COVID_19 IMPACT ANALYSIS EDA

OBJECTIVE OF PROJECT:

The first wave of covid-19 impacted the global economy as the world was never ready for the pandemic. It resulted in a rise in cases, a rise in deaths, a rise in unemployment and a rise in poverty, resulting in an economic slowdown. Here, you are required to analyze the spread of Covid-19 cases and all the impacts of covid-19 on the economy.

In [38]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
```

In [46]:
```
df = pd.read_csv('transformed_data.csv')
```

In [47]:
```
df.head()
```

Out[47]:

	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

DATA CLEANING

In [48]:
```
df.isnull().sum()
```

Out[48]:
```
CODE          0
COUNTRY       0
DATE          0
HDI        6202
TC            0
TD            0
STI           0
POP           0
GDPCAP        0
dtype: int64
```

In [49]:
```
df = df.dropna()
df.head()
```

Out[49]:

	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 [50]:
```
df.isnull().sum()
```

Out[50]:
```
CODE       0
COUNTRY    0
DATE       0
HDI        0
TC         0
TD         0
STI        0
POP        0
GDPCAP     0
dtype: int64
```

In [51]:
```
df.dtypes
```

Out[51]:
```
CODE        object
COUNTRY     object
DATE        object
HDI        float64
TC         float64
TD         float64
STI        float64
POP        float64
GDPCAP     float64
dtype: object
```

DATA PREPROCESSING

In [52]:
```
df['infection_rate'] = df['TC'] / df['POP']
df.head()
```

Out[52]:

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

In [53]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 44216 entries, 0 to 50417
Data columns (total 10 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   CODE            44216 non-null  object 
 1   COUNTRY         44216 non-null  object 
 2   DATE            44216 non-null  object 
 3   HDI             44216 non-null  float64
 4   TC              44216 non-null  float64
 5   TD              44216 non-null  float64
 6   STI             44216 non-null  float64
 7   POP             44216 non-null  float64
 8   GDPCAP          44216 non-null  float64
 9   infection_rate  44216 non-null  float64
dtypes: float64(7), object(3)
memory usage: 3.7+ MB
```

In [54]:
```
df.describe()
```

Out[54]:

	HDI	TC	TD	STI	POP	GDPCAP	infection_rate
count	44216.000000	44216.000000	44216.000000	44216.000000	44216.000000	44216.000000	44216.000000
mean	0.720139	7.102211	3.729883	3.372483	16.018503	9.154138	0.440603
std	0.160902	3.676823	3.111586	1.534602	1.932219	1.753255	0.213317
min	0.000000	0.000000	0.000000	0.000000	10.548940	0.000000	0.000000
25%	0.601000	4.672829	0.000000	3.324316	14.901792	8.421078	0.308705
50%	0.752000	7.577634	3.663562	4.050219	16.127974	9.492126	0.484075
75%	0.847000	9.827902	5.966147	4.353884	17.326136	10.266848	0.605330
max	0.953000	15.914092	12.299900	4.605170	21.087439	11.669379	0.811214

In [55]:
```
country = df.groupby(["COUNTRY"]).sum()
country.head()
```

```
C:\Users\Raj\AppData\Local\Temp\ipykernel_11948\2339608990.py:1: FutureWarning: The de
fault value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future versio
n, numeric_only will default to False. Either specify numeric_only or select only colu
mns which should be valid for the function.
  country = df.groupby(["COUNTRY"]).sum()
```

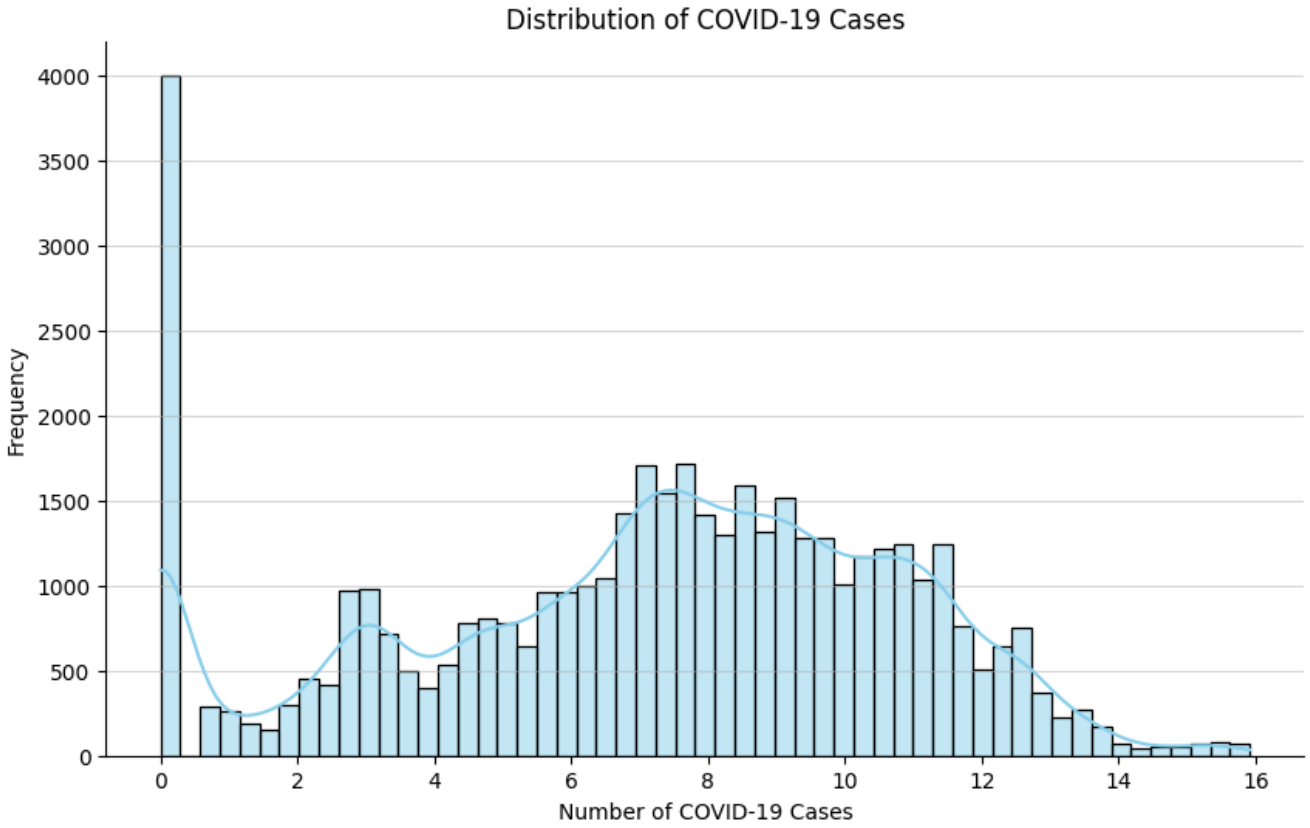
Out[55]:

COUNTRY	HDI	TC	TD	STI	POP	GDPCAP	infection_rate
Afghanistan	146.412	2000.646094	1226.948181	896.603996	5138.306531	2204.339821	114.471557
Albania	176.625	1702.240756	929.368800	883.653354	3346.320751	2109.632744	114.455307
Algeria	221.676	2052.510847	1406.216387	939.379534	5173.314864	2804.947935	116.644396
Andorra	193.908	1465.828250	786.655112	787.230267	2543.629175	0.000000	130.238003
Angola	123.172	1203.978763	590.138675	871.874556	3669.286969	1837.821379	69.562152

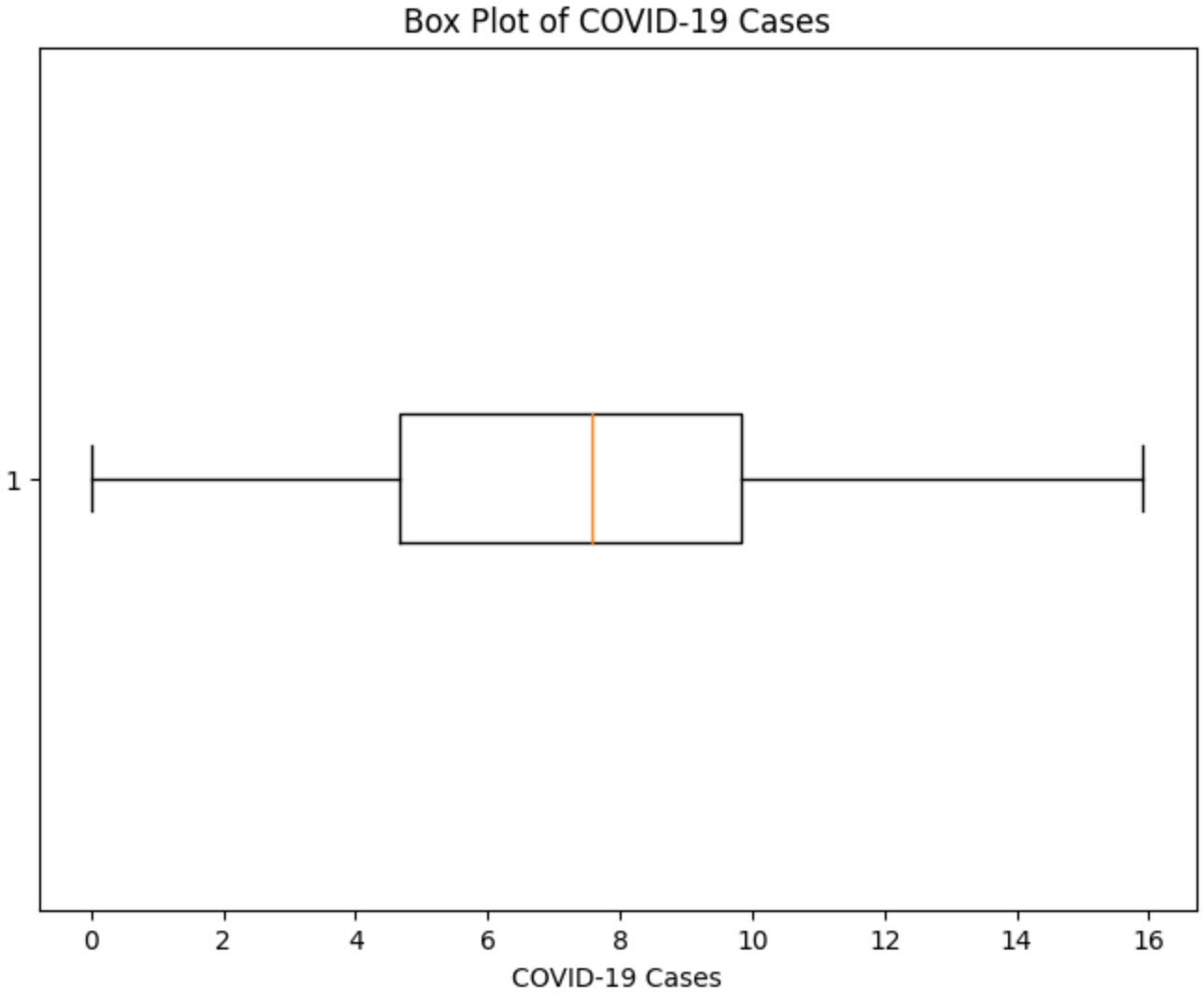
EXPLORATORY DATA ANALYSIS

In [56]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

In [57]:
```
# Assuming 'cases' is a Pandas Series containing COVID-19 case data
plt.figure(figsize=(10, 6))
sns.histplot(df['TC'], kde=True, color='skyblue')
plt.xlabel('Number of COVID-19 Cases')
plt.ylabel('Frequency')
plt.title('Distribution of COVID-19 Cases')
plt.grid(axis='y', alpha=0.5)
sns.despine()
plt.show()
```

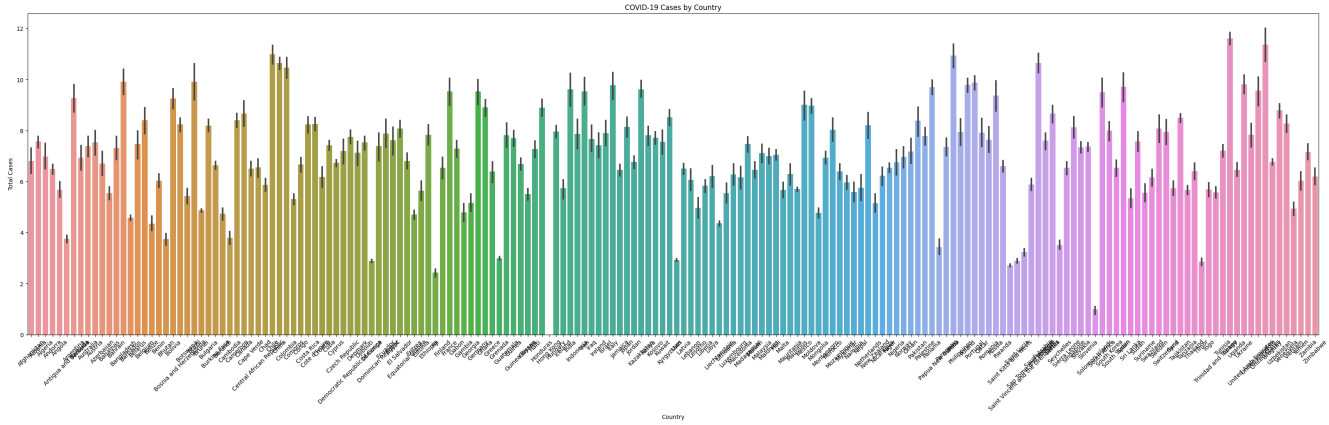


In [58]:
```
# Box Plot
plt.figure(figsize=(8, 6))
plt.boxplot(df['TC'], vert=False)
plt.xlabel('COVID-19 Cases')
plt.title('Box Plot of COVID-19 Cases')
plt.show()
```



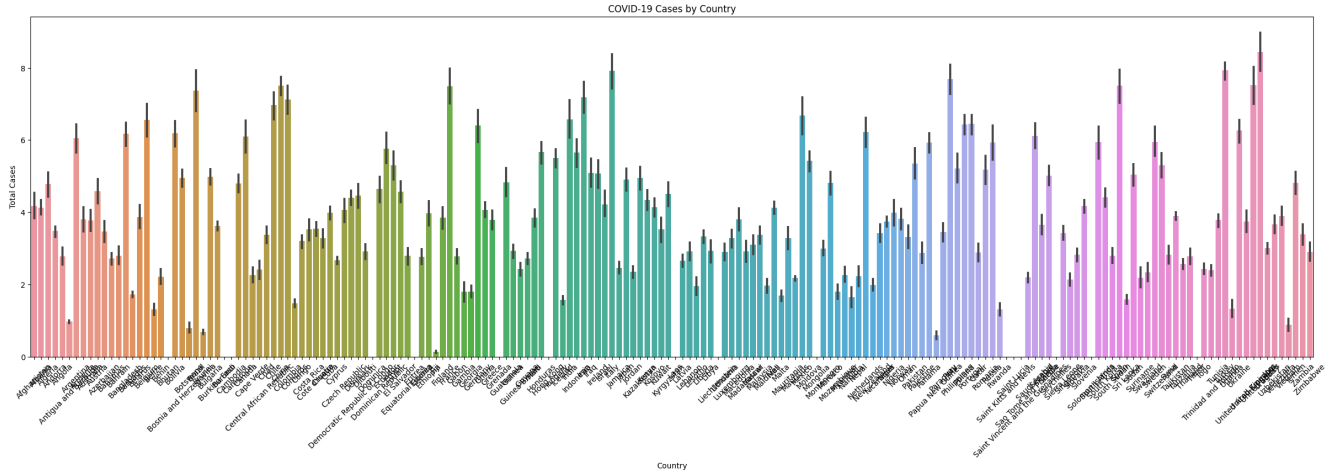
BAR PLOT OF COVID-19 CASES BY COUNTRY

In [61]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(40, 10))
sns.barplot(x='COUNTRY', y='TC', data=df)
plt.xlabel('Country')
plt.ylabel('Total Cases')
plt.title('COVID-19 Cases by Country')
plt.xticks(rotation=45)
plt.show()
```



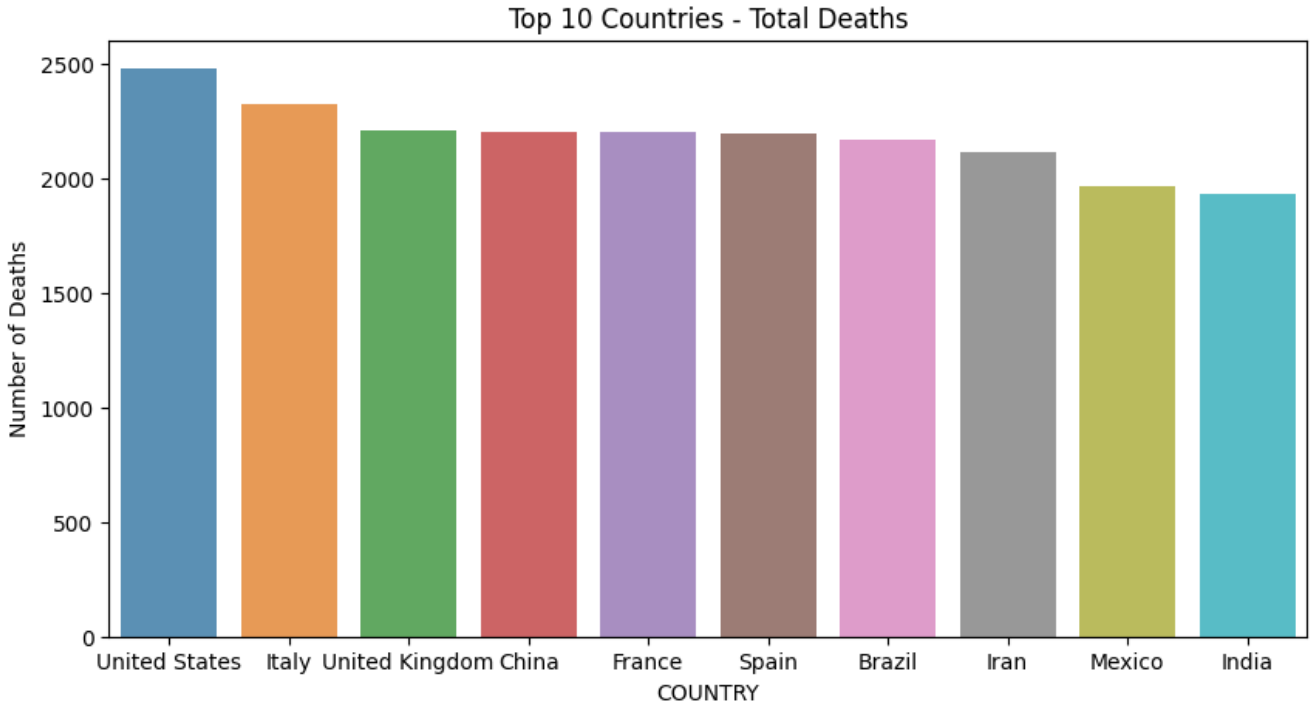
BAR PLOT OF COVID-19 DEATHS OF COUNTRY

In [60]:
```
import matplotlib.pyplot as plt
import seaborn as sns

plt.figure(figsize=(30, 8))
sns.barplot(x='COUNTRY', y='TD', data=df)
plt.xlabel('Country')
plt.ylabel('Total Deaths')
plt.title('COVID-19 Deaths by Country')
plt.xticks(rotation=45)
plt.show()
```



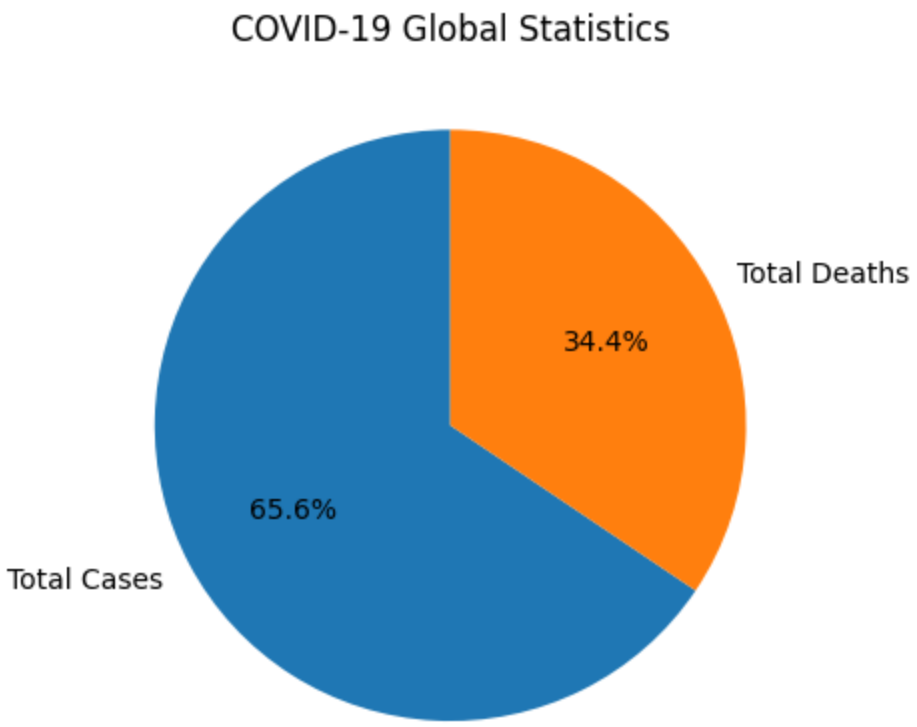
MOST DEATHS COUNTRY WISE

In [16]:
```
Country_death = df.groupby("COUNTRY")["TD"].sum().sort_values(ascending=False)[:10]
plt.figure(figsize=(10, 5))
sns.barplot(x=Country_death.index, y=Country_death.values, alpha=0.8)
plt.title('Top 10 Countries - Total Deaths')
plt.ylabel('Number of Deaths')
plt.show()
```



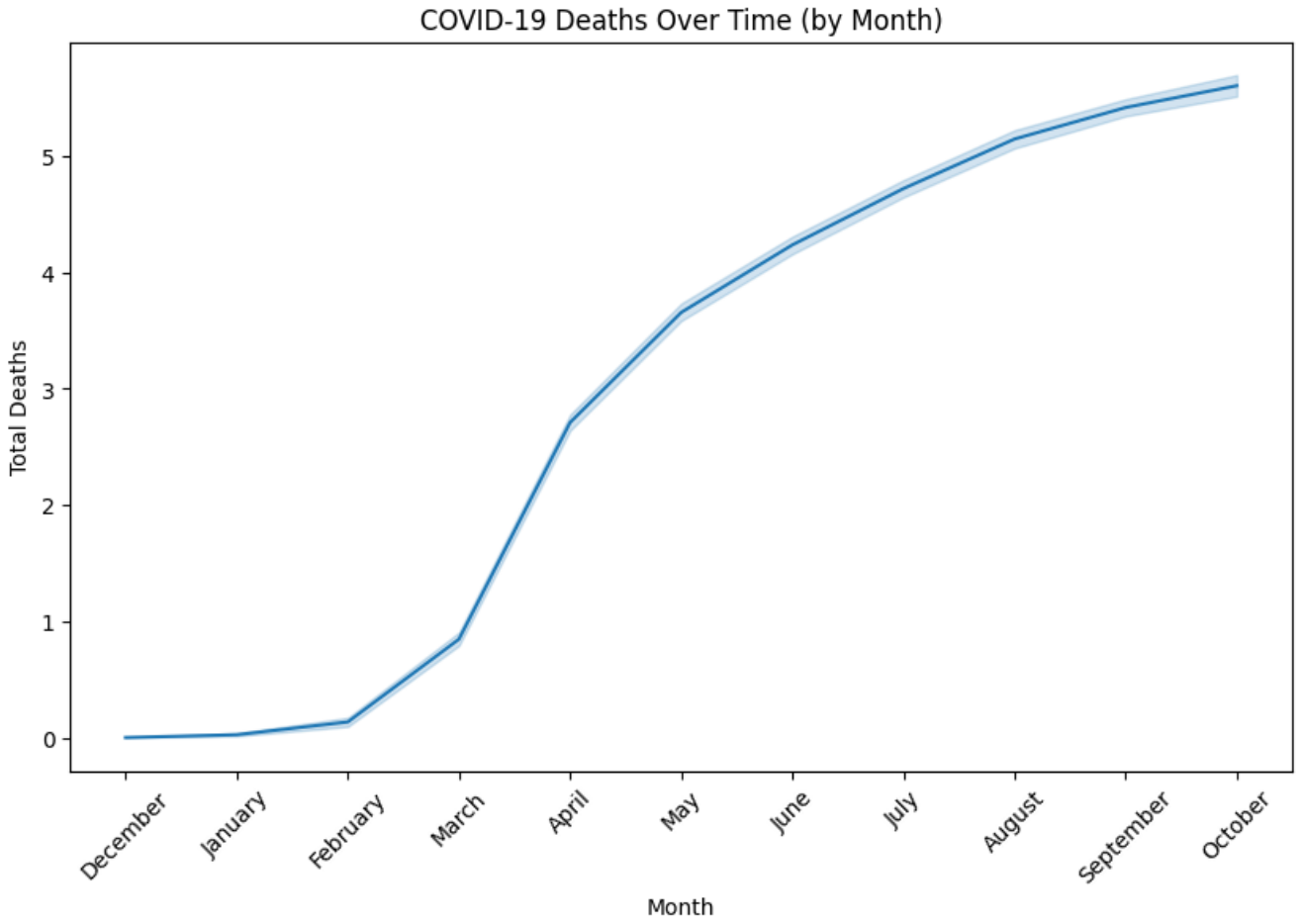
TOTAL CASES VS TOTAL DEATHS

In [17]:
```
import matplotlib.pyplot as plt

total_cases = df['TC'].sum()
total_deaths = df['TD'].sum()
data = [total_cases, total_deaths]
labels = ['Total Cases', 'Total Deaths']

plt.pie(data, labels=labels, autopct='%1.1f%%', startangle=90)

plt.title('COVID-19 Global Statistics')

plt.show()
```



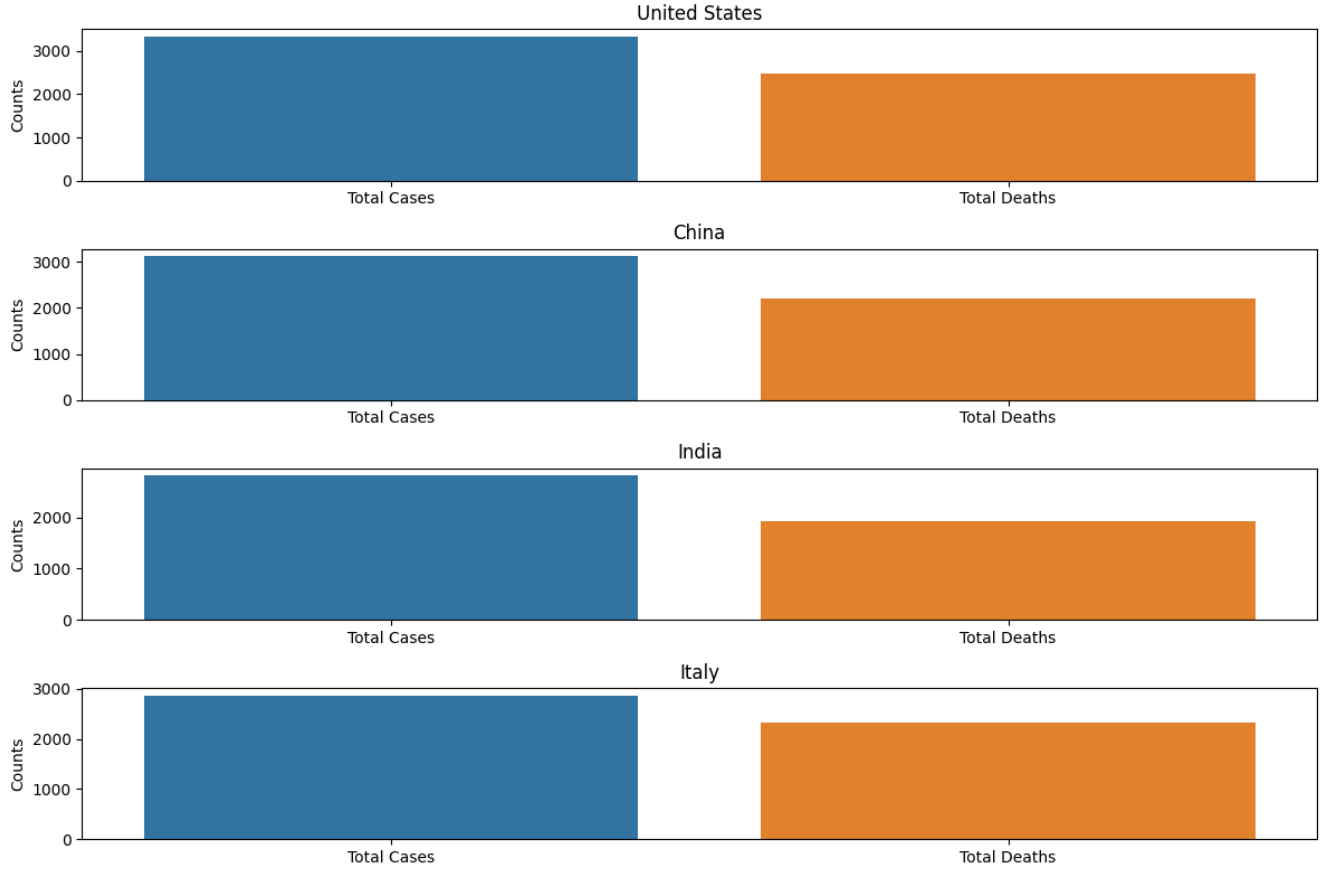
LINE PLOT OF COVID-19 DEATHS OVER TIME

In [63]:
```
df['DATE'] = pd.to_datetime(df['DATE'])

df['Month'] = df['DATE'].dt.month_name()

plt.figure(figsize=(10, 6))
sns.lineplot(x='Month', y='TD', data=df)
plt.xlabel('Month')
plt.ylabel('Total Deaths')
plt.title('COVID-19 Deaths Over Time (by Month)')
plt.xticks(rotation=45)
plt.show()
```



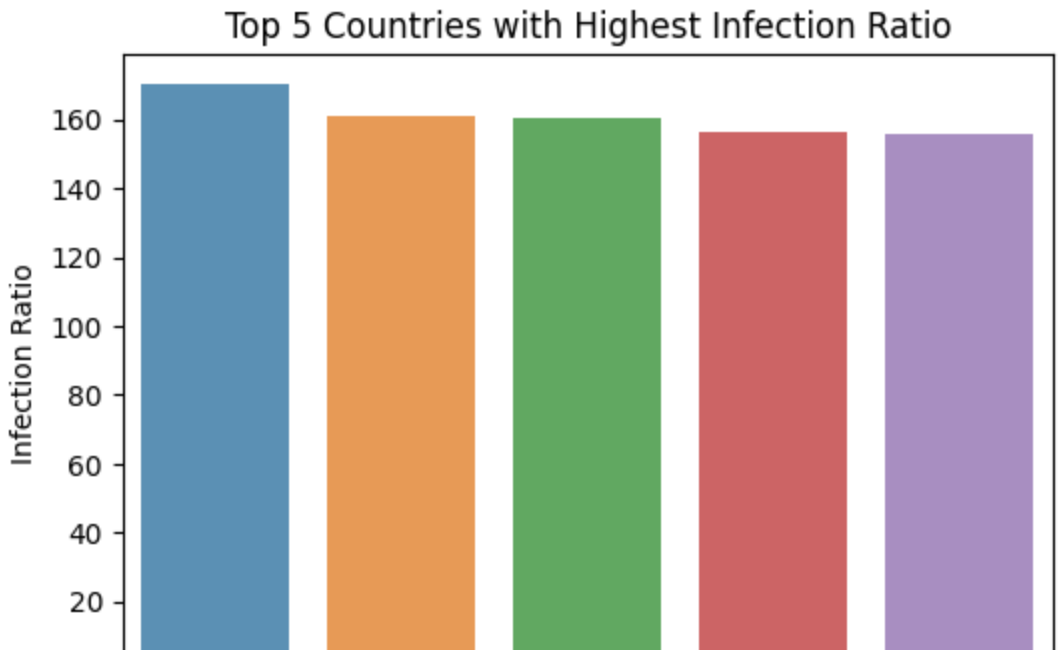
MOST POPULATED COUNTRIES CASES COMPARISON

In [20]:
```
countries = ['United States', 'China', 'India','Italy']

fig, axes = plt.subplots(len(countries), 1, figsize=(12, 8))

for i, c in enumerate(countries):
    country_data = df[df['COUNTRY'] == c]

    total_cases = country_data['TC'].sum()
    total_deaths = country_data['TD'].sum()

    sns.barplot(x=['Total Cases', 'Total Deaths'], y=[total_cases, total_deaths], ax=ax
    axes[i].set_title(c)
    axes[i].set_ylabel('Counts')

plt.tight_layout()

plt.show()
```



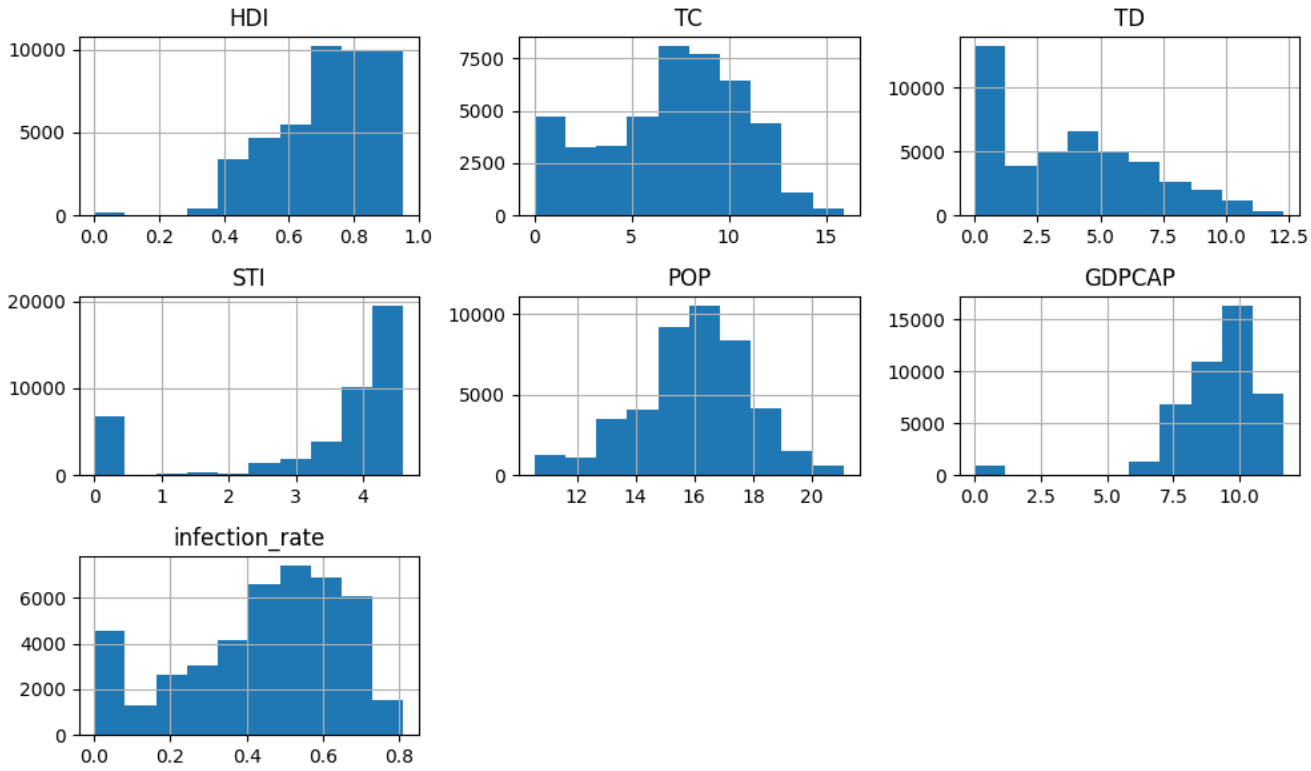
COUNTRIES WITH HIGHEST INFLECTION RATIO

In [95]:
```
Country_death = df.groupby("COUNTRY")["infection_rate"].sum().sort_values(ascending=Fal
plt.figure(figsize=(6, 4))
sns.barplot(x=Country_death.index, y=Country_death.values, alpha=0.8)

plt.xlabel('Country')
plt.ylabel('Infection Ratio')
plt.title(f'Top 5 Countries with Highest Infection Ratio')
plt.xticks(rotation=45)
plt.show()
```



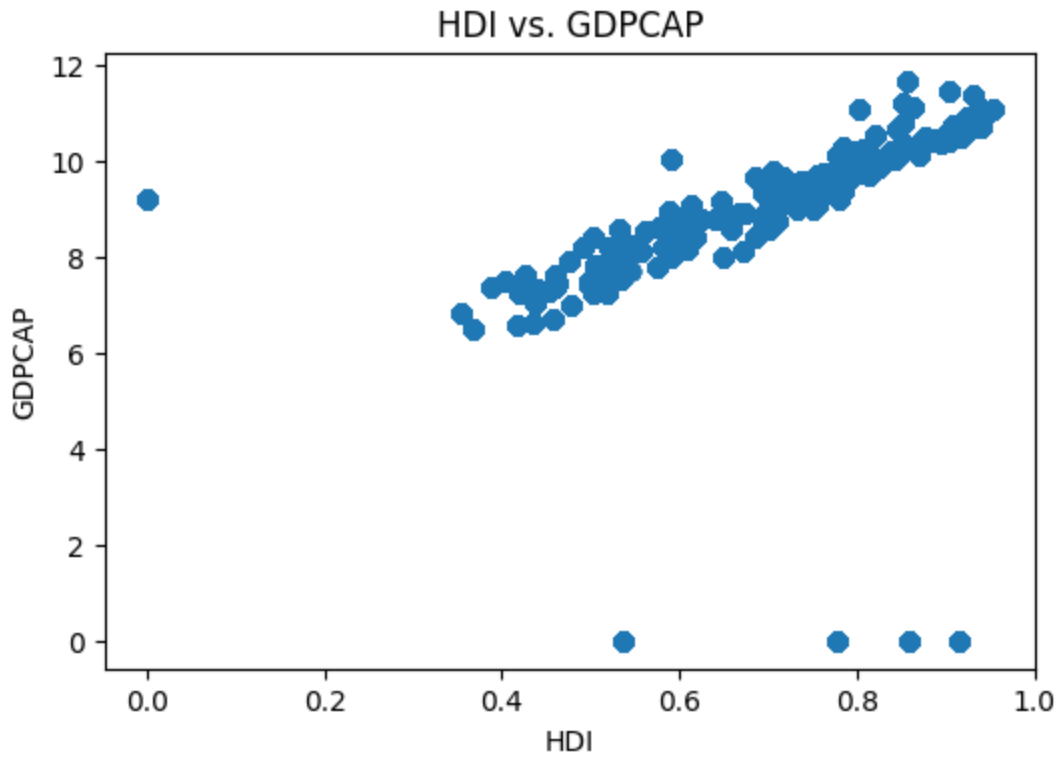
Visualising the distributions of the variables using histograms

In [25]:
```
df.hist(figsize=(10, 6))
plt.tight_layout()
plt.show()
```



VISUALISE THE RELATIONSHIP BETWEEN HDI AND GDPCAP

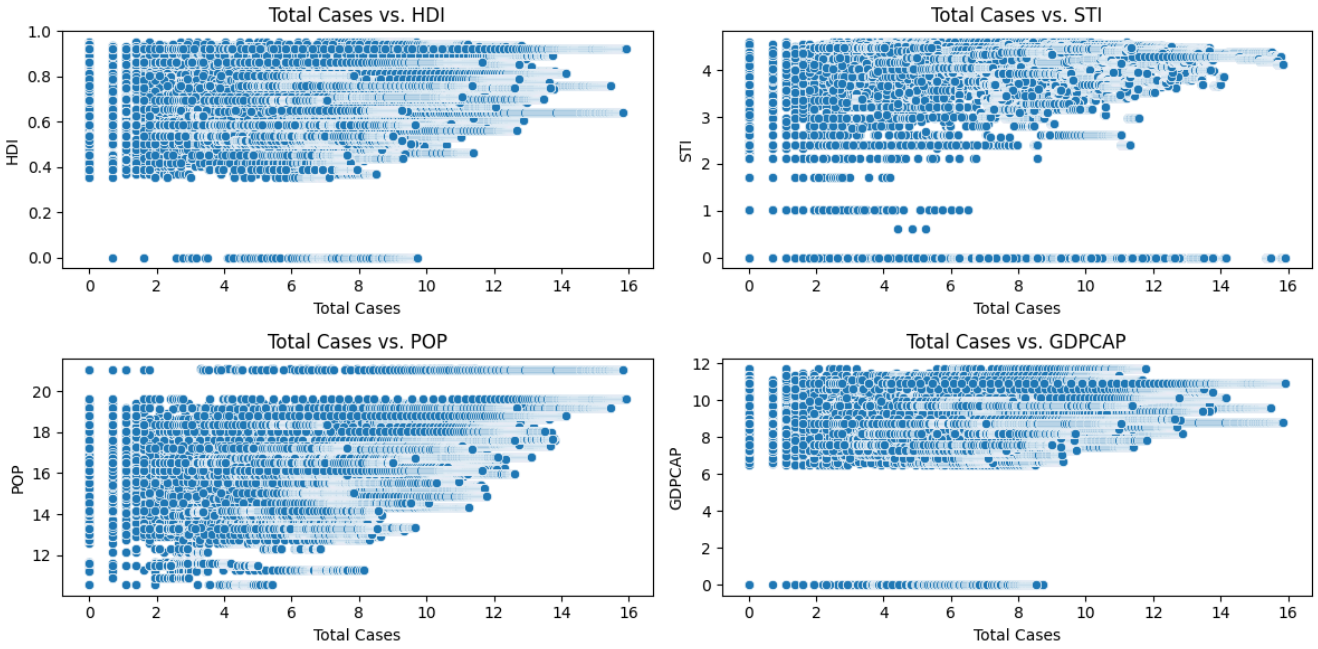
In [104]:
```
plt.figure(figsize=(6, 4))
plt.scatter(df['HDI'], df['GDPCAP'])
plt.xlabel('HDI')
plt.ylabel('GDPCAP')
plt.title('HDI vs. GDPCAP')
plt.show()
```



RELATIONSHIP BETWEEN TOTAL CASES AND DIFFERENT VARIABLES

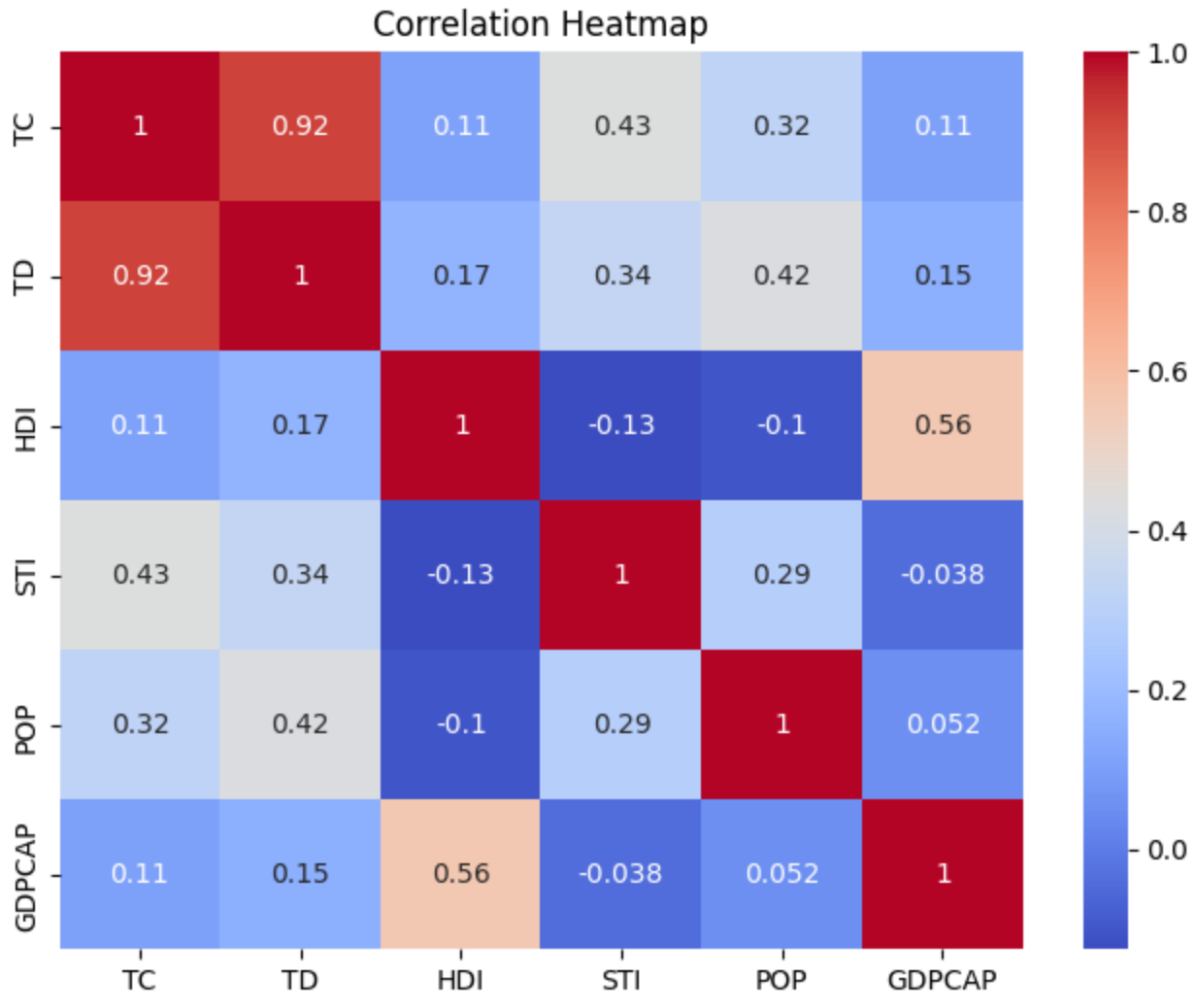
In [70]:
```
from scipy.stats import pearsonr
```

In [71]:
```
subset_df = df[['TC', 'TD', 'HDI', 'STI', 'POP', 'GDPCAP']]

plt.figure(figsize=(12, 6))

plt.subplot(2, 2, 1)
sns.scatterplot(data=df, x='TC', y='HDI')
plt.xlabel('Total Cases')
plt.ylabel('HDI')
plt.title('Total Cases vs. HDI')

plt.subplot(2, 2, 2)
sns.scatterplot(data=df, x='TC', y='STI')
plt.xlabel('Total Cases')
plt.ylabel('STI')
plt.title('Total Cases vs. STI')

plt.subplot(2, 2, 3)
sns.scatterplot(data=df, x='TC', y='POP')
plt.xlabel('Total Cases')
plt.ylabel('POP')
plt.title('Total Cases vs. POP')

plt.subplot(2, 2, 4)
sns.scatterplot(data=df, x='TC', y='GDPCAP')
plt.xlabel('Total Cases')
plt.ylabel('GDPCAP')
plt.title('Total Cases vs. GDPCAP')

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



VISUALISING CORRELATIONS USING HEATMAP

In [74]:
```
correlation_matrix = subset_df.corr()

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()

tc_hdi_corr, _ = pearsonr(df['TC'], df['HDI'])
tc_sti_corr, _ = pearsonr(df['TC'], df['STI'])
tc_pop_corr, _ = pearsonr(df['TC'], df['POP'])
tc_gdpcap_corr, _ = pearsonr(df['TC'], df['GDPCAP'])

td_hdi_corr, _ = pearsonr(df['TD'], df['HDI'])
td_sti_corr, _ = pearsonr(df['TD'], df['STI'])
td_pop_corr, _ = pearsonr(df['TD'], df['POP'])
td_gdpcap_corr, _ = pearsonr(df['TD'], df['GDPCAP'])

print(f"Correlation between Total Cases and HDI: {tc_hdi_corr:.3f}")
print(f"Correlation between Total Cases and STI: {tc_sti_corr:.3f}")
print(f"Correlation between Total Cases and POP: {tc_pop_corr:.3f}")
print(f"Correlation between Total Cases and GDPCAP: {tc_gdpcap_corr:.3f}")
print(f"Correlation between Total Deaths and HDI: {td_hdi_corr:.3f}")
print(f"Correlation between Total Deaths and STI: {td_sti_corr:.3f}")
print(f"Correlation between Total Deaths and POP: {td_pop_corr:.3f}")
print(f"Correlation between Total Deaths and GDPCAP: {td_gdpcap_corr:.3f}")
```



```
Correlation between Total Cases and HDI: 0.108
Correlation between Total Cases and STI: 0.433
Correlation between Total Cases and POP: 0.320
Correlation between Total Cases and GDPCAP: 0.106
Correlation between Total Deaths and HDI: 0.171
Correlation between Total Deaths and STI: 0.338
Correlation between Total Deaths and POP: 0.424
Correlation between Total Deaths and GDPCAP: 0.153
```

GDPCAP PERCENT CHANGE DURING COVID-19 PANDEMIC

In [75]:
```
df['GDPCAP_change'] = (df['GDPCAP'] - df['GDPCAP'].mean()) / df['GDPCAP'].mean() * 100
```

In [82]:
```
gdp_change = df.groupby('COUNTRY')['GDPCAP_change'].sum().sort_values(ascending=False)
print(gdp_change)
```

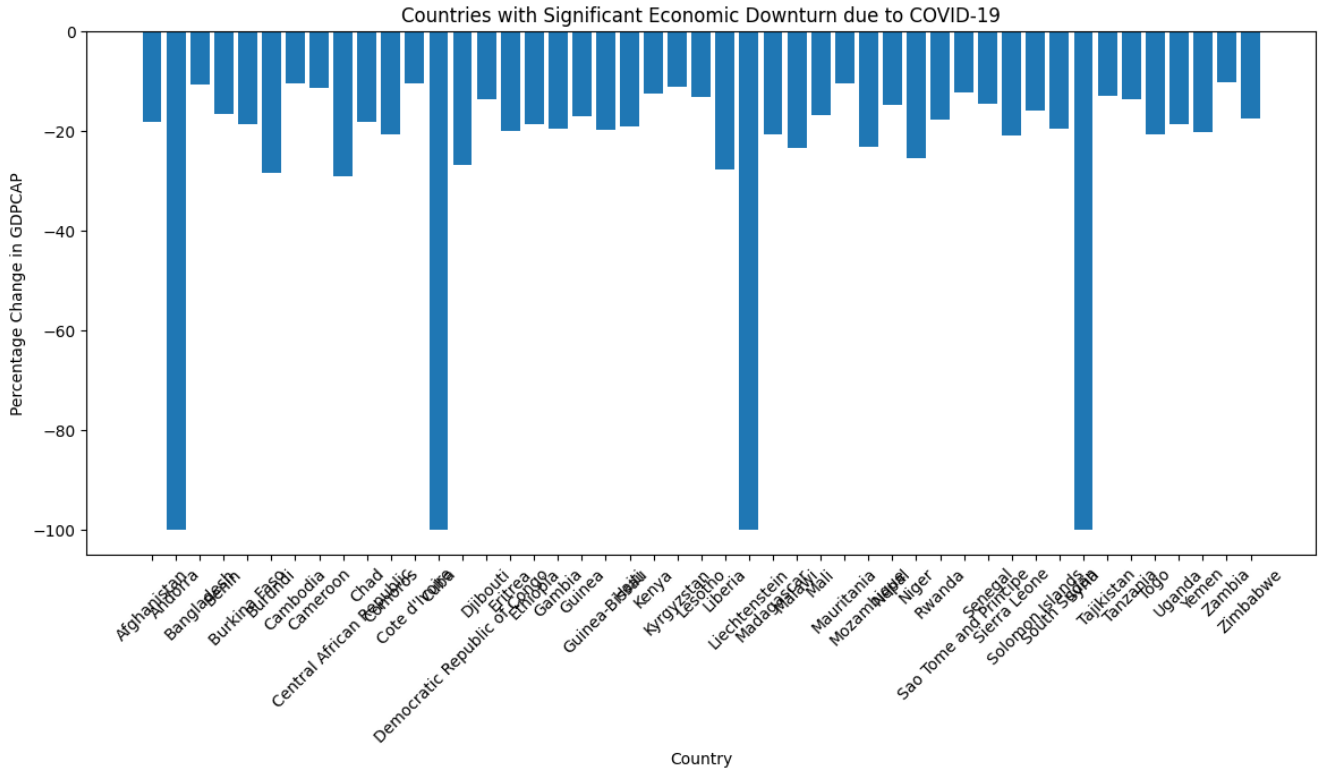
```
COUNTRY
Qatar                          8078.102994
Luxembourg                     7386.388472
Singapore                      7073.837979
Ireland                        6305.461850
United Arab Emirates           6303.467054
                                     ...
Central African Republic      -6334.672133
Syria                        -21100.000000
Cuba                         -21900.000000
Liechtenstein                -22400.000000
Andorra                      -22600.000000
Name: GDPCAP_change, Length: 182, dtype: float64
```

In [76]:
```
df.head()
```

Out[76]:

	CODE	COUNTRY	DATE	HDI	TC	TD	STI	POP	GDPCAP	infection_rate	Month	GDPCAP_
0	AFG	Afghanistan	2019-12-31	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	December	-18
1	AFG	Afghanistan	2020-01-01	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18
2	AFG	Afghanistan	2020-01-02	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18
3	AFG	Afghanistan	2020-01-03	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18
4	AFG	Afghanistan	2020-01-04	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18

In [91]:
```
threshold = -10  # Defining a threshold for percentage decrease in GDPCAP
downturn_countries = df[df['GDPCAP_change'] < threshold]

plt.figure(figsize=(14, 6))
plt.bar(downturn_countries['COUNTRY'], downturn_countries['GDPCAP_change'])
plt.xlabel('Country')
plt.ylabel('Percentage Change in GDPCAP')
plt.title('Countries with Significant Economic Downturn due to COVID-19')
plt.xticks(rotation=45)
plt.show()

summary_stats = downturn_countries[['COUNTRY', 'HDI', 'STI', 'POP', 'GDPCAP', 'GDPCAP_c
print(summary_stats)
```

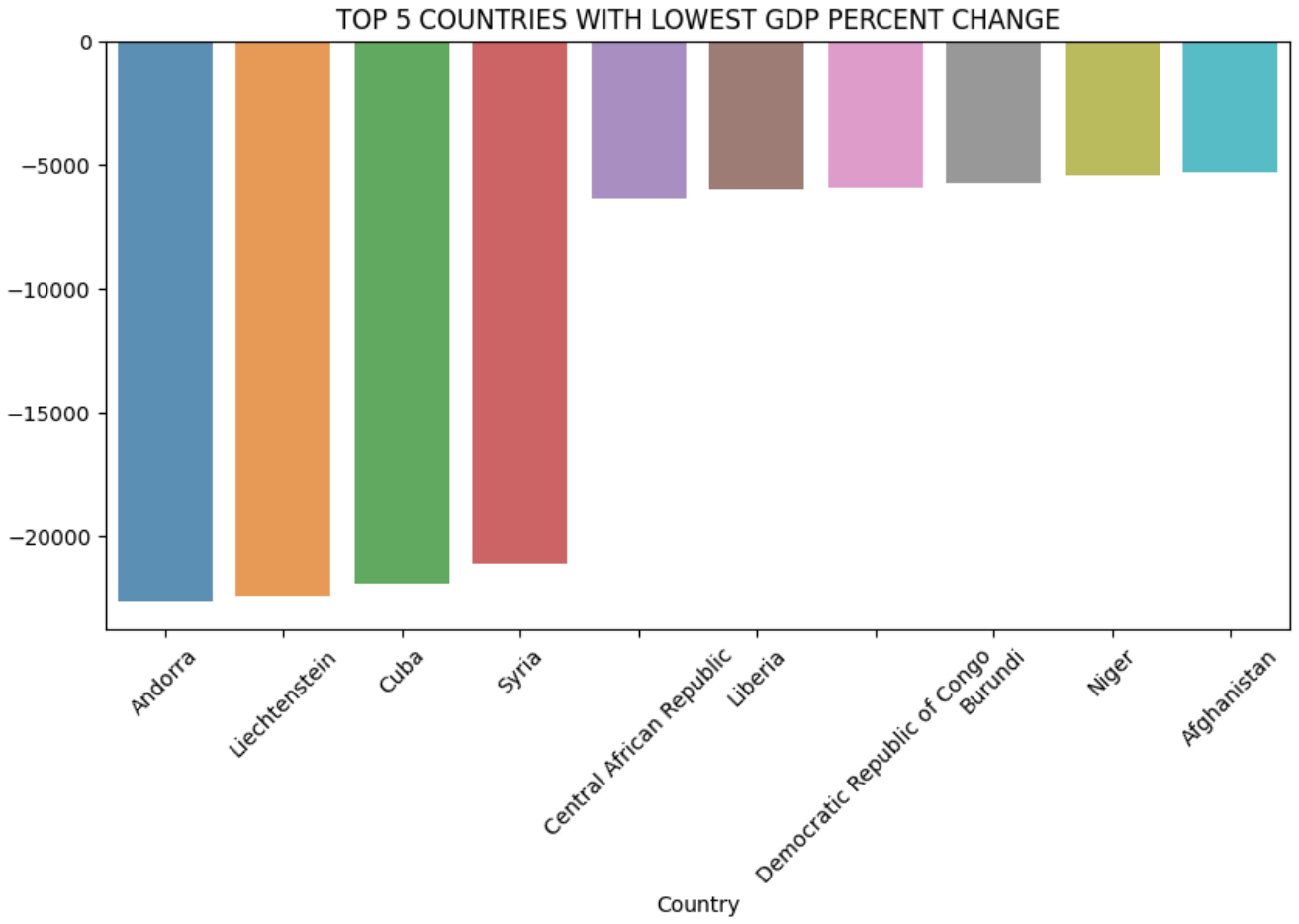


	COUNTRY	HDI	STI	POP	GDPCAP	GDPCAP_change
0	Afghanistan	0.498	0.000000	17.477233	7.497754	-18.094370
1	Afghanistan	0.498	0.000000	17.477233	7.497754	-18.094370
2	Afghanistan	0.498	0.000000	17.477233	7.497754	-18.094370
3	Afghanistan	0.498	0.000000	17.477233	7.497754	-18.094370
4	Afghanistan	0.498	0.000000	17.477233	7.497754	-18.094370
...	...	...	...	...	...	...
50413	Zimbabwe	0.535	4.341855	16.514381	7.549491	-17.529202
50414	Zimbabwe	0.535	4.341855	16.514381	7.549491	-17.529202
50415	Zimbabwe	0.535	4.341855	16.514381	7.549491	-17.529202
50416	Zimbabwe	0.535	4.341855	16.514381	7.549491	-17.529202
50417	Zimbabwe	0.535	4.341855	16.514381	7.549491	-17.529202

[9994 rows x 6 columns]

TOP 10 COUNTRIES WHOSE GDP SUFFERED THE MOST

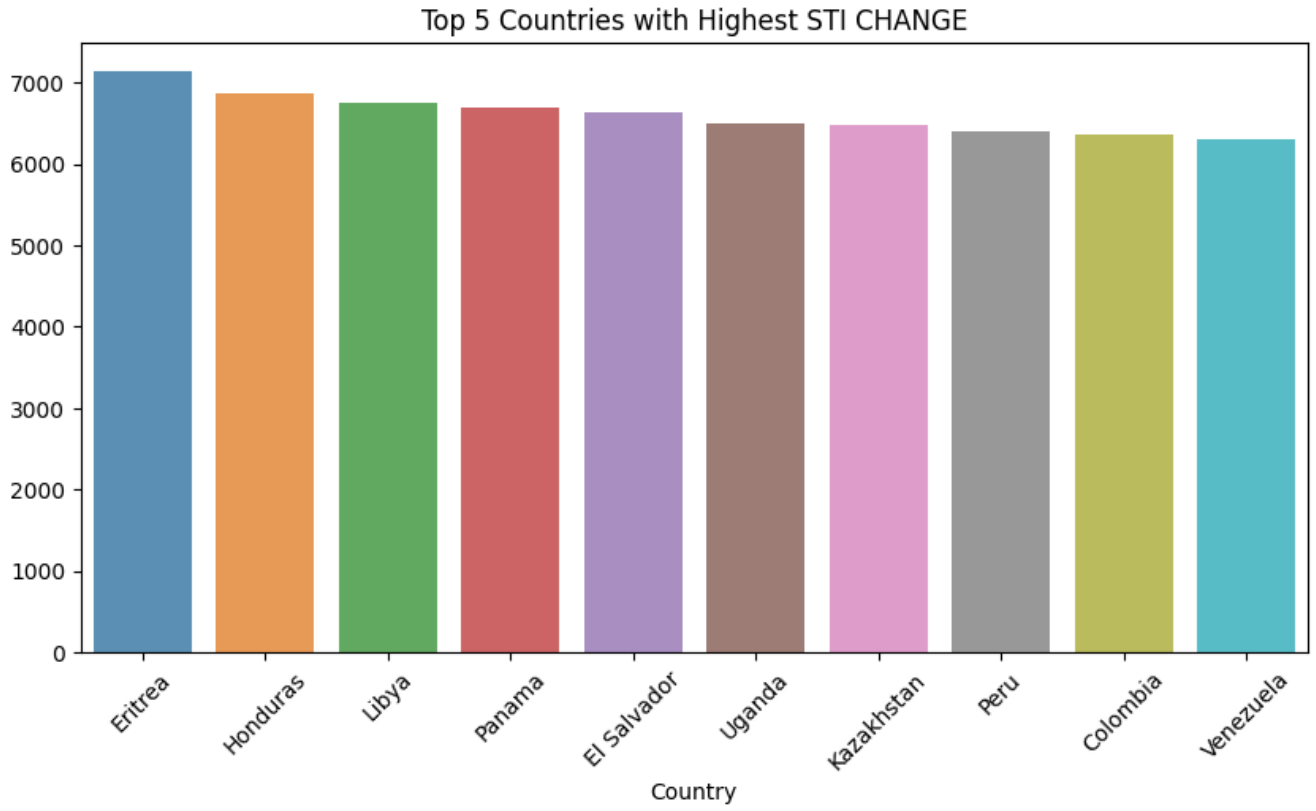
In [106]:
```
Country_death = df.groupby("COUNTRY")["GDPCAP_change"].sum().sort_values(ascending=True
plt.figure(figsize=(10, 5))
sns.barplot(x=Country_death.index, y=Country_death.values, alpha=0.8)

plt.xlabel('Country')
plt.ylabel('')
plt.title(f'TOP 5 COUNTRIES WITH LOWEST GDP PERCENT CHANGE')
plt.xticks(rotation=45)
plt.show()
```



COUNTRIES WHOSE STI INCREASED IN COVID TIMES

In [97]:
```
df['STI_change'] = (df['STI'] - df['STI'].mean()) / df['STI'].mean() * 100
```

In [105]:
```
Country_sti = df.groupby("COUNTRY")["STI_change"].sum().sort_values(ascending=False)[:1
plt.figure(figsize=(10, 5))
sns.barplot(x=Country_sti.index, y=Country_sti.values, alpha=0.8)

plt.xlabel('Country')
plt.ylabel('')
plt.title(f'Top 5 Countries with Highest STI CHANGE')
plt.xticks(rotation=45)
plt.show()
```



HDI PERCENT CHANGE DURING COVID-19 PANDEMIC

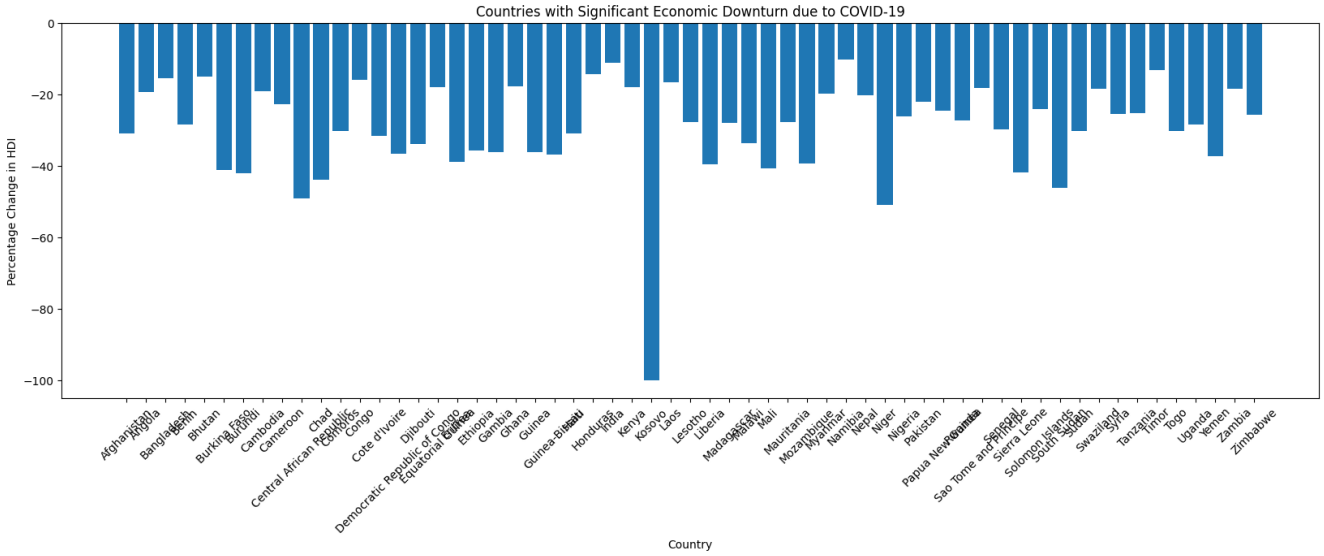
In [107]:
```
df['HDI_change'] = (df['HDI'] - df['HDI'].mean()) / df['HDI'].mean() * 100
```

In [108]:
```
df.head()
```

Out[108]:

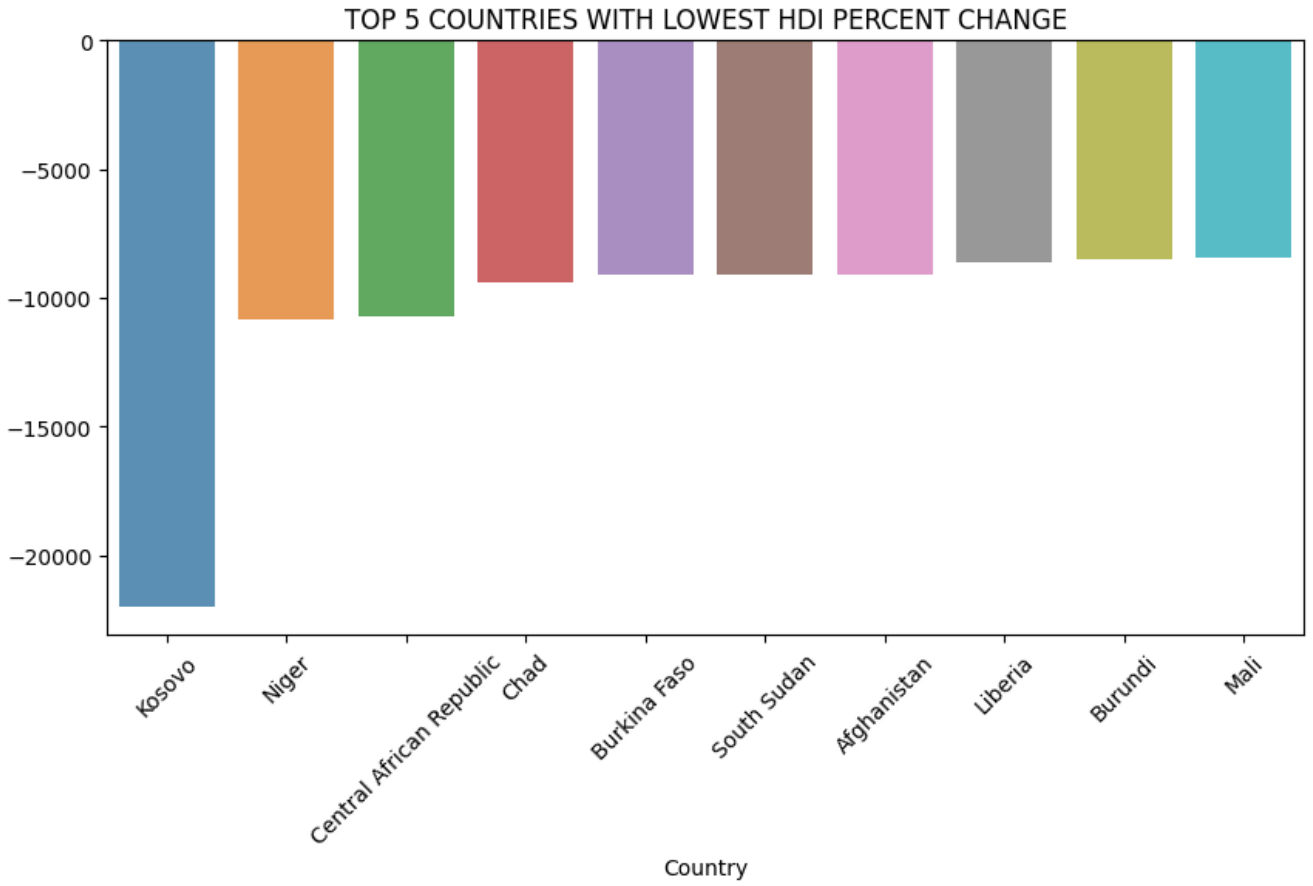
	CODE	COUNTRY	DATE	HDI	TC	TD	STI	POP	GDPCAP	infection_rate	Month	GDPCAP_
0	AFG	Afghanistan	2019-12-31	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	December	-18
1	AFG	Afghanistan	2020-01-01	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18
2	AFG	Afghanistan	2020-01-02	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18
3	AFG	Afghanistan	2020-01-03	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18
4	AFG	Afghanistan	2020-01-04	0.498	0.0	0.0	0.0	17.477233	7.497754	0.0	January	-18

In [112]:
```
threshold = -10  # Defining a threshold for percentage decrease in GDPCAP
downturn_countries = df[df['HDI_change'] < threshold]

plt.figure(figsize=(20, 6))
plt.bar(downturn_countries['COUNTRY'], downturn_countries['HDI_change'])
plt.xlabel('Country')
plt.ylabel('Percentage Change in HDI')
plt.title('Countries with Significant Economic Downturn due to COVID-19')
plt.xticks(rotation=45)
plt.show()
```



TOP 10 COUNTRIES WHOSE HDI SUFFERED THE MOST

In [110]:
```
Country_death = df.groupby("COUNTRY")["HDI_change"].sum().sort_values(ascending=True)[:
plt.figure(figsize=(10, 5))
sns.barplot(x=Country_death.index, y=Country_death.values, alpha=0.8)

plt.xlabel('Country')
plt.ylabel('')
plt.title(f'TOP 5 COUNTRIES WITH LOWEST HDI PERCENT CHANGE')
plt.xticks(rotation=45)
plt.show()
```



In [ ]:

In [ ]: