place	pop198
0	India	1	3287590.0	2973190.0	IN	IND	0.4184	0.0081	0.1785	480.5033	1244.5036	356	69682838
1	China	2	9706961.0	9424702.9	CN	CHN	-0.0113	-0.0002	0.1781	151.2696	391.7884	156	98237246
2	United States	3	9372610.0	9147420.0	US	USA	0.0581	0.0050	0.0425	37.1686	96.2666	840	22314001
3	Indonesia	4	1904569.0	1877519.0	ID	IDN	0.0727	0.0074	0.0347	147.8196	382.8528	360	14817709

```
top_10_countries = data3.nlargest(10, 'pop2023')
plt.figure(figsize=(12, 6))
sns.barplot(x='country', y='pop2023', data=top_10_countries)
plt.title('Top 10 Countries by Population (2023)')
plt.xlabel('Country')
plt.ylabel('Population (in billions)')
plt.xticks(rotation=45)
plt.show()
```



```
data4.head()
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoot
0	AFG	Asia	Afghanistan	2020-02-24	5.0	5.0	NaN	NaN	NaN	N
1	AFG	Asia	Afghanistan	2020-02-25	5.0	0.0	NaN	NaN	NaN	N
2	AFG	Asia	Afghanistan	2020-02-26	5.0	0.0	NaN	NaN	NaN	N
3	AFG	Asia	Afghanistan	2020-02-27	5.0	0.0	NaN	NaN	NaN	N
4	AFG	Asia	Afghanistan	2020-02-28	5.0	0.0	NaN	NaN	NaN	N

5 rows × 67 columns

```
data4.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 107104 entries, 0 to 107103
Data columns (total 67 columns):
 #   Column                                   Non-Null Count   Dtype  
---  ------                                   --------------   -----  
 0   iso_code                                 107104 non-null  object 
 1   continent                                100984 non-null  object 
 2   location                                 107104 non-null  object 
 3   date                                     107103 non-null  object 
 4   total_cases                              105696 non-null  float64
 5   new_cases                                105586 non-null  float64
 6   new_cases_smoothed                       104292 non-null  float64
 7   total_deaths                             95345 non-null   float64
 8   new_deaths                               95344 non-null   float64
 9   new_deaths_smoothed                      93946 non-null   float64
 10  total_cases_per_million                  104938 non-null  float64
 11  new_cases_per_million                    104828 non-null  float64
 12  new_cases_smoothed_per_million           103540 non-null  float64
 13  total_deaths_per_million                 94600 non-null   float64
 14  new_deaths_per_million                   94599 non-null   float64
 15  new_deaths_smoothed_per_million          93207 non-null   float64
 16  reproduction_rate                        80413 non-null   float64
 17  icu_patients                             15074 non-null   float64
 18  icu_patients_per_million                 15074 non-null   float64
 19  hosp_patients                            15640 non-null   float64
 20  hosp_patients_per_million                15640 non-null   float64
 21  weekly_icu_admissions                    4178 non-null    float64
 22  weekly_icu_admissions_per_million        4178 non-null    float64
 23  weekly_hosp_admissions                   7413 non-null    float64
 24  weekly_hosp_admissions_per_million       7413 non-null    float64
 25  new_tests                                41668 non-null   float64
 26  total_tests                              42489 non-null   float64
 27  total_tests_per_thousand                 42489 non-null   float64
 28  new_tests_per_thousand                   41668 non-null   float64
 29  new_tests_smoothed                       53677 non-null   float64
 30  new_tests_smoothed_per_thousand          53677 non-null   float64
 31  positive_rate                            49226 non-null   float64
 32  tests_per_case                           48821 non-null   float64
 33  tests_units                              55573 non-null   object 
 34  total_vaccinations                       28794 non-null   float64
 35  people_vaccinated                        27843 non-null   float64
 36  people_fully_vaccinated                  25973 non-null   float64
 37  total_boosters                           10348 non-null   float64
 38  new_vaccinations                         23717 non-null   float64
 39  new_vaccinations_smoothed                53716 non-null   float64
 40  total_vaccinations_per_hundred           28794 non-null   float64
 41  people_vaccinated_per_hundred            27843 non-null   float64
 42  people_fully_vaccinated_per_hundred      25973 non-null   float64
 43  total_boosters_per_hundred               10348 non-null   float64
 44  new_vaccinations_smoothed_per_million    53716 non-null   float64
 45  new_people_vaccinated_smoothed           53020 non-null   float64
 46  new_people_vaccinated_smoothed_per_hundred 53020 non-null   float64
 47  stringency_index                         84724 non-null   float64
 48  population                               106345 non-null  float64
 49  population_density                       97112 non-null   float64
 50  median_age                               87646 non-null   float64
 51  aged_65_older                            87646 non-null   float64
 52  aged_70_older                            87646 non-null   float64
```

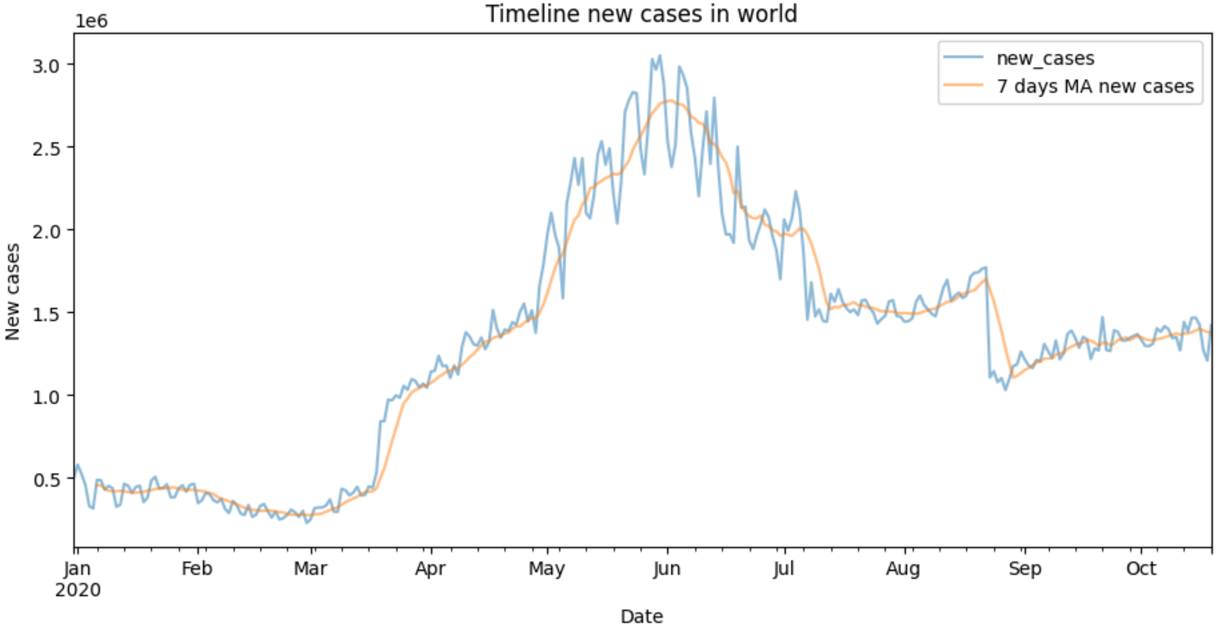
```
data4[data4['location']=='United States']
```

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed

0 rows × 67 columns

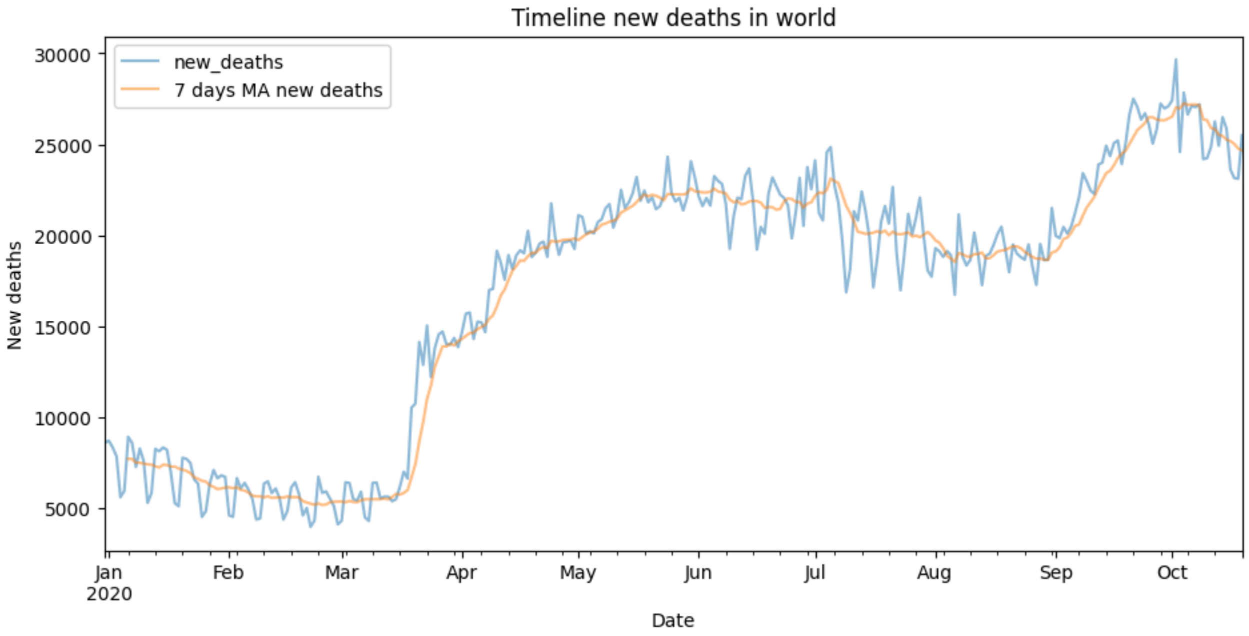
```
data5=data4.copy()
data5 = data4.copy()
data5.date = pd.to_datetime(data2['date'])
data5 = data5.groupby('date').sum()
data5['7 days MA new cases'] = 0
data5['7 days MA new cases'] = data5['new_cases'].rolling(7).mean() #Moving average of new cases with window=10
data5['7 days MA new deaths'] = 0
data5['7 days MA new deaths'] = data5['new_deaths'].rolling(7).mean()  #Moving average of new deaths with window=10

data5[['new_cases', '7 days MA new cases']].plot(figsize = (11, 5), alpha = 0.5)
plt.title('Timeline new cases in world')
plt.xlabel('Date')
plt.ylabel('New cases')
```

```
Text(0, 0.5, 'New cases')
```



```
data5[['new_deaths', '7 days MA new deaths']].plot(figsize = (11, 5), alpha = 0.5)
plt.title('Timeline new deaths in world')
plt.xlabel('Date')
plt.ylabel('New deaths')
```

```
Text(0, 0.5, 'New deaths')
```



```
print(data)
```

```
      CODE      COUNTRY        DATE    HDI        TC        TD       STI  \
0      AFG  Afghanistan  2019-12-31  0.498  0.000000  0.000000  0.000000   
1      AFG  Afghanistan  2020-01-01  0.498  0.000000  0.000000  0.000000   
2      AFG  Afghanistan  2020-01-02  0.498  0.000000  0.000000  0.000000   
3      AFG  Afghanistan  2020-01-03  0.498  0.000000  0.000000  0.000000   
4      AFG  Afghanistan  2020-01-04  0.498  0.000000  0.000000  0.000000   
...    ...          ...         ...    ...       ...       ...       ...   
50413  ZWE     Zimbabwe  2020-10-15  0.535  8.994048  5.442418  4.341855   
50414  ZWE     Zimbabwe  2020-10-16  0.535  8.996528  5.442418  4.341855   
50415  ZWE     Zimbabwe  2020-10-17  0.535  8.999496  5.442418  4.341855   
50416  ZWE     Zimbabwe  2020-10-18  0.535  9.000853  5.442418  4.341855   
50417  ZWE     Zimbabwe  2020-10-19  0.535  9.005405  5.442418  4.341855   

             POP    GDPCAP  
0      17.477233  7.497754  
1      17.477233  7.497754  
2      17.477233  7.497754  
3      17.477233  7.497754  
4      17.477233  7.497754  
...          ...       ...  
50413  16.514381  7.549491  
50414  16.514381  7.549491  
50415  16.514381  7.549491  
50416  16.514381  7.549491  
50417  16.514381  7.549491  

[50418 rows x 9 columns]
```

```
print(data.head())

       CODE      COUNTRY        DATE    HDI   TC   TD  STI        POP    GDPCAP
     0  AFG  Afghanistan  2019-12-31  0.498  0.0  0.0  0.0  17.477233  7.497754
     1  AFG  Afghanistan  2020-01-01  0.498  0.0  0.0  0.0  17.477233  7.497754
     2  AFG  Afghanistan  2020-01-02  0.498  0.0  0.0  0.0  17.477233  7.497754
     3  AFG  Afghanistan  2020-01-03  0.498  0.0  0.0  0.0  17.477233  7.497754
     4  AFG  Afghanistan  2020-01-04  0.498  0.0  0.0  0.0  17.477233  7.497754
```

```
print(data2.head())

       iso_code     location        date  total_cases  total_deaths  \
     0      AFG  Afghanistan  2019-12-31          0.0           0.0
     1      AFG  Afghanistan  2020-01-01          0.0           0.0
     2      AFG  Afghanistan  2020-01-02          0.0           0.0
     3      AFG  Afghanistan  2020-01-03          0.0           0.0
     4      AFG  Afghanistan  2020-01-04          0.0           0.0

        stringency_index  population  gdp_per_capita  human_development_index  \
     0               0.0    38928341        1803.987                    0.498
     1               0.0    38928341        1803.987                    0.498
     2               0.0    38928341        1803.987                    0.498
     3               0.0    38928341        1803.987                    0.498
     4               0.0    38928341        1803.987                    0.498

       Unnamed: 9 Unnamed: 10 Unnamed: 11  Unnamed: 12  Unnamed: 13
     0      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
     1      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
     2      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
     3      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
     4      #NUM!       #NUM!       #NUM!    17.477233  7.497754494
```

```
data["COUNTRY"].value_counts()

     COUNTRY
     Afghanistan        294
     Indonesia          294
     Macedonia          294
     Luxembourg         294
     Lithuania          294
                       ...
     Tajikistan         172
     Comoros            171
     Lesotho            158
     Hong Kong           51
     Solomon Islands      4
     Name: count, Length: 210, dtype: int64
```

```
data["COUNTRY"].value_counts().mode()

     0    294
     Name: count, dtype: int64
```

```
# Aggregating the data

code = data["CODE"].unique().tolist()
country = data["COUNTRY"].unique().tolist()
hdi = []
tc = []
td = []
sti = []
population = data["POP"].unique().tolist()
gdp = []

for i in country:
    hdi.append((data.loc[data["COUNTRY"] == i, "HDI"]).sum()/294)
    tc.append((data2.loc[data2["location"] == i, "total_cases"]).sum())
    td.append((data2.loc[data2["location"] == i, "total_deaths"]).sum())
    sti.append((data.loc[data["COUNTRY"] == i, "STI"]).sum()/294)
    population.append((data2.loc[data2["location"] == i, "population"]).sum()/294)

aggregated_data = pd.DataFrame(list(zip(code, country, hdi, tc, td, sti, population)),
                               columns = ["Country Code", "Country", "HDI",
                                          "Total Cases", "Total Deaths",
                                          "Stringency Index", "Population"])
print(aggregated_data.head())
```

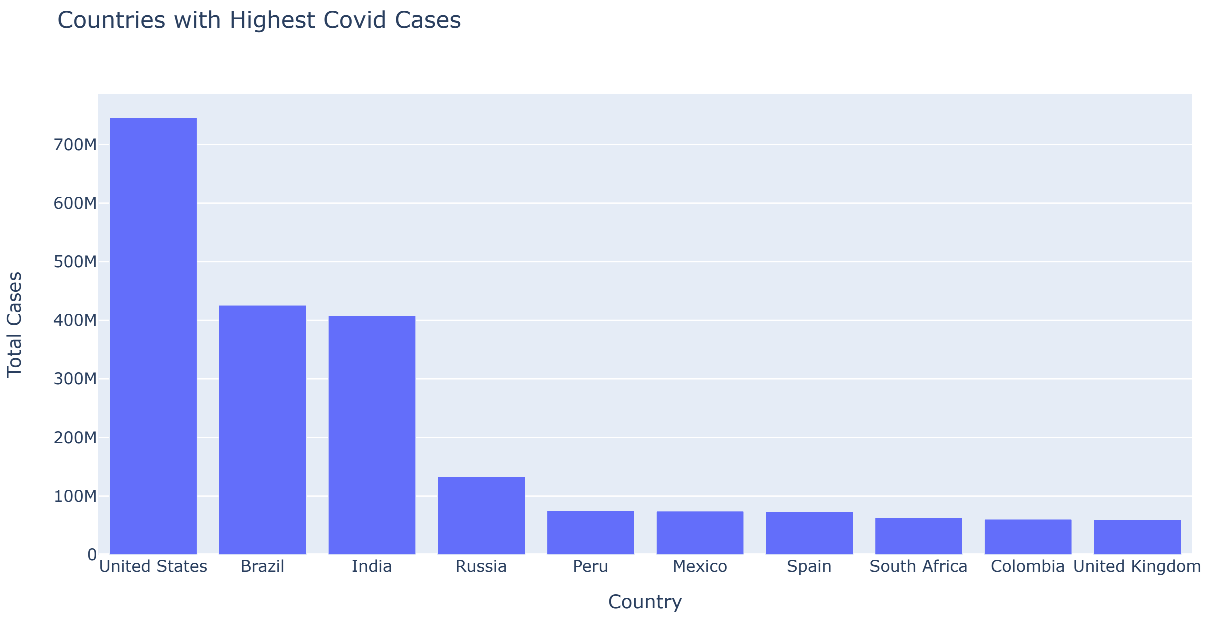
```
       Country Code      Country       HDI  Total Cases  Total Deaths  \
     0          AFG  Afghanistan  0.498000    5126433.0      165875.0
     1          ALB      Albania  0.600765    1071951.0       31056.0
     2          DZA      Algeria  0.754000    4893999.0      206429.0
     3          AND      Andorra  0.659551     223576.0        9850.0
     4          AGO       Angola  0.418952     304005.0       11820.0

        Stringency Index  Population
     0          3.049673   17.477233
     1          3.005624   14.872537
     2          3.195168   17.596309
     3          2.677654   11.254996
     4          2.965560   17.307957
```

```
# Sorting Data According to Total Cases

data = aggregated_data.sort_values(by=["Total Cases"], ascending=False)
print(data.head())
```

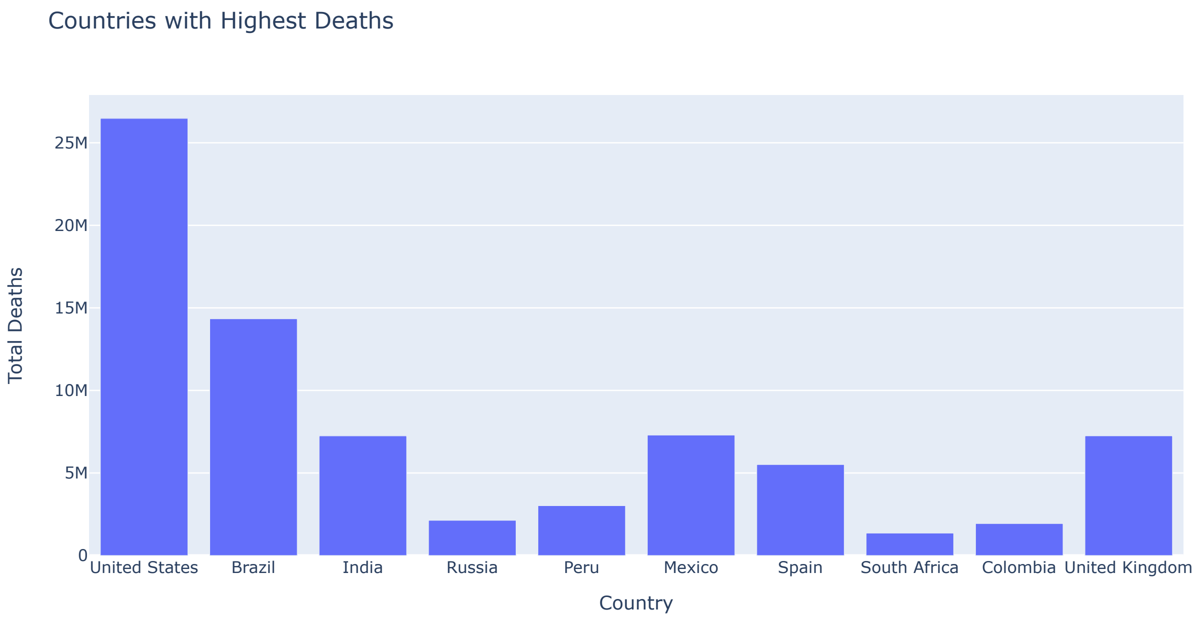
```
         Country Code        Country      HDI  Total Cases  Total Deaths  \
     200          USA  United States  0.92400  746014098.0    26477574.0
     27           BRA         Brazil  0.75900  425704517.0    14340567.0
     90           IND          India  0.64000  407771615.0     7247327.0
     157          RUS         Russia  0.81600  132888951.0     2131571.0
     150          PER           Peru  0.59949   74882695.0     3020038.0

          Stringency Index  Population
     200          3.350949   19.617637
     27           3.136028   19.174732
     90           3.610552   21.045353
     157          3.380088   18.798668
     150          3.430126   17.311165
```

```
# Top 10 Countries with Highest Covid Cases

data = data.head(10)
print(data)

    Country Code         Country       HDI  Total Cases  Total Deaths  \
200          USA   United States  0.924000  746014098.0    26477574.0
27           BRA          Brazil  0.759000  425704517.0    14340567.0
90           IND           India  0.640000  407771615.0     7247327.0
157          RUS          Russia  0.816000  132888951.0     2131571.0
150          PER            Peru  0.599490   74882695.0     3020038.0
125          MEX          Mexico  0.774000   74347548.0     7295850.0
178          ESP           Spain  0.887969   73717676.0     5510624.0
175          ZAF    South Africa  0.608653   63027659.0     1357682.0
42           COL        Colombia  0.581847   60543682.0     1936134.0
199          GBR  United Kingdom  0.922000   59475032.0     7249573.0

    Stringency Index  Population
200         3.350949   19.617637
27          3.136028   19.174732
90          3.610552   21.045353
157         3.380088   18.798668
150         3.430126   17.311165
125         3.019289   18.674802
178         3.393922   17.660427
175         3.364333   17.898266
42          3.357923   17.745037
199         3.353883   18.033340


data["GDP Before Covid"] = [65279.53, 8897.49, 2100.75, 
                            11497.65, 7027.61, 9946.03, 
                            29564.74, 6001.40, 6424.98, 42354.41]
data["GDP During Covid"] = [63543.58, 6796.84, 1900.71, 
                            10126.72, 6126.87, 8346.70, 
                            27057.16, 5090.72, 5332.77, 40284.64]
print(data)

    Country Code         Country       HDI  Total Cases  Total Deaths  \
200          USA   United States  0.924000  746014098.0    26477574.0
27           BRA          Brazil  0.759000  425704517.0    14340567.0
90           IND           India  0.640000  407771615.0     7247327.0
157          RUS          Russia  0.816000  132888951.0     2131571.0
150          PER            Peru  0.599490   74882695.0     3020038.0
125          MEX          Mexico  0.774000   74347548.0     7295850.0
178          ESP           Spain  0.887969   73717676.0     5510624.0
175          ZAF    South Africa  0.608653   63027659.0     1357682.0
42           COL        Colombia  0.581847   60543682.0     1936134.0
199          GBR  United Kingdom  0.922000   59475032.0     7249573.0

    Stringency Index  Population  GDP Before Covid  GDP During Covid
200         3.350949   19.617637          65279.53          63543.58
27          3.136028   19.174732           8897.49           6796.84
90          3.610552   21.045353           2100.75           1900.71
157         3.380088   18.798668          11497.65          10126.72
150         3.430126   17.311165           7027.61           6126.87
125         3.019289   18.674802           9946.03           8346.70
178         3.393922   17.660427          29564.74          27057.16
175         3.364333   17.898266           6001.40           5090.72
42          3.357923   17.745037           6424.98           5332.77
199         3.353883   18.033340          42354.41          40284.64
```

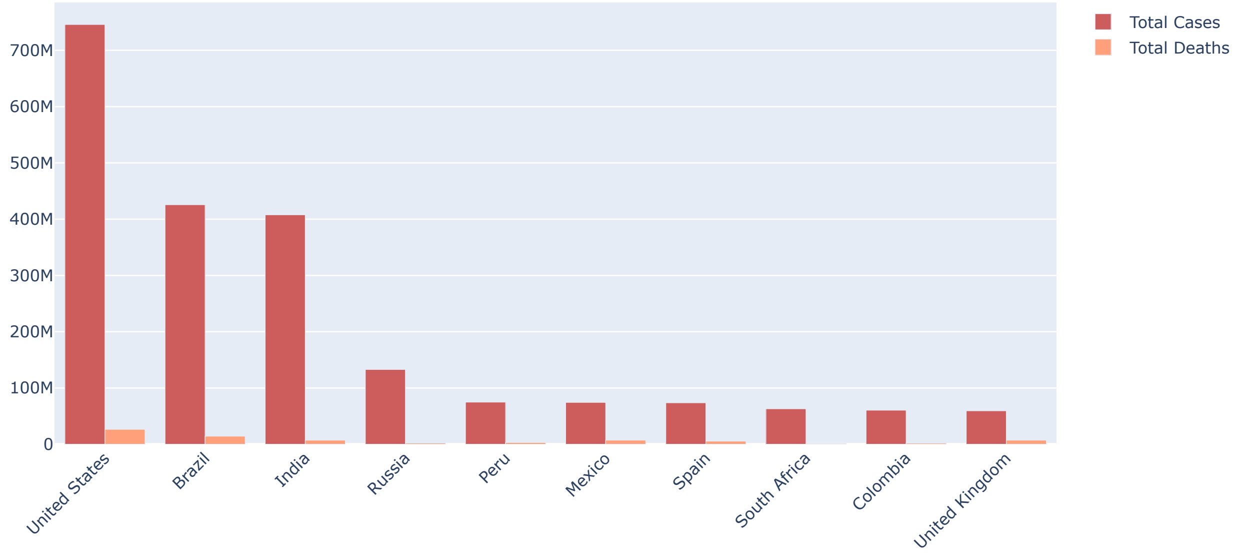
```
figure = px.bar(data, y='Total Cases', x='Country',
            title="Countries with Highest Covid Cases")
figure.show()
```



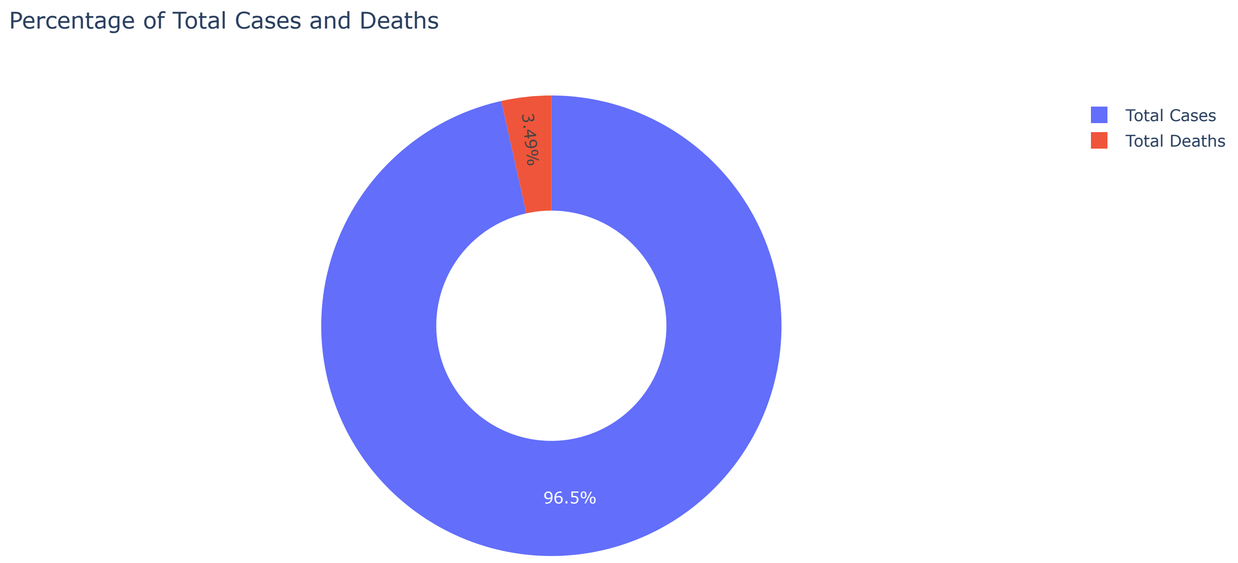
```
figure = px.bar(data, y='Total Deaths', x='Country',
            title="Countries with Highest Deaths")
figure.show()
```



```
fig = go.Figure()
fig.add_trace(go.Bar(
    x=data["Country"],
    y=data["Total Cases"],
    name='Total Cases',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=data["Country"],
    y=data["Total Deaths"],
    name='Total Deaths',
    marker_color='lightsalmon'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```



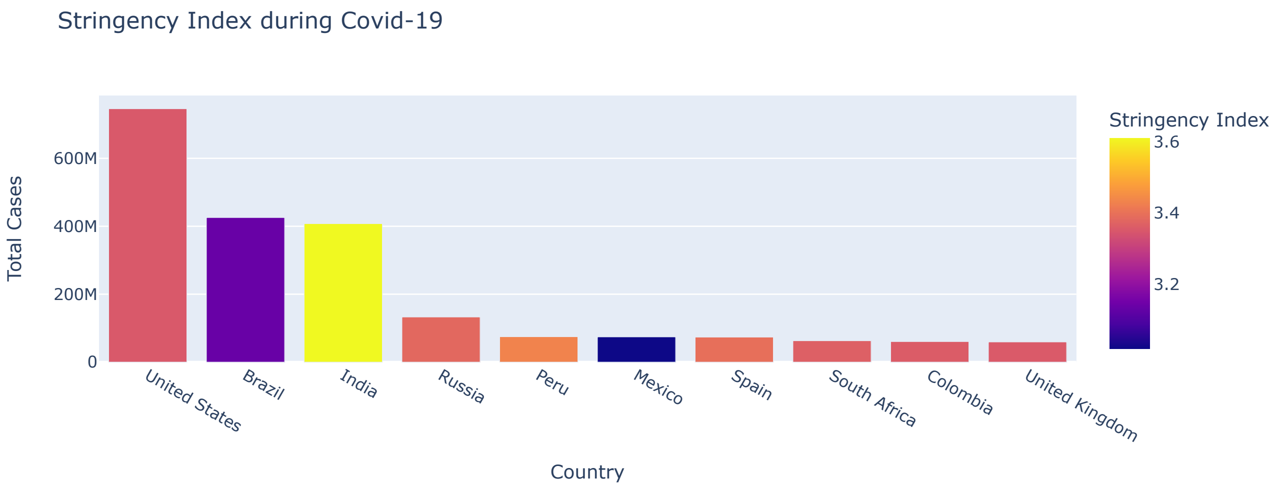
```
# Percentage of Total Cases and Deaths
cases = data["Total Cases"].sum()
deceased = data["Total Deaths"].sum()

labels = ["Total Cases", "Total Deaths"]
values = [cases, deceased]

fig = px.pie(data, values=values, names=labels,
             title='Percentage of Total Cases and Deaths', hole=0.5)
fig.show()
```



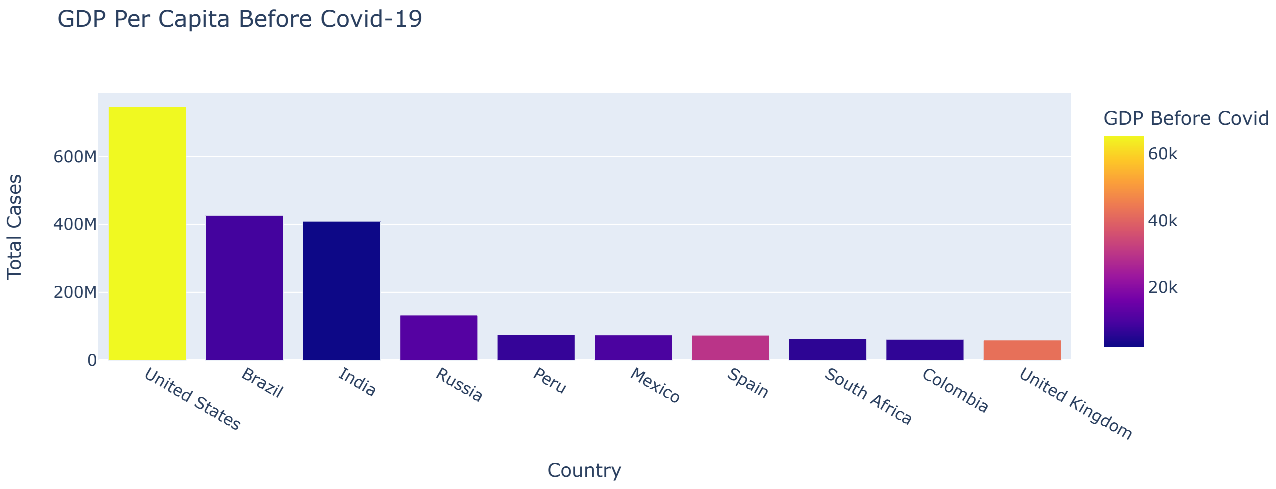
```
death_rate = (data["Total Deaths"].sum() / data["Total Cases"].sum()) * 100
print("Death Rate = ", death_rate)
```

```
     Death Rate =  3.6144212045653767
```

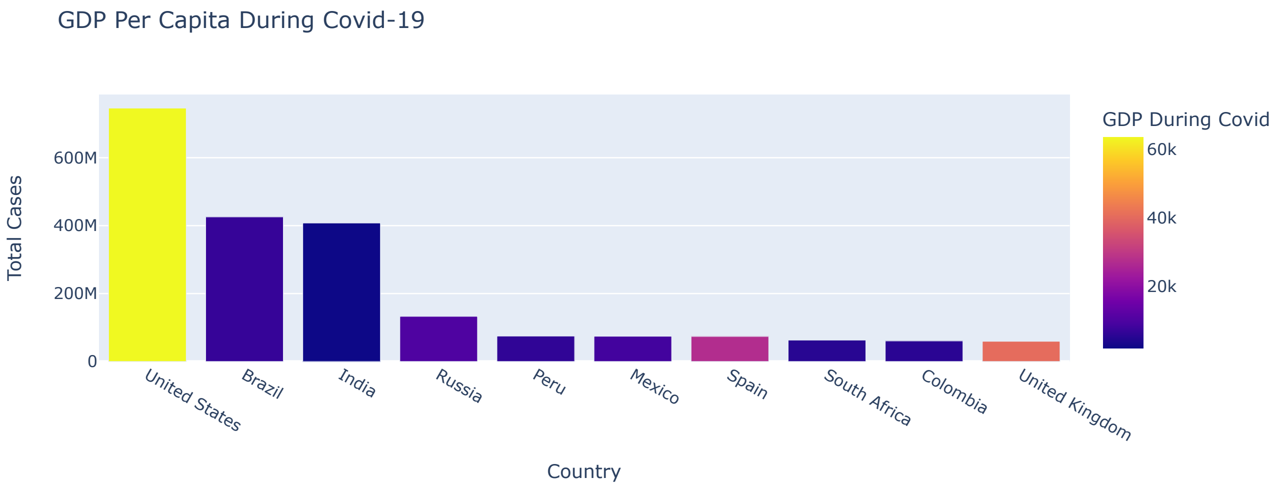
```
fig = px.bar(data, x='Country', y='Total Cases',
             hover_data=['Population', 'Total Deaths'],
             color='Stringency Index', height=400,
             title= "Stringency Index during Covid-19")
fig.show()
```



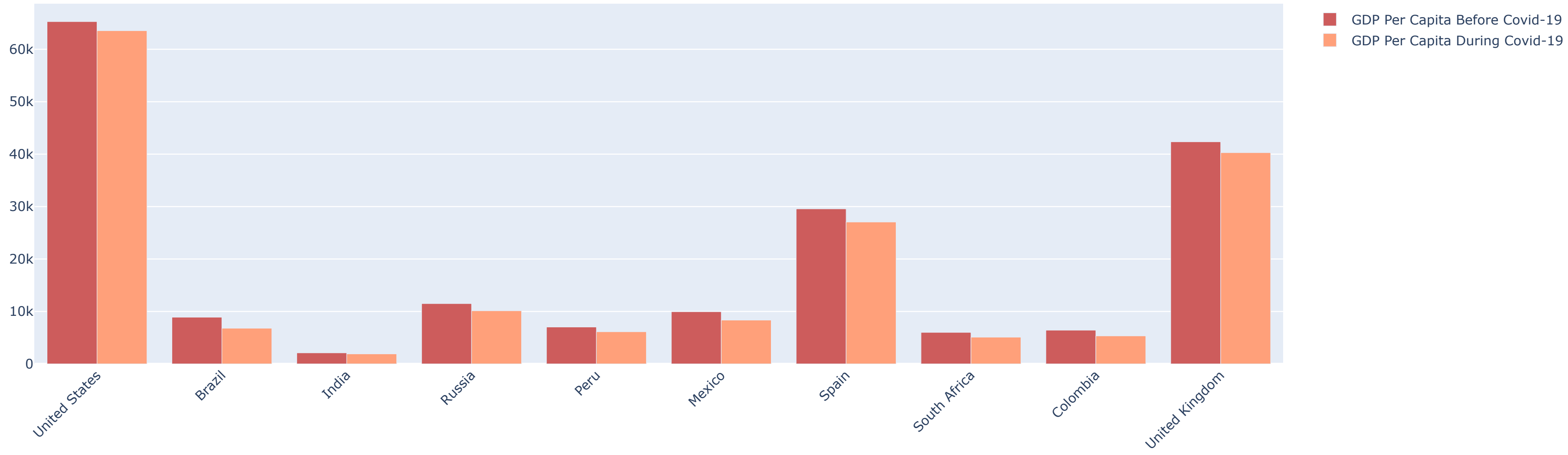
```
fig = px.bar(data, x='Country', y='Total Cases',
             hover_data=['Population', 'Total Deaths'],
             color='GDP Before Covid', height=400,
             title="GDP Per Capita Before Covid-19")
fig.show()
```



```
fig = px.bar(data, x='Country', y='Total Cases',
             hover_data=['Population', 'Total Deaths'],
             color='GDP During Covid', height=400,
             title="GDP Per Capita During Covid-19")
fig.show()
```



```
fig = go.Figure()
fig.add_trace(go.Bar(
    x=data["Country"],
    y=data["GDP Before Covid"],
    name='GDP Per Capita Before Covid-19',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=data["Country"],
    y=data["GDP During Covid"],
    name='GDP Per Capita During Covid-19',
    marker_color='lightsalmon'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```



```
fig = px.bar(data, x='Country', y='Total Cases',
             hover_data=['Population', 'Total Deaths'],
             color='HDI', height=400,
             title="Human Development Index during Covid-19")
fig.show()
```

```
import pandas as pd
import json

# Load the COVID-19 data
covid_data = pd.read_csv("owid-covid-data.csv")

# Assuming your data has columns named 'Date' and 'Cases'
dates = covid_data['Date']
cases = covid_data['Cases']

# Convert dates to a format suitable for d3.js
dates = pd.to_datetime(dates).dt.strftime('%Y-%m-%d')

# Create a JSON-like data structure for d3.js
data = [{'date': date, 'cases': case} for date, case in zip(dates, cases)]

# Write the data to a JSON file
with open('covid_data.json', 'w') as f:
    json.dump(data, f)
```