

```
In [1]: import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
```

```
In [2]: # url of datasets: https://www.kaggle.com/datasets/shashwatwork/impact-of-covid19-pandem
dataset = pd.read_csv("transformed_data.csv")
dataset1 = pd.read_csv("raw_data.csv")
print(dataset)
```

	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]

```
In [3]: # first five rows of dataset
print(dataset.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

```
In [4]: # first five rows of dataset2
print(dataset1.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

```
In [5]: # count the number of unique values in the dataset column 'COUNTRY'
dataset["COUNTRY"].value_counts()
```

```
Out[5]: Afghanistan      294
Indonesia      294
Macedonia      294
Luxembourg      294
Lithuania      294
...
Tajikistan      172
Comoros      171
Lesotho      158
Hong Kong      51
Solomon Islands      4
Name: COUNTRY, Length: 210, dtype: int64
```

```
In [6]: # the value that appears most often
dataset["COUNTRY"].value_counts().mode()
```

```
Out[6]: 0      294
Name: COUNTRY, dtype: int64
```

```
In [7]: # make one set of data
code = dataset["CODE"].unique().tolist()
country = dataset["COUNTRY"].unique().tolist()
hdi = []
tc = []
td = []
sti = []
population = dataset["POP"].unique().tolist()
gdp = []

for n in country:
    hdi.append((dataset.loc[dataset["COUNTRY"] == n, "HDI"]).sum()/294)
    tc.append((dataset1.loc[dataset1["location"] == n, "total_cases"]).sum())
    td.append((dataset1.loc[dataset1["location"] == n, "total_deaths"]).sum())
    sti.append((dataset.loc[dataset["COUNTRY"] == n, "STI"]).sum()/294)
    population.append((dataset1.loc[dataset1["location"] == n, "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

```
In [8]: # sorting by 'Total Case' in descending order
dataset = aggregated_data.sort_values(by=["Total Cases"], ascending=False)
print(dataset.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

```
In [9]: # make a data set of 10 countries with the most total cases
a = dataset.head(10)
b = dataset[dataset['Country']=='Serbia']
# add Serbia to that dataset
dataset2=pd.concat([a,b],axis=0)
print(dataset2)
```

	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	
166	SRB	Serbia	0.634418	3984174.0	87503.0	

	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
166	3.073841	15.733109

```
In [10]: # add columns: GDP per capita before and during covid
dataset2["GDP Before Covid"] = [65279.53, 8897.49, 2100.75,
                                11497.65, 7027.61, 9946.03,
                                29564.74, 6001.40, 6424.98, 42354.41, 6562.67]
dataset2["GDP During Covid"] = [63543.58, 6796.84, 1900.71,
                                10126.72, 6126.87, 8346.70,
                                27057.16, 5090.72, 5332.77, 40284.64, 6533.19]
print(dataset2)
```

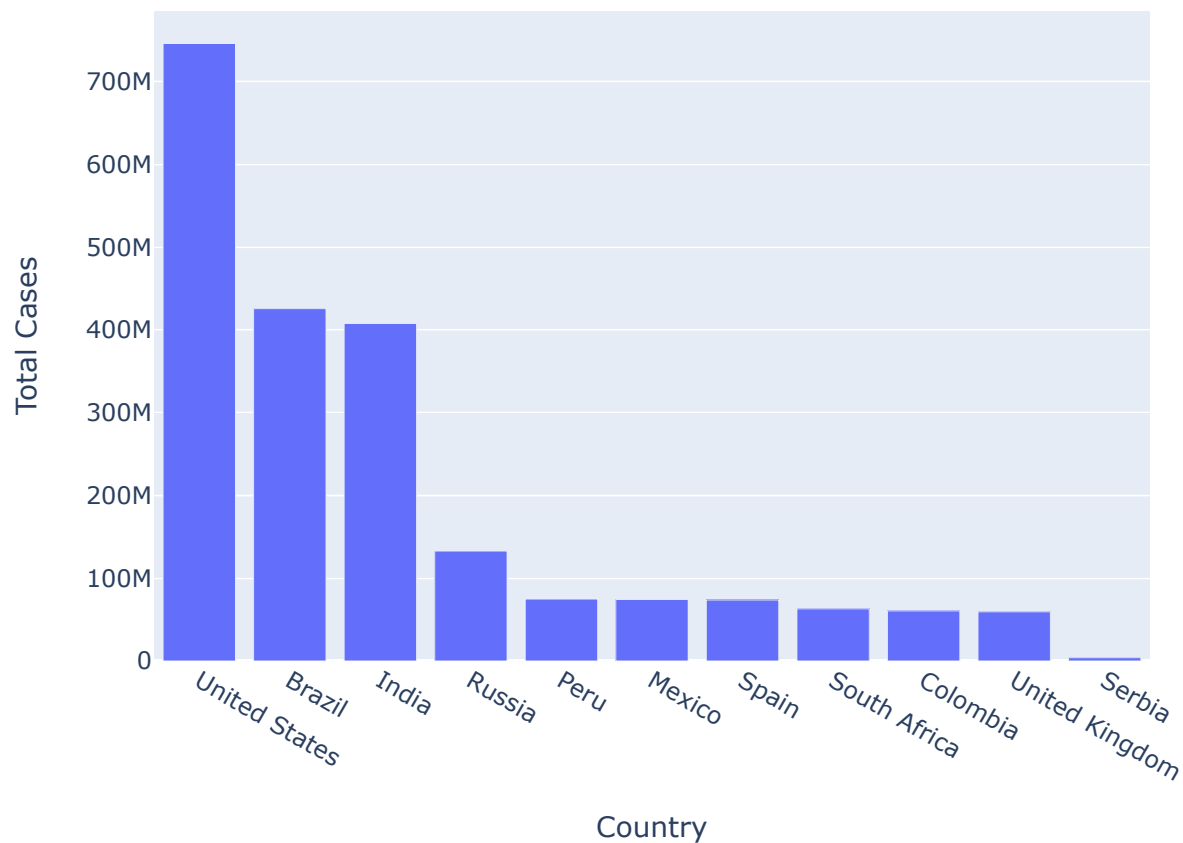
	Country Code	Country	HDI	Total Cases	Total Deaths	\
200	USA	United States	0.924000	746014098.0	26477574.0	
27	BRA	Brazil	0.759000	425704517.0	14340567.0	
90	IND	India	0.640000	407771615.0	7247327.0	
157	RUS	Russia	0.816000	132888951.0	2131571.0	
150	PER	Peru	0.599490	74882695.0	3020038.0	

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
166	SRB	Serbia	0.634418	3984174.0	87503.0

	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
166	3.073841	15.733109	6562.67	6533.19

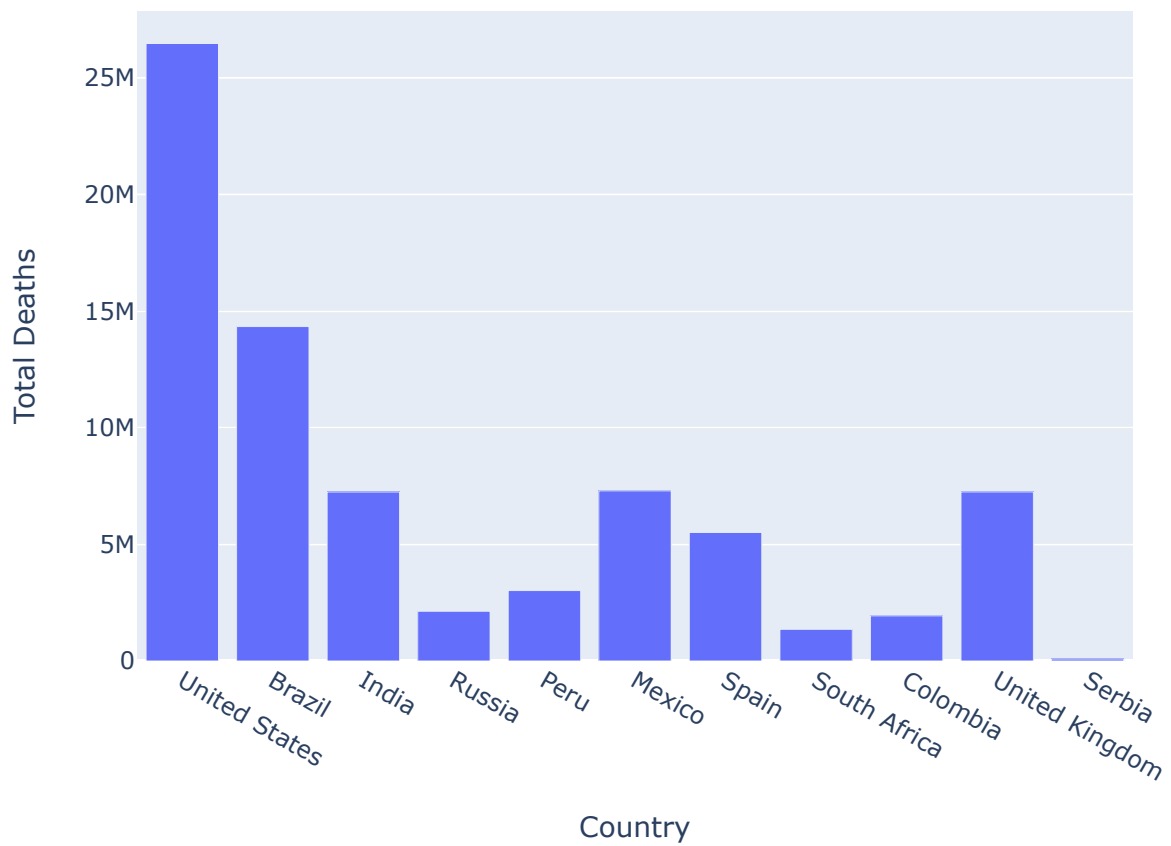
```
In [11]: figure = px.bar(dataset2, y='Total Cases', x='Country',
                        title="Countries with Highest Covid Cases")
figure.show()
```

Countries with Highest Covid Cases

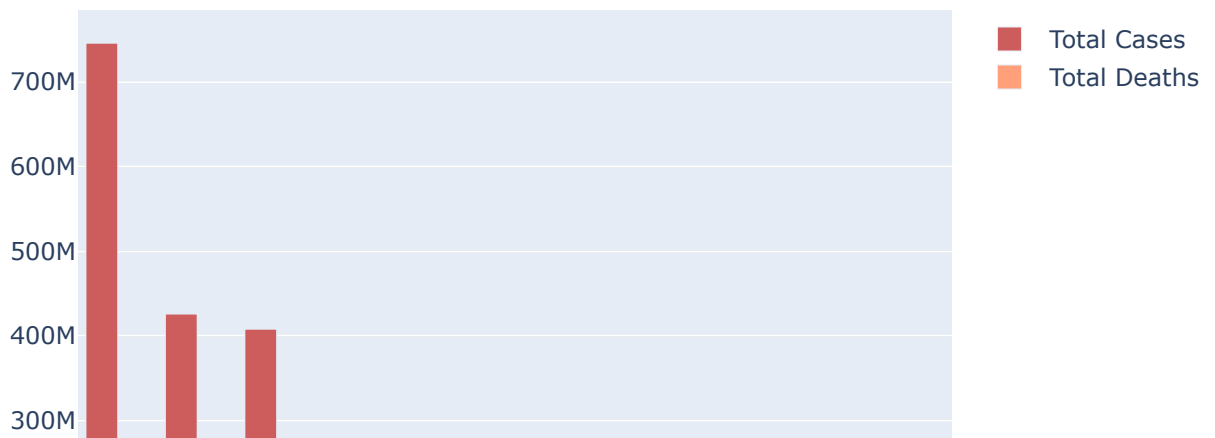


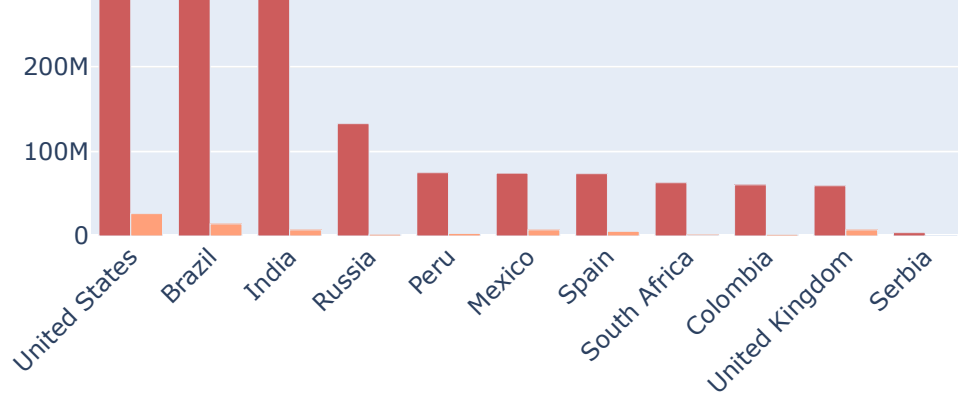
```
In [12]: figure = px.bar(dataset2, y='Total Deaths', x='Country',
                        title="Countries with Highest Deaths")
figure.show()
```

Countries with Highest Deaths



```
In [13]: fig = go.Figure()
fig.add_trace(go.Bar(
    x=dataset2["Country"],
    y=dataset2["Total Cases"],
    name='Total Cases',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=dataset2["Country"],
    y=dataset2["Total Deaths"],
    name='Total Deaths',
    marker_color='lightsalmon'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```



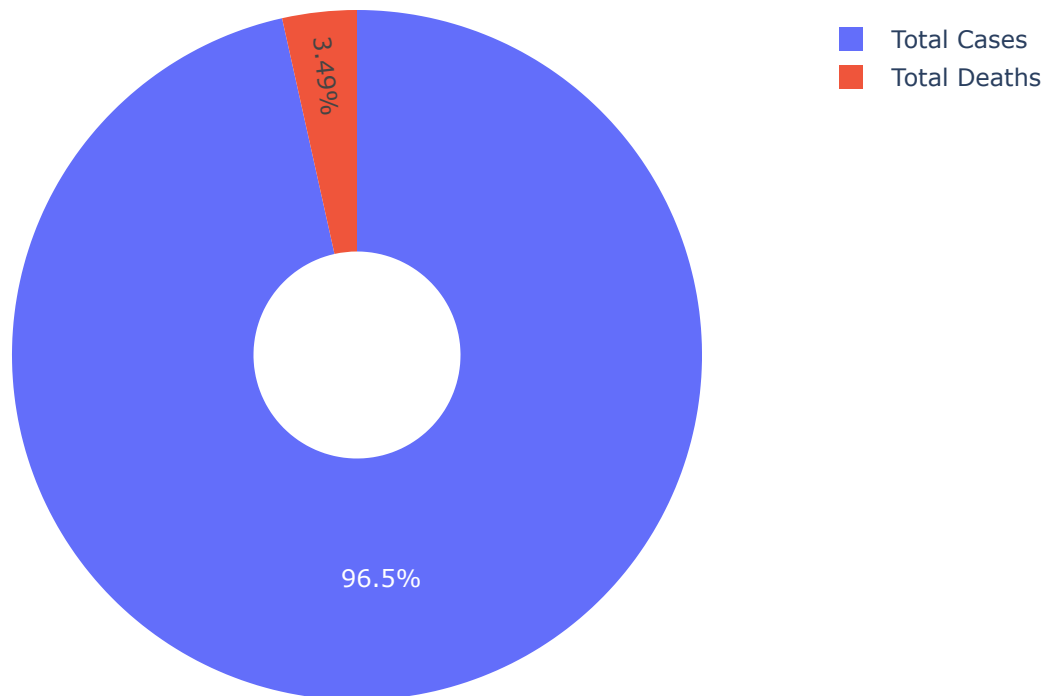


```
In [14]: # Percentage of total cases and deaths
cases = dataset2["Total Cases"].sum()
deceased = dataset2["Total Deaths"].sum()

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

fig = px.pie(dataset2, values=values, names=labels,
             title='Percentage of Total Cases and Deaths', hole=0.3)
fig.show()
```

Percentage of Total Cases and Deaths



```
In [15]: death_rate = (dataset2["Total Deaths"].sum() / dataset2["Total Cases"].sum()) * 100
print("Death Rate = ", death_rate)

Death Rate = 3.6117589845591187
```

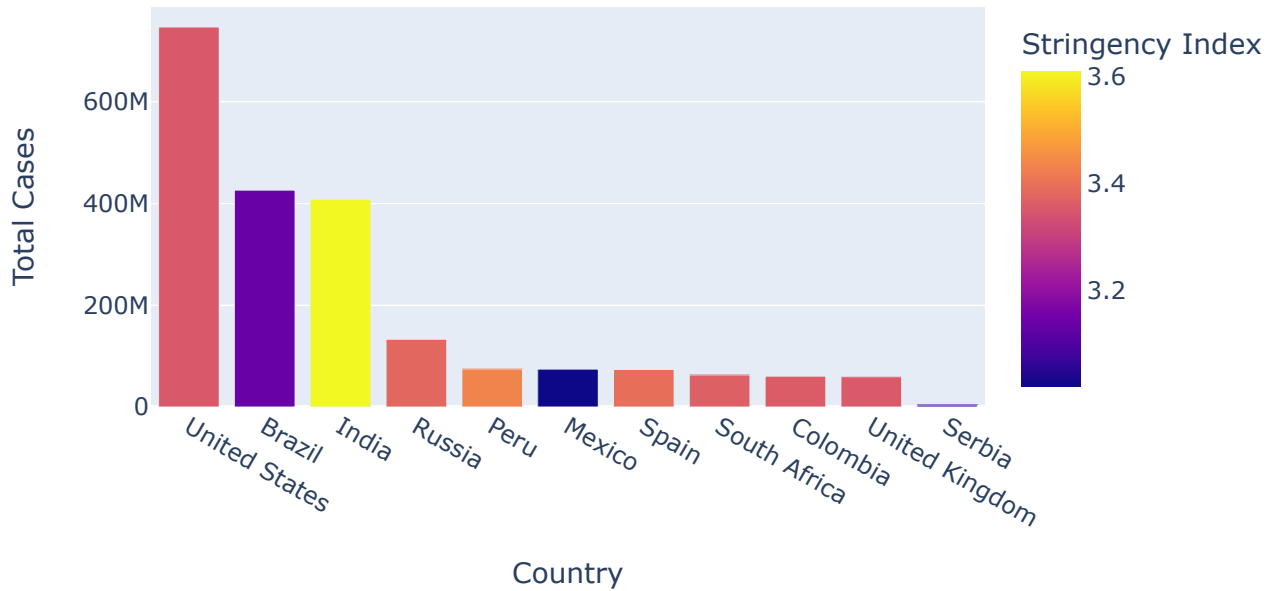
```
In [16]: fig = px.bar(dataset2, x='Country', y='Total Cases',
```

```

        hover_data=['Population', 'Total Deaths'],
        color='Stringency Index', height=400,
        title= "Stringency Index during Covid-19")
fig.show()

```

Stringency Index during Covid-19

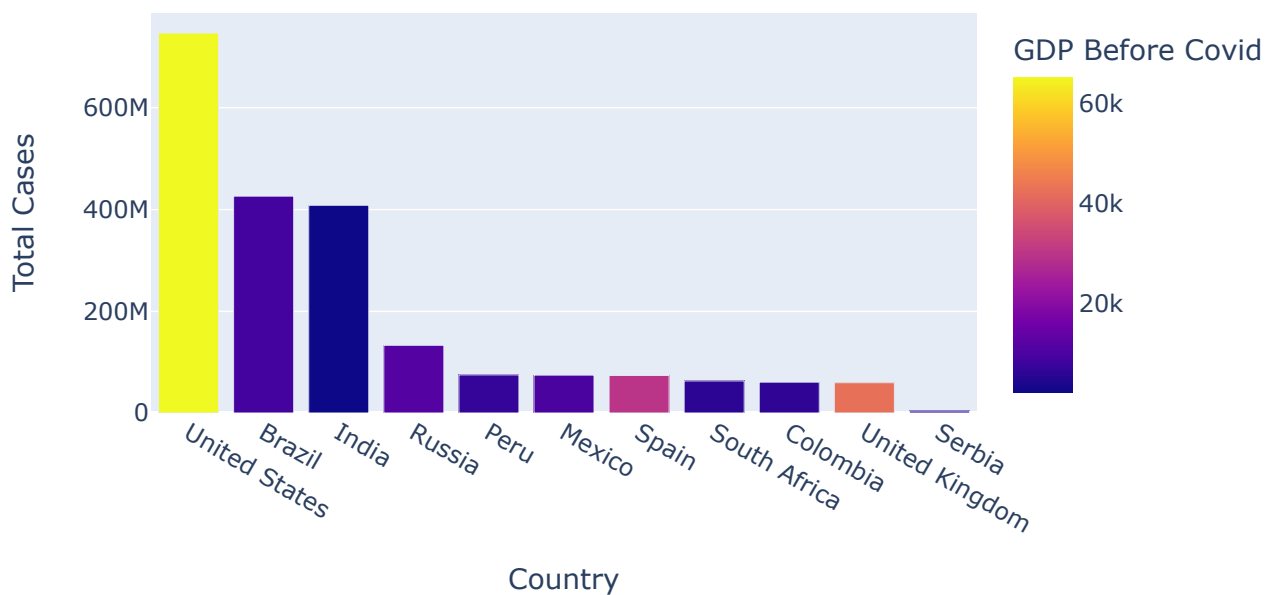


```

In [17]: fig = px.bar(dataset2, 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()

```

GDP Per Capita Before Covid-19



```

In [18]: fig = px.bar(dataset2, x='Country', y='Total Cases',
        hover_data=['Population', 'Total Deaths'],

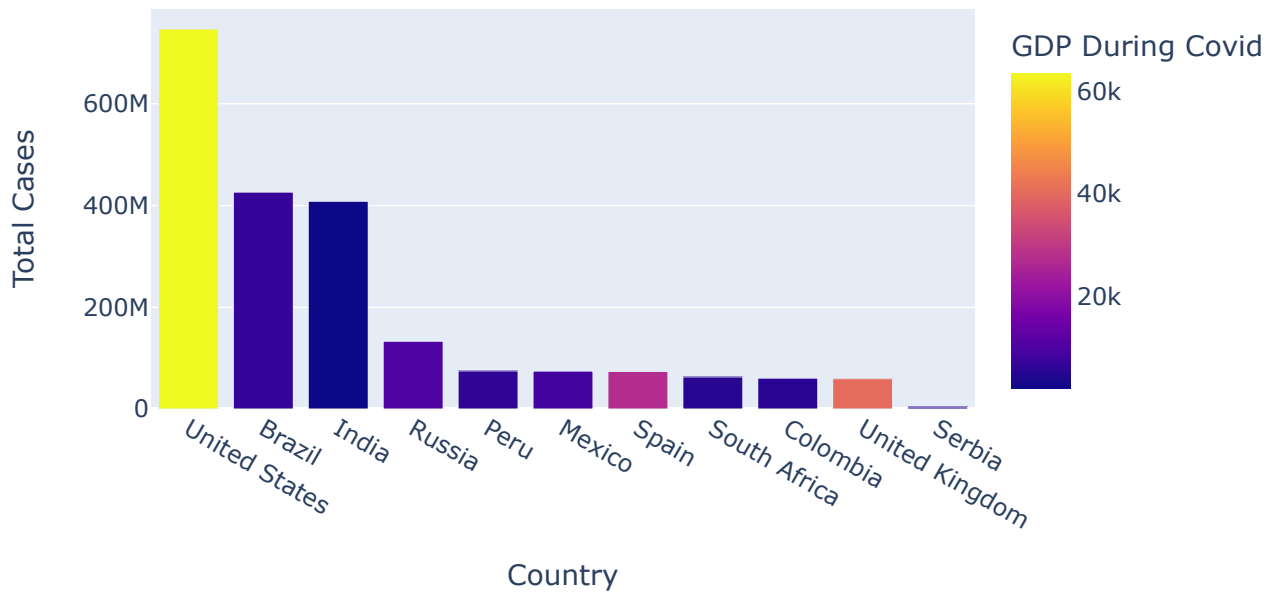
```

```

color='GDP During Covid', height=400,
title="GDP Per Capita During Covid-19")
fig.show()

```

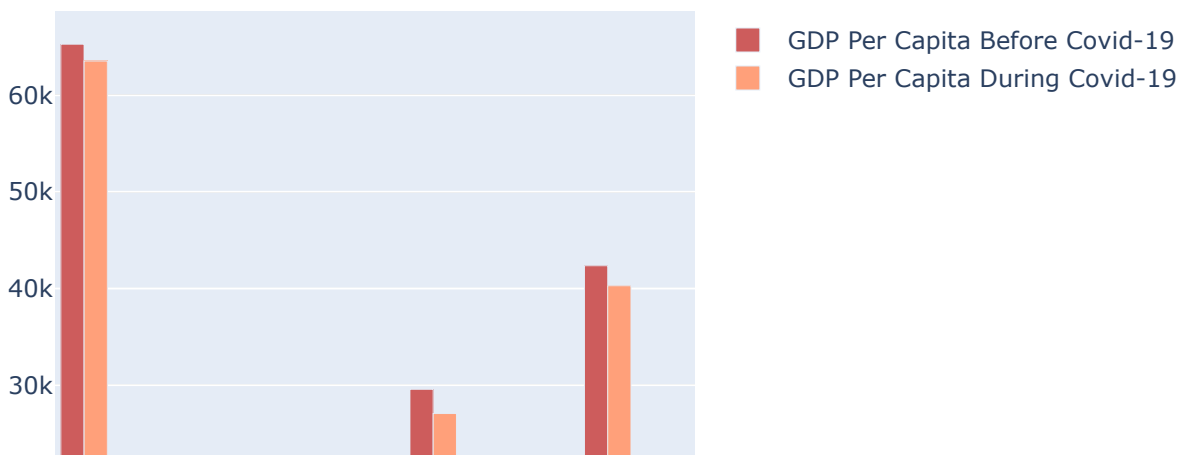
GDP Per Capita During Covid-19

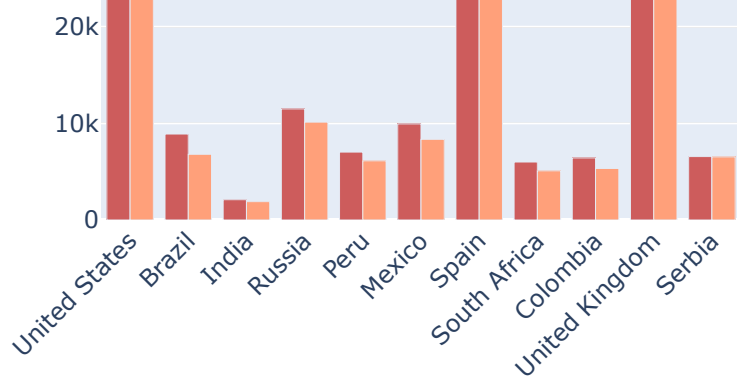


```

In [19]: fig = go.Figure()
fig.add_trace(go.Bar(
    x=dataset2["Country"],
    y=dataset2["GDP Before Covid"],
    name='GDP Per Capita Before Covid-19',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=dataset2["Country"],
    y=dataset2["GDP During Covid"],
    name='GDP Per Capita During Covid-19',
    marker_color='lightsalmon'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()

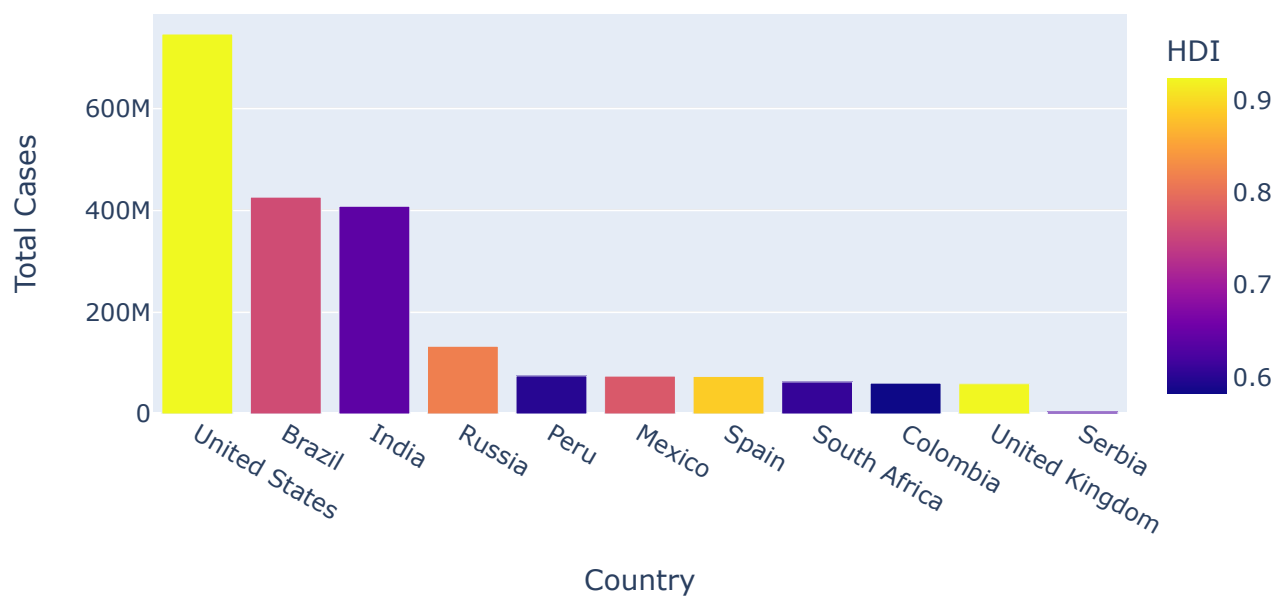
```





```
In [20]: fig = px.bar(dataset2, x='Country', y='Total Cases',
                    hover_data=['Population', 'Total Deaths'],
                    color='HDI', height=400,
                    title="Human Development Index during Covid-19")
fig.show()
```

Human Development Index during Covid-19



```
In [21]: # make a data set of 10 countries with the least total cases
dataset3 = dataset.tail(10)
print(dataset3)
```

	Country Code	Country	HDI	Total Cases	Total Deaths	\
107	LAO	Laos	0.427241	4039.0	0.0	
55	DMA	Dominica	0.513146	3977.0	0.0	
159	KNA	Saint Kitts and Nevis	0.550422	3246.0	0.0	
77	GRL	Greenland	0.000000	2678.0	0.0	
65	FLK	Falkland Islands	0.000000	2467.0	0.0	
204	VAT	Vatican	0.000000	2415.0	0.0	
130	MSR	Montserrat	0.000000	2383.0	177.0	
5	AIA	Anguilla	0.000000	614.0	0.0	
173	SLB	Solomon Islands	0.007429	11.0	0.0	
87	HKG	Hong Kong	0.161847	0.0	0.0	
	Stringency Index	Population				
107	2.516712	15.800031				
55	2.681967	11.184296				
159	0.000000	10.881663				

77	2.588435	10.946799
65	2.122958	8.155649
204	0.000000	6.695799
130	2.477077	8.516993
5	2.326181	9.615939
173	0.000000	13.439912
87	0.643980	15.830012

```
In [22]: # add columns: GDP per capita before and during covid
pd.options.mode.chained_assignment = None
dataset3["GDP Before Covid"] = [2635.8, 8516.3, 22046.7,
                                54270.8, 70233, 21198,
                                13000, 20000, 2258.40, 48000]
dataset3["GDP During Covid"] = [2629.7, 7003.9, 18437.7,
                                53041.3, 65245, 20000,
                                12000, 18000, 2000, 46000]

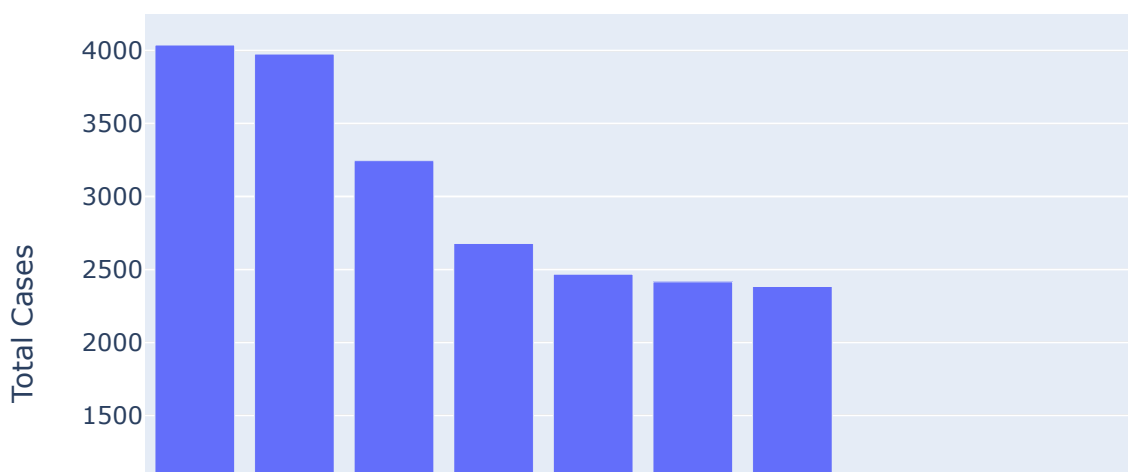
print(dataset3)
```

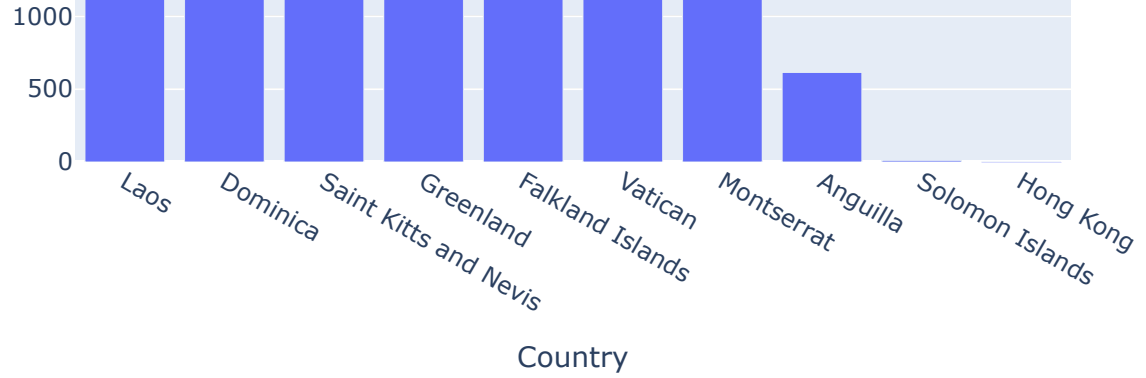
	Country Code	Country	HDI	Total Cases	Total Deaths	\
107	LAO	Laos	0.427241	4039.0	0.0	
55	DMA	Dominica	0.513146	3977.0	0.0	
159	KNA	Saint Kitts and Nevis	0.550422	3246.0	0.0	
77	GRL	Greenland	0.000000	2678.0	0.0	
65	FLK	Falkland Islands	0.000000	2467.0	0.0	
204	VAT	Vatican	0.000000	2415.0	0.0	
130	MSR	Montserrat	0.000000	2383.0	177.0	
5	AIA	Anguilla	0.000000	614.0	0.0	
173	SLB	Solomon Islands	0.007429	11.0	0.0	
87	HKG	Hong Kong	0.161847	0.0	0.0	

	Stringency Index	Population	GDP Before Covid	GDP During Covid
107	2.516712	15.800031	2635.8	2629.7
55	2.681967	11.184296	8516.3	7003.9
159	0.000000	10.881663	22046.7	18437.7
77	2.588435	10.946799	54270.8	53041.3
65	2.122958	8.155649	70233.0	65245.0
204	0.000000	6.695799	21198.0	20000.0
130	2.477077	8.516993	13000.0	12000.0
5	2.326181	9.615939	20000.0	18000.0
173	0.000000	13.439912	2258.4	2000.0
87	0.643980	15.830012	48000.0	46000.0

```
In [23]: figure = px.bar(dataset3, y='Total Cases', x='Country',
                        title="Countries with Highest Covid Cases")
figure.show()
```

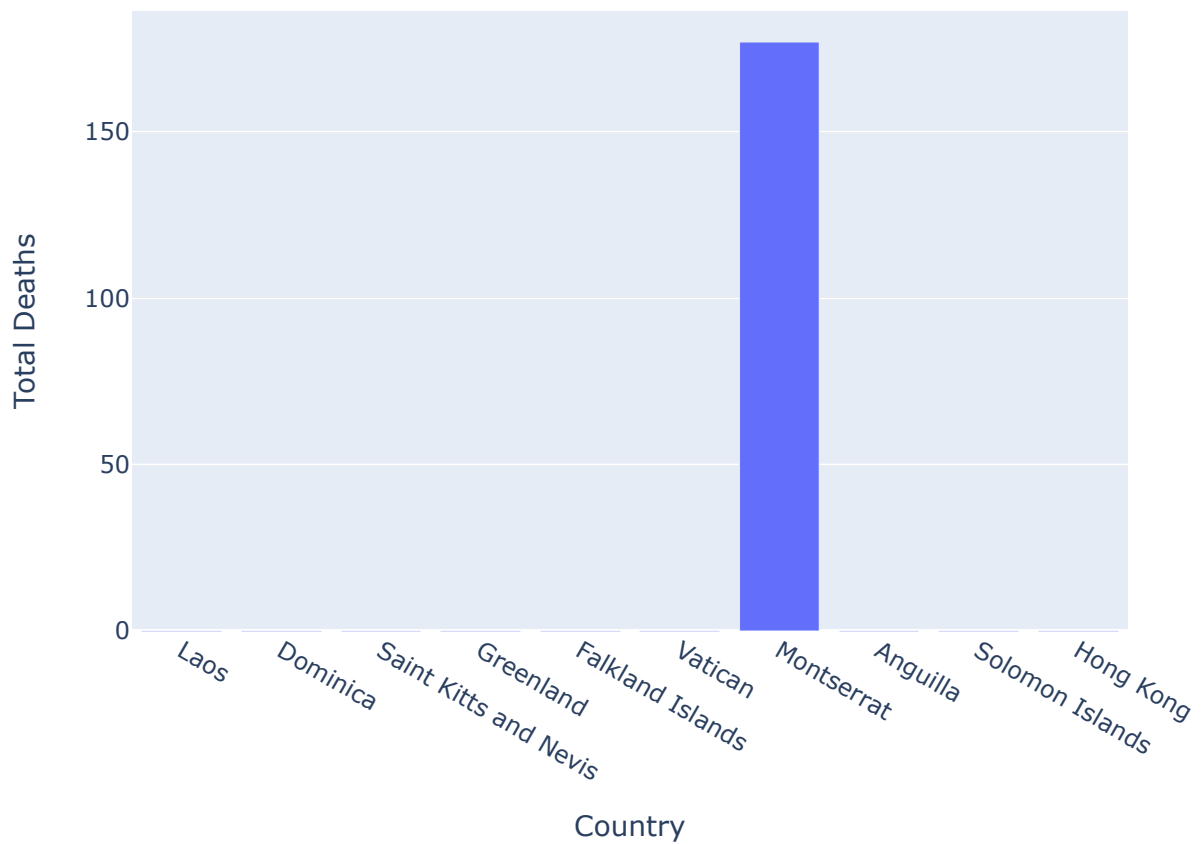
Countries with Highest Covid Cases





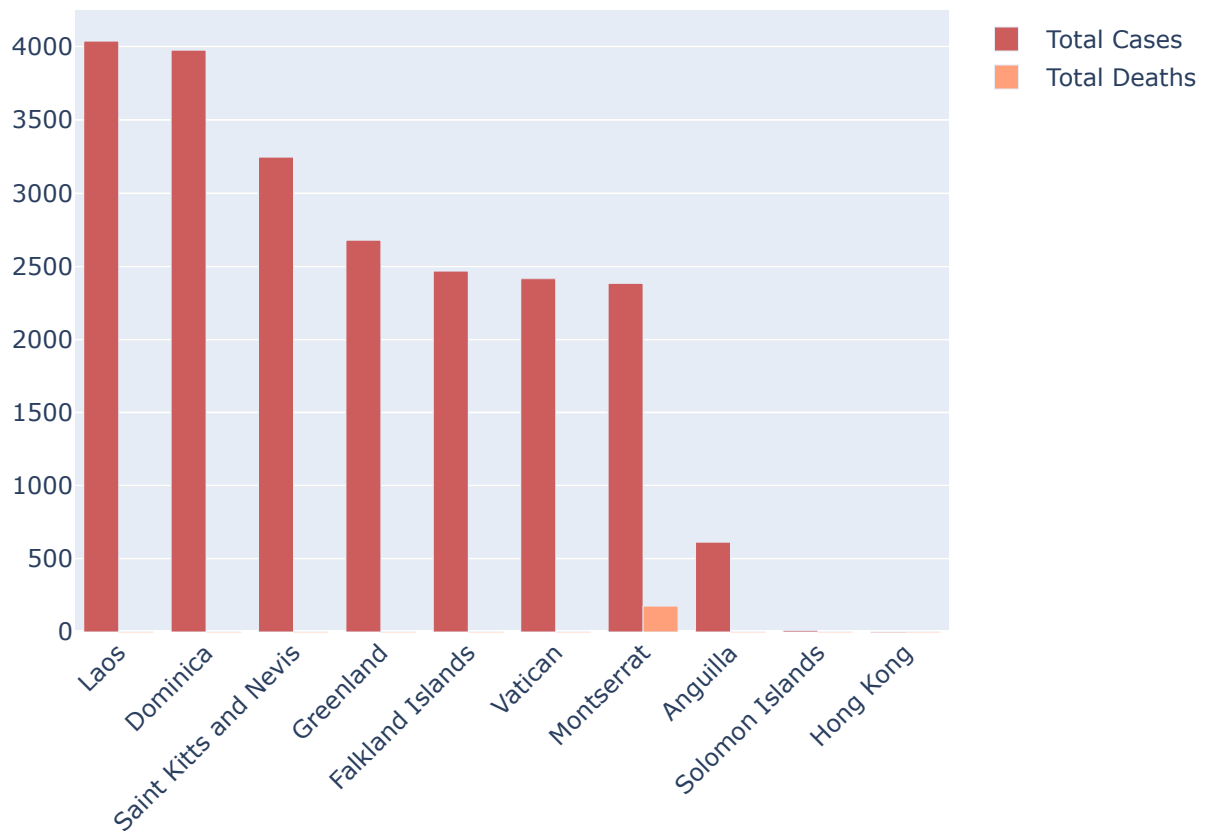
```
In [24]: figure = px.bar(dataset3, y='Total Deaths', x='Country',
                        title="Countries with Highest Deaths")
figure.show()
```

Countries with Highest Deaths



```
In [25]: fig = go.Figure()
fig.add_trace(go.Bar(
    x=dataset3["Country"],
    y=dataset3["Total Cases"],
    name='Total Cases',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=dataset3["Country"],
    y=dataset3
    ["Total Deaths"],
    name='Total Deaths',
    marker_color='lightsalmon'
))
```

```
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```

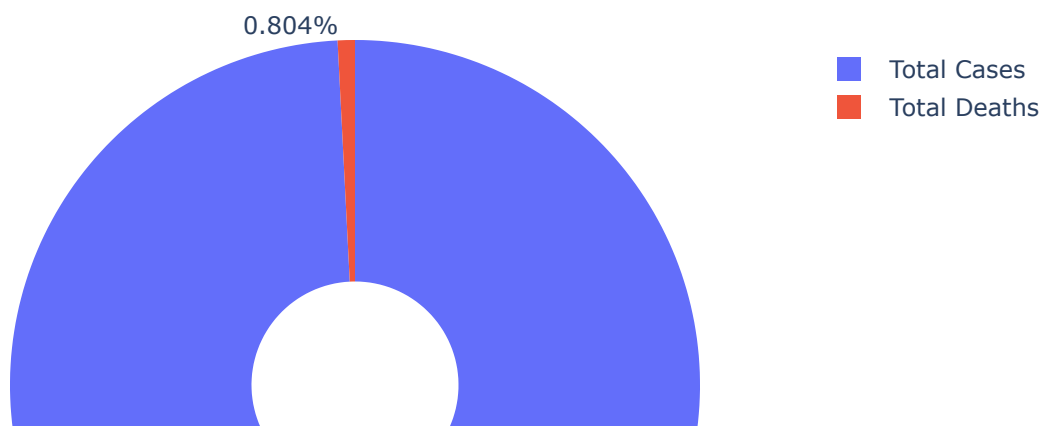


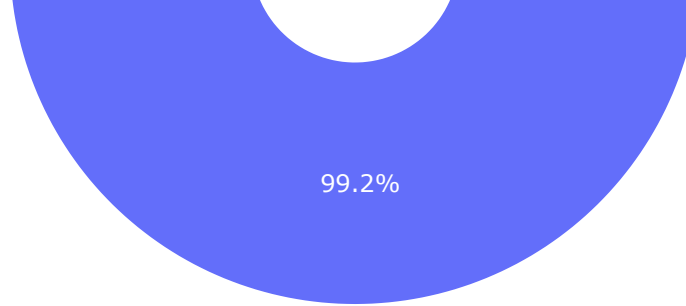
```
In [26]: # Percentage of Total Cases and Deaths
cases = dataset3["Total Cases"].sum()
deceased = dataset3["Total Deaths"].sum()

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

fig = px.pie(dataset3, values=values, names=labels,
             title='Percentage of Total Cases and Deaths', hole=0.3)
fig.show()
```

Percentage of Total Cases and Deaths



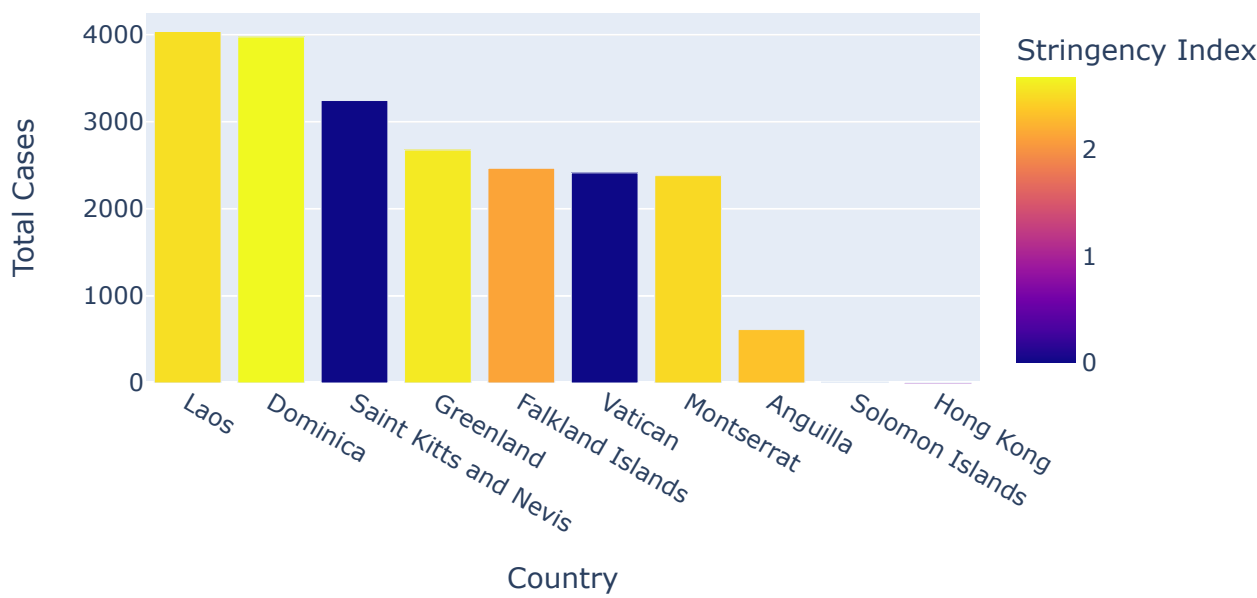


```
In [27]: death_rate = (dataset3["Total Deaths"].sum() / dataset3["Total Cases"].sum()) * 100
print("Death Rate = ", death_rate)
```

Death Rate = 0.8108108108108109

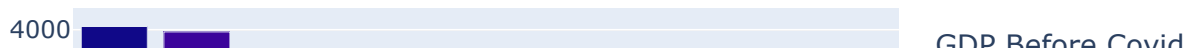
```
In [28]: fig = px.bar(dataset3, x='Country', y='Total Cases',
                    hover_data=['Population', 'Total Deaths'],
                    color='Stringency Index', height=400,
                    title="Stringency Index during Covid-19")
fig.show()
```

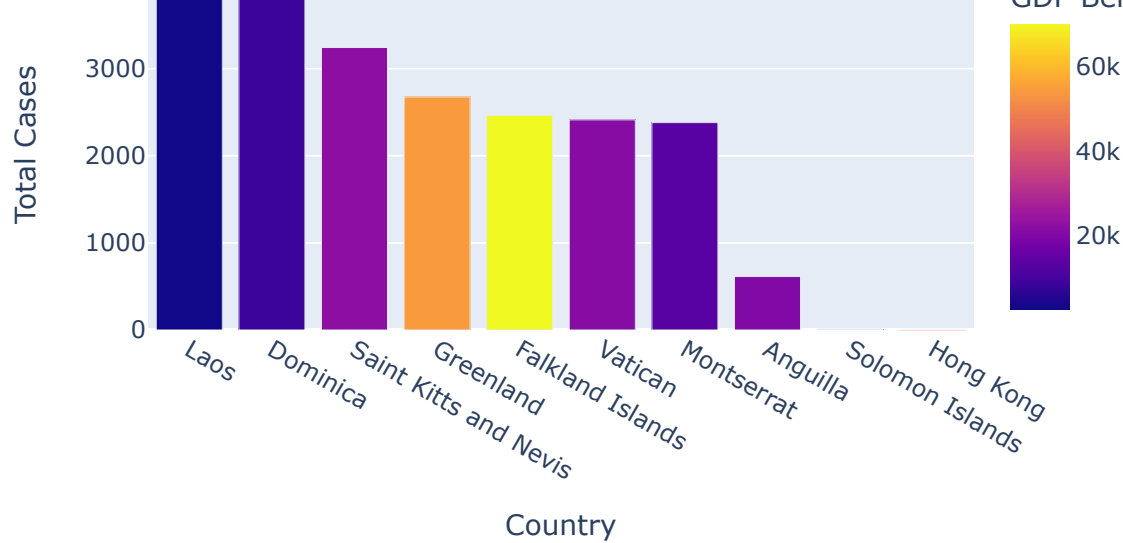
Stringency Index during Covid-19



```
In [29]: fig = px.bar(dataset3, 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()
```

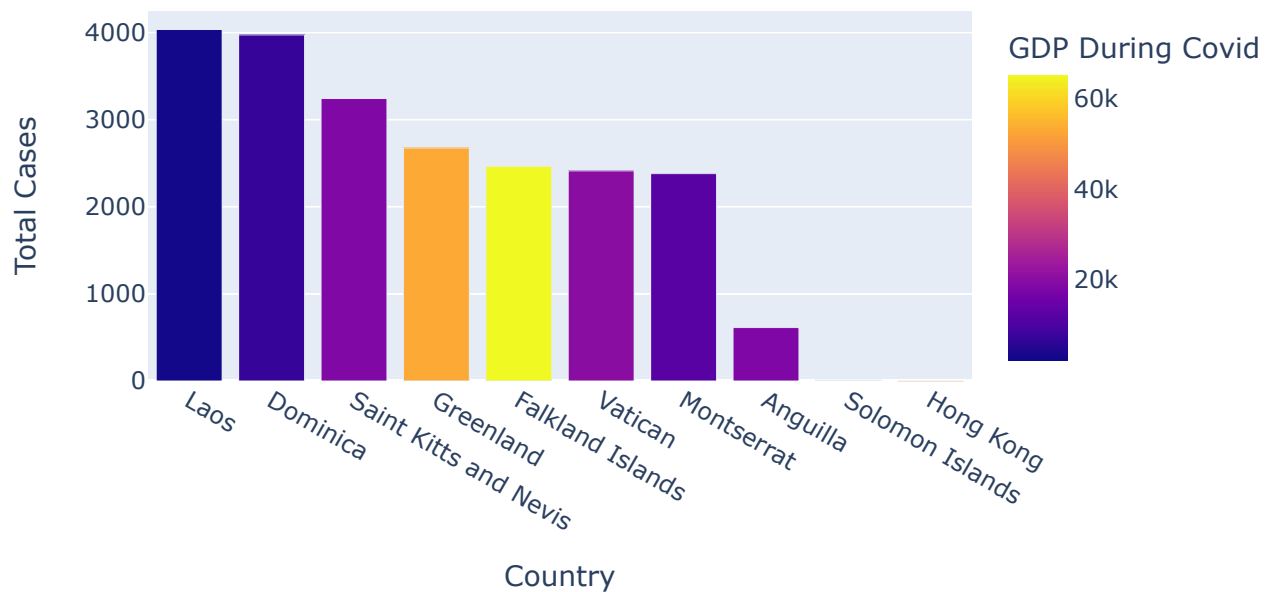
GDP Per Capita Before Covid-19





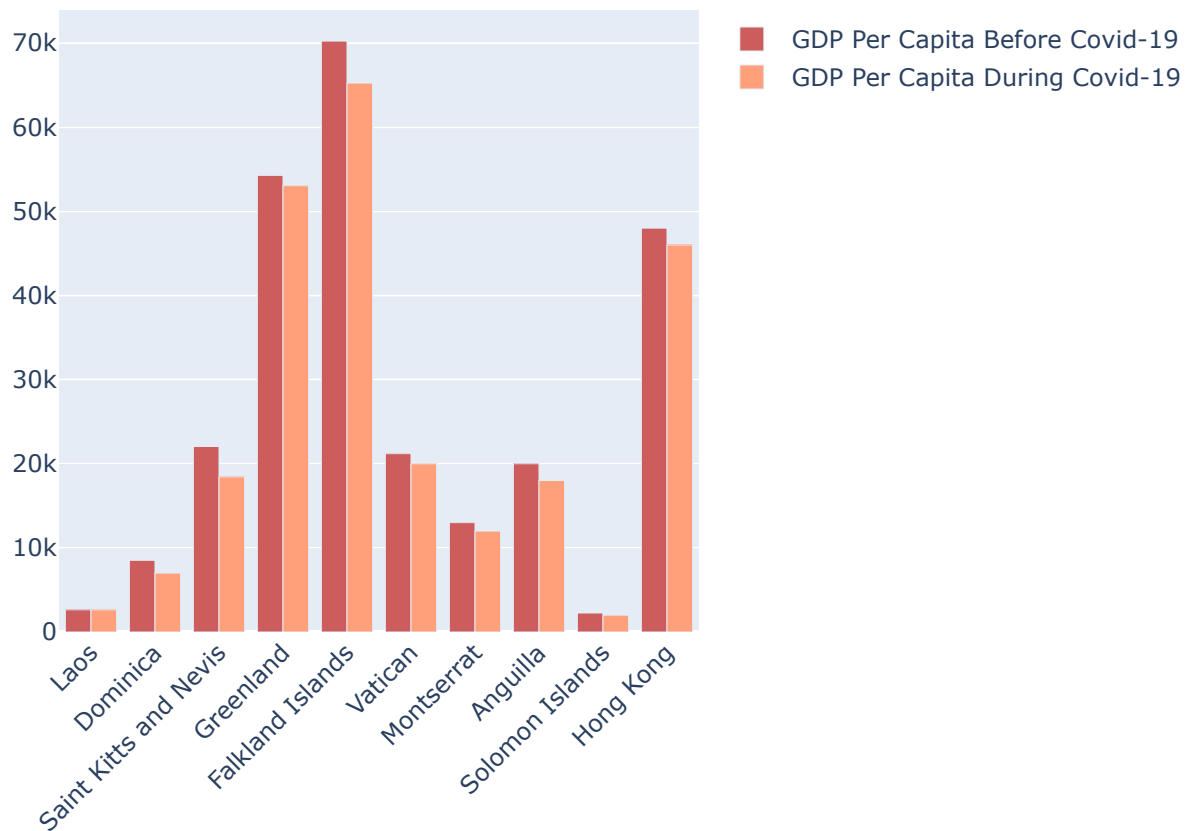
```
In [30]: fig = px.bar(dataset3, x='Country', y='Total Cases',
                    hover_data=['Population', 'Total Deaths'],
                    color='GDP During Covid', height=400,
                    title="GDP Per Capita During Covid-19")
fig.show()
```

GDP Per Capita During Covid-19



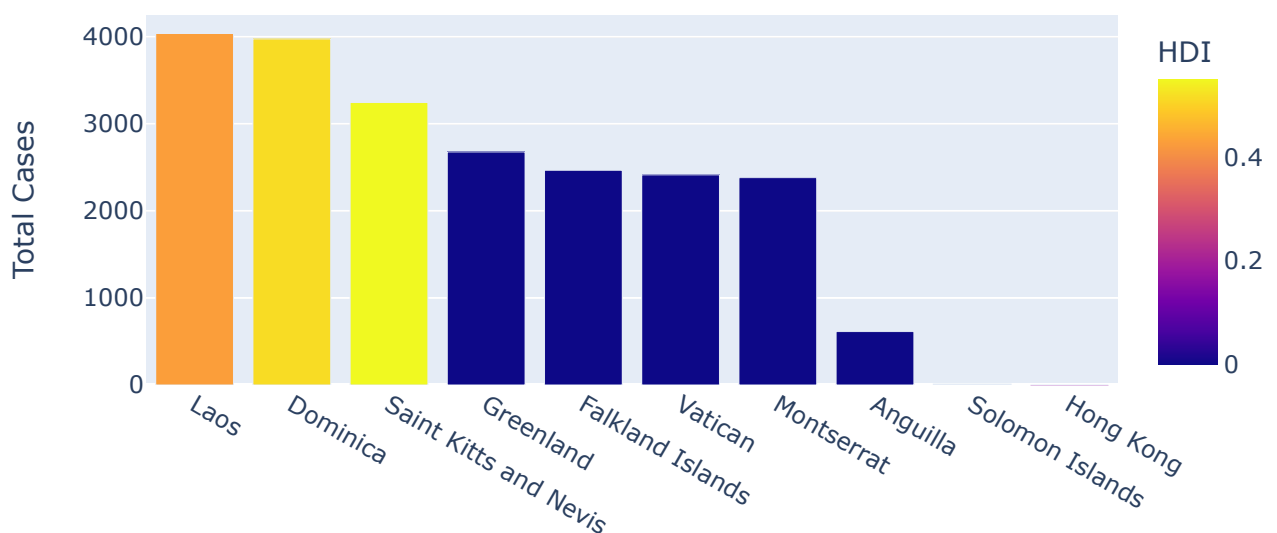
```
In [31]: fig = go.Figure()
fig.add_trace(go.Bar(
    x=dataset3["Country"],
    y=dataset3["GDP Before Covid"],
    name='GDP Per Capita Before Covid-19',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=dataset3["Country"],
    y=dataset3["GDP During Covid"],
    name='GDP Per Capita During Covid-19',
    marker_color='lightsalmon'
))
```

```
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```



```
In [32]: fig = px.bar(dataset3, x='Country', y='Total Cases',
                    hover_data=['Population', 'Total Deaths'],
                    color='HDI', height=400,
                    title="Human Development Index during Covid-19")
fig.show()
```

Human Development Index during Covid-19



```
In [33]: bb=dataset2.tail(1)
```

```
In [34]: fig = go.Figure()
fig.add_trace(go.Bar(
    x=bb["Country"],
    y=bb["GDP Before Covid"],
    name='GDP Per Capita Before Covid-19',
    marker_color='indianred'
))
fig.add_trace(go.Bar(
    x=bb["Country"],
    y=bb["GDP During Covid"],
    name='GDP Per Capita During Covid-19',
    marker_color='lightsalmon'
))
fig.update_layout(barmode='group', xaxis_tickangle=-45)
fig.show()
```

