PROBLEM STATEMENT - Impact of Renewable Energy Adoption on Global Carbon Emissions

Project visualized by: GROUP 3

- Janani Karthikeyan
- Kruthika Srinivas Vasisht
- Sneha Manjunath Chakrabhavi

April 2nd, 2024

Presented to: Professor Satwik Kamarthi

The three type of Data Analysis to focus on in this project:

- Descriptive Analysis
- Exploratory Analysis
- Diagnostic Analysis

Citation: https://globalcarbonatlas.org/

Source: IRENA (2019) "The role of renewable energy in the global energy transformation" Energy Strategy Reviews, Volume 24 - April 2019

ABOUT THE DATASET:

The Global Carbon Atlas dataset offers extensive data on CO2 emissions from fossil fuel combustion and cement production, categorized by country and year. This dataset is instrumental for analyzing global and national trends in carbon emissions, providing a benchmark for assessing the impact of renewable energy adoption.

In [75]:
```
#import necessary libraries here
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Descriptive Analysis

"Descriptive Analysis involves summarizing and describing the main features of the dataset. It focuses on organizing and presenting the data in a meaningful way, often using measures such as mean, median, mode, and standard deviation. It provides an overview of the data and helps identify patterns or trends."

In [76]:
```
#Decriptive Analysis here
df = pd.read_csv("/content/export_emissions.csv")
```

```
df
```

Out[76]:

	Afghanistan	Albania	Algeria	Andorra	Angola	Anguilla	Antigua and Barbuda	Argentir
1960	0.41388	2.0222	6.1512	NaN	0.54895	NaN	0.036640	48.764
1961	0.4908	2.2785	6.0559	NaN	0.45371	NaN	0.047632	51.126
1962	0.68859	2.4617	5.6610	NaN	1.17910	NaN	0.102590	53.640
1963	0.70674	2.0806	5.4192	NaN	1.14970	NaN	0.084272	50.032
1964	0.83855	2.0147	5.6430	NaN	1.22290	NaN	0.091600	55.670
...	...	...	...	...	...	...	...	...
2021	12.2831	4.9037	180.2252	0.36305	17.50570	0.13770	0.596690	189.744
2022	12.1479	4.9547	176.3451	0.36865	16.07030	0.13897	0.602190	192.864
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
SOURCES	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
Territorial	<b>For sources and methods, see Andrew and Pet...	NaN	NaN	NaN	NaN	NaN	NaN	Na

66 rows × 221 columns

In [77]:

```
# Calculating mean, median, and standard deviation for each country over the availa
descriptive_stats = df.describe().transpose()[['mean', '50%', 'std']]
descriptive_stats.columns = ['Mean', 'Median', 'Standard Deviation']
descriptive_stats.head()
```

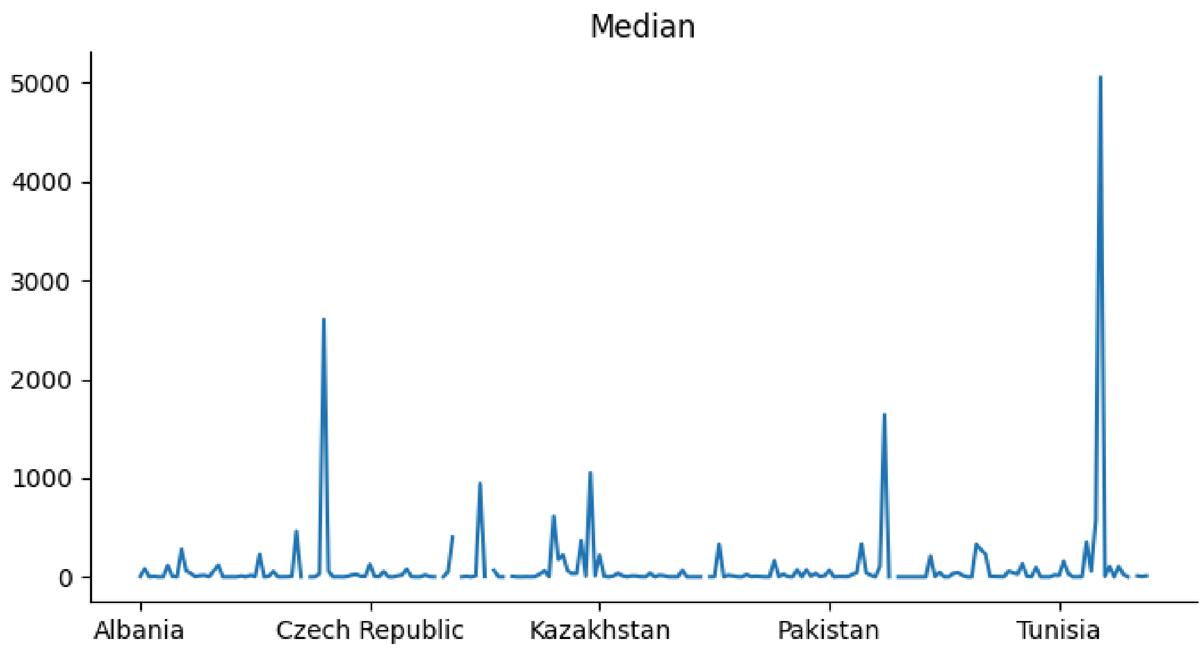
Out[77]:

	Mean	Median	Standard Deviation
Albania	4.509203	4.40660	1.869789
Algeria	79.739425	81.94400	51.798707
Andorra	0.475509	0.48365	0.058210
Angola	10.712060	5.48920	8.170389
Anguilla	0.100872	0.11725	0.034139

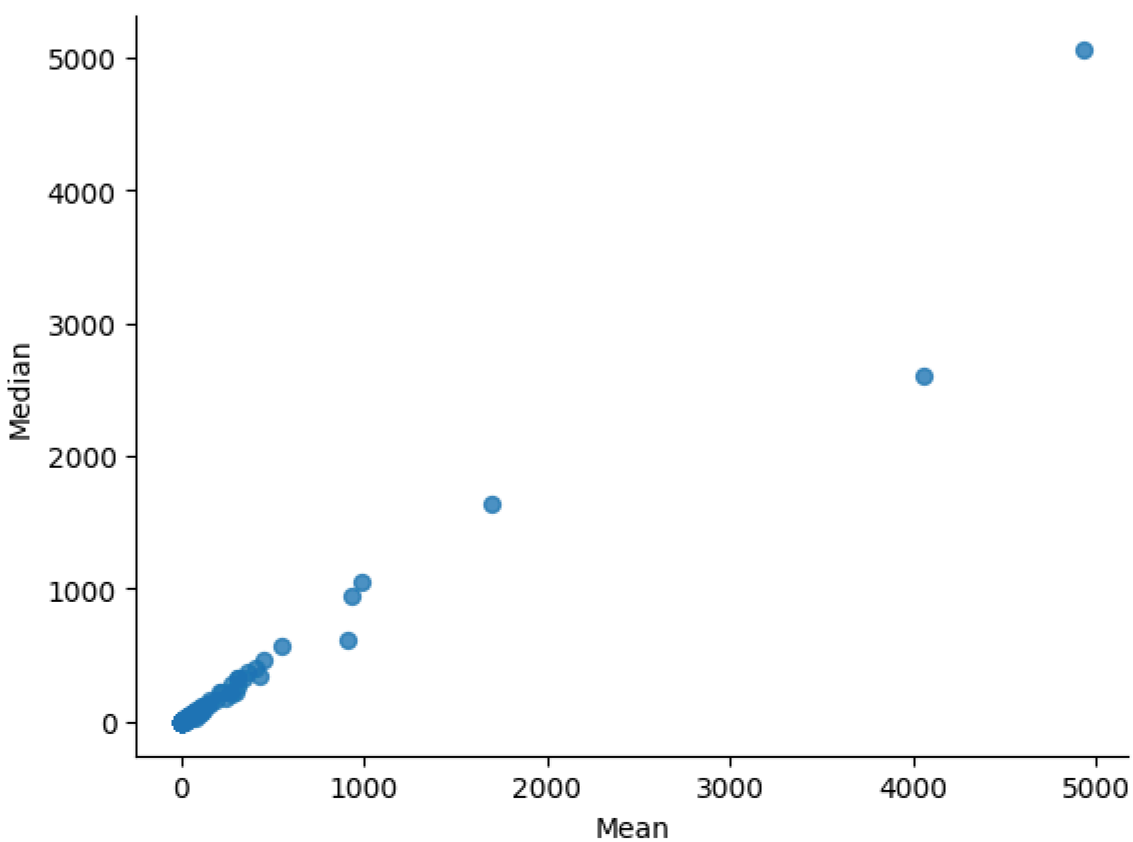
In [78]:

```
#Median
```

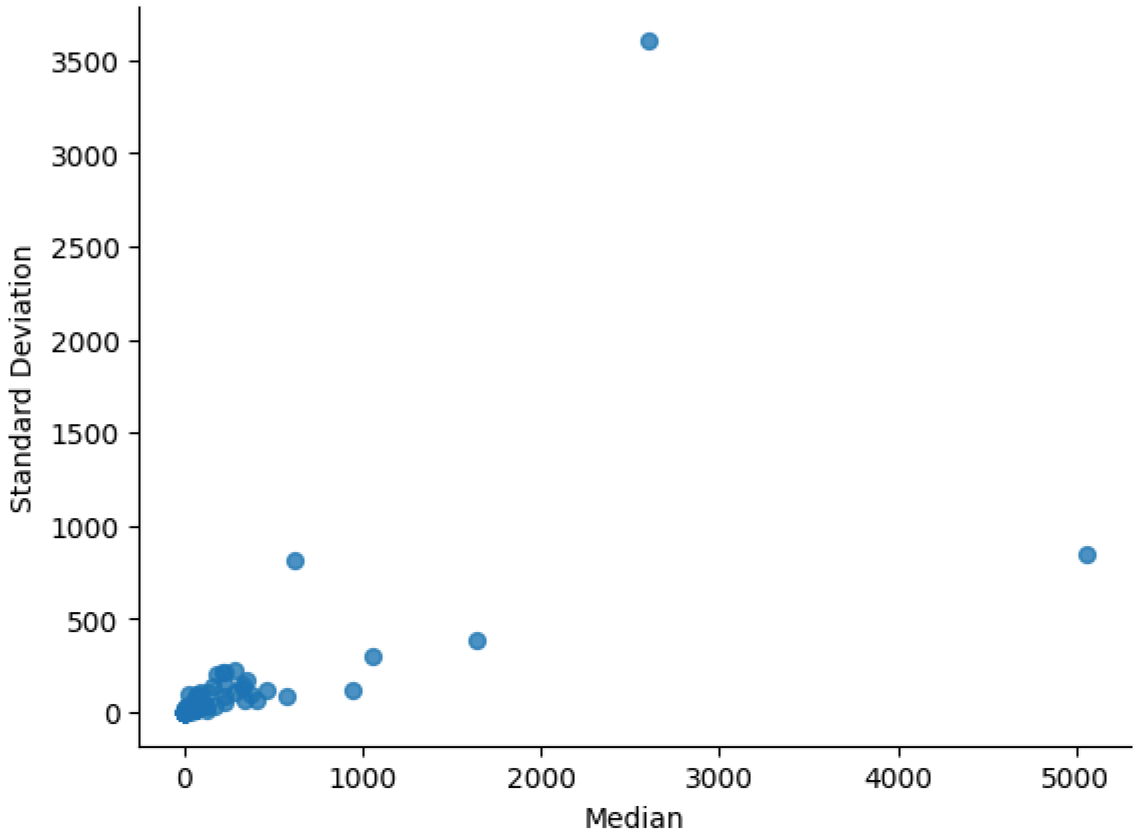
```
descriptive_stats['Median'].plot(kind='line', figsize=(8, 4), title='Median')
plt.gca().spines[['top', 'right']].set_visible(False)
```



In [79]:
```
# Mean vs Median

descriptive_stats.plot(kind='scatter', x='Mean', y='Median', s=32, alpha=.8)
plt.gca().spines[['top', 'right',]].set_visible(False)
```



In [80]:
```
# Median vs Standard Deviation

descriptive_stats.plot(kind='scatter', x='Median', y='Standard Deviation', s=32, al
plt.gca().spines[['top', 'right',]].set_visible(False)
```



In [81]:
```
descriptive_stats.describe()
```

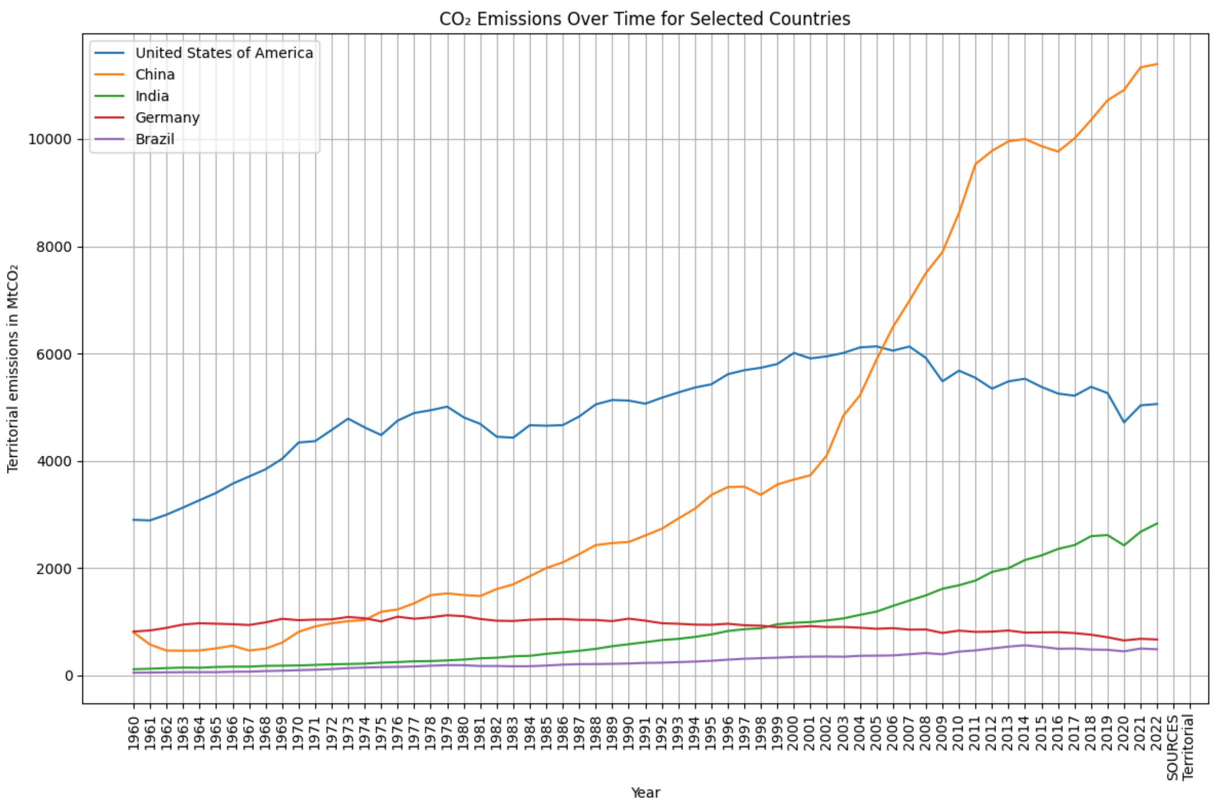
Out[81]:

	Mean	Median	Standard Deviation
count	212.000000	212.000000	212.000000
mean	107.274709	96.434433	48.470814
std	467.283419	421.257789	262.790864
min	0.005610	0.003664	0.002117
25%	0.790815	0.602530	0.526112
50%	5.900894	5.498550	3.545199
75%	44.457751	40.898050	19.051699
max	4930.969175	5057.303800	3604.971780

Exploratory Analysis

"Exploratory Analysis focuses on exploring and understanding the data without preconceived hypotheses. It involves visualizations, summary statistics, and data profiling techniques to uncover patterns, relationships, and interesting features. It helps generate hypotheses for further analysis"

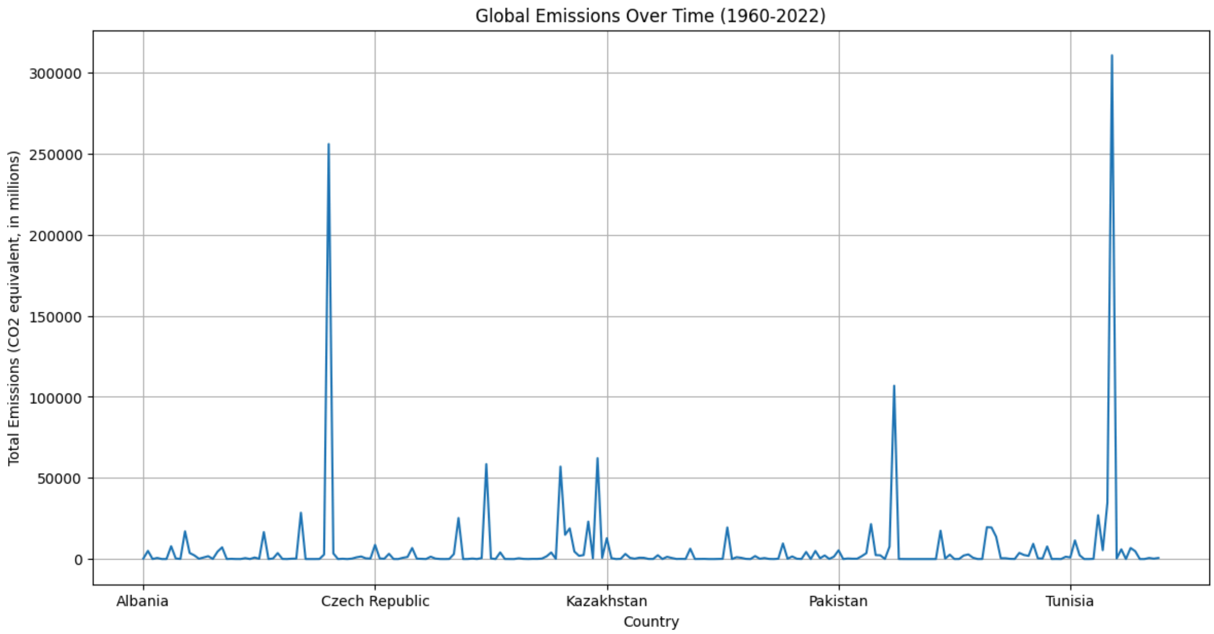
In [82]:
```
#Exploratory Analysis
countries_to_explore = ['United States of America', 'China', 'India', 'Germany', 'B
df_selected_countries = df[countries_to_explore]
plt.figure(figsize=(12, 8))
for country in countries_to_explore:
    sns.lineplot(data=df_selected_countries, x=df_selected_countries.index, y=count
plt.title('CO₂ Emissions Over Time for Selected Countries')
plt.xlabel('Year')
plt.ylabel('Territorial emissions in MtCO₂')  #MtCO2 means metric tons of CO2
plt.xticks(rotation=90)
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```



In [83]:
```
import matplotlib.pyplot as plt
import numpy as np
data_transposed = df.T
data_transposed.columns = data_transposed.iloc[0]
data_transposed = data_transposed.drop(data_transposed.index[0])
global_emissions = df.sum()
plt.figure(figsize=(14, 7))
global_emissions.plot()
plt.title('Global Emissions Over Time (1960-2022)')
plt.ylabel('Total Emissions (CO2 equivalent, in millions)')
```

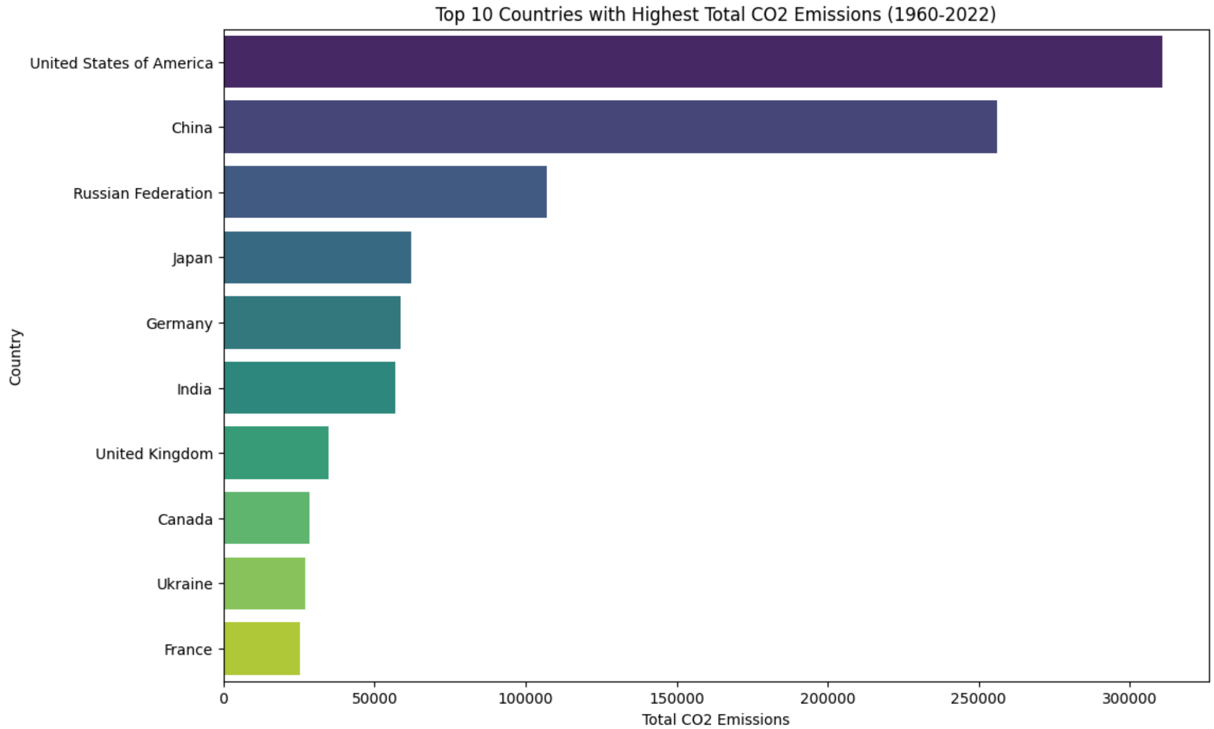
```
plt.xlabel('Country')
plt.grid(True)
plt.show()
```

```
<ipython-input-83-45eff187c8c8>:6: FutureWarning: The default value of numeric_only in DataFrame.sum is deprecated. In a future version, it will default to False. In addition, specifying 'numeric_only=None' is deprecated. Select only valid columns or specify the value of numeric_only to silence this warning.
  global_emissions = df.sum()
```



In [84]:
```
data = data.apply(pd.to_numeric, errors='coerce')
total_emissions = data.sum(axis=0).reset_index()
total_emissions.columns = ['Country', 'Total Emissions']
top_countries = total_emissions.sort_values(by='Total Emissions', ascending=False).
plt.figure(figsize=(12, 8))
sns.barplot(x='Total Emissions', y='Country', data=top_countries, palette='viridis'
plt.title('Top 10 Countries with Highest Total CO2 Emissions (1960-2022)')
plt.xlabel('Total CO2 Emissions')
plt.ylabel('Country')
plt.show()
```

```
<ipython-input-84-7967d745b7ba>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x='Total Emissions', y='Country', data=top_countries, palette='viridis')
```

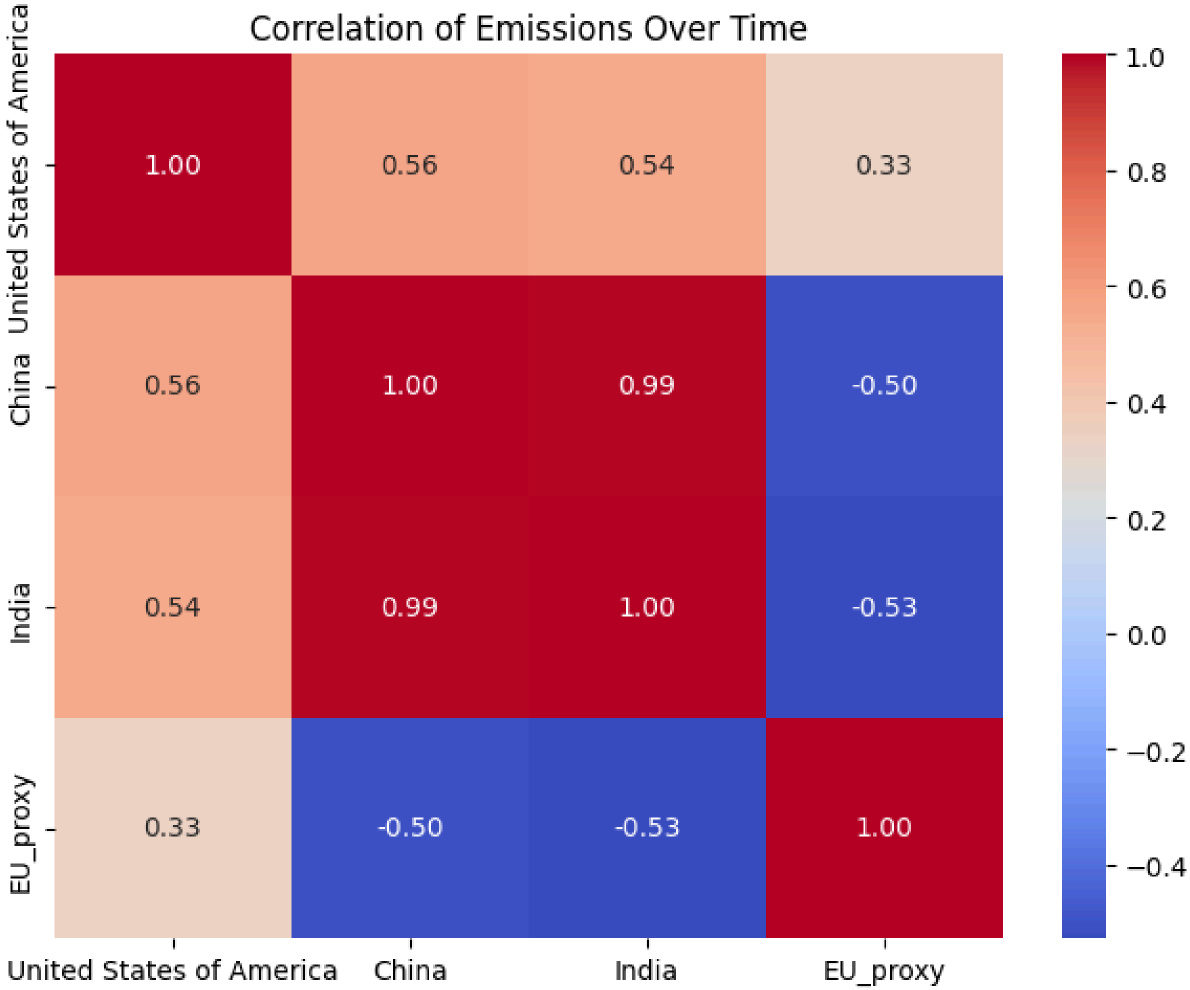


Diagnostic Analysis

"Diagnostic analysis aims to understand the cause-and-effect relationships within the data. It investigates the factors or variables that contribute to specific outcomes or behaviors. Techniques such as regression analysis or correlation analysis are commonly used."

In [85]:
```
# Calculate the EU proxy
# "EU proxy" is created by averaging the emissions data of several key EU member st
selected_countries = df[['United States of America', 'China', 'India', 'Germany', '
selected_countries['EU_proxy'] = selected_countries[['Germany', 'France', 'Italy']]
comparison_data = selected_countries[['United States of America', 'China', 'India',

# Correlation matrix
correlation_matrix = comparison_data.corr()

# Visualization
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.title('Correlation of Emissions Over Time')
plt.show()
```

```
<ipython-input-85-30410ad29b2f>:4: SettingWithCopyWarning:
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-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy
  selected_countries['EU_proxy'] = selected_countries[['Germany', 'France', 'Ital
y']].mean(axis=1)
```



In [86]:
```python
# Coverting columns to numeric, handling inconsistent data
df = pd.read_csv("/content/export_emissions.csv",index_col=0)
df = df.apply(pd.to_numeric, errors='coerce')
data_filled = df.ffill().bfill()
# Annual percentage change for each country
annual_percentage_change = data_filled.pct_change().multiply(100)
# Average annual percentage change for each country
average_annual_change = annual_percentage_change.mean()
# Countries with the most significant increases and decreases in emissions
sig_increases = average_annual_change.nlargest(5)
sig_decreases = average_annual_change.nsmallest(5)

print("Countries with the Most Significant Increases in Emissions:",sig_increases)
print("\nCountries with the Most Significant Decreases in Emissions:",sig_decreases
```

```
Countries with the Most Significant Increases in Emissions: Botswana
inf
British Virgin Islands    inf
Cook Islands              inf
Kiribati                  inf
Maldives                  inf
dtype: float64

Countries with the Most Significant Decreases in Emissions: Moldova          -0.9799
53
United Kingdom   -0.855311
Ukraine          -0.758988
Luxembourg       -0.389190
Latvia           -0.331255
dtype: float64
```

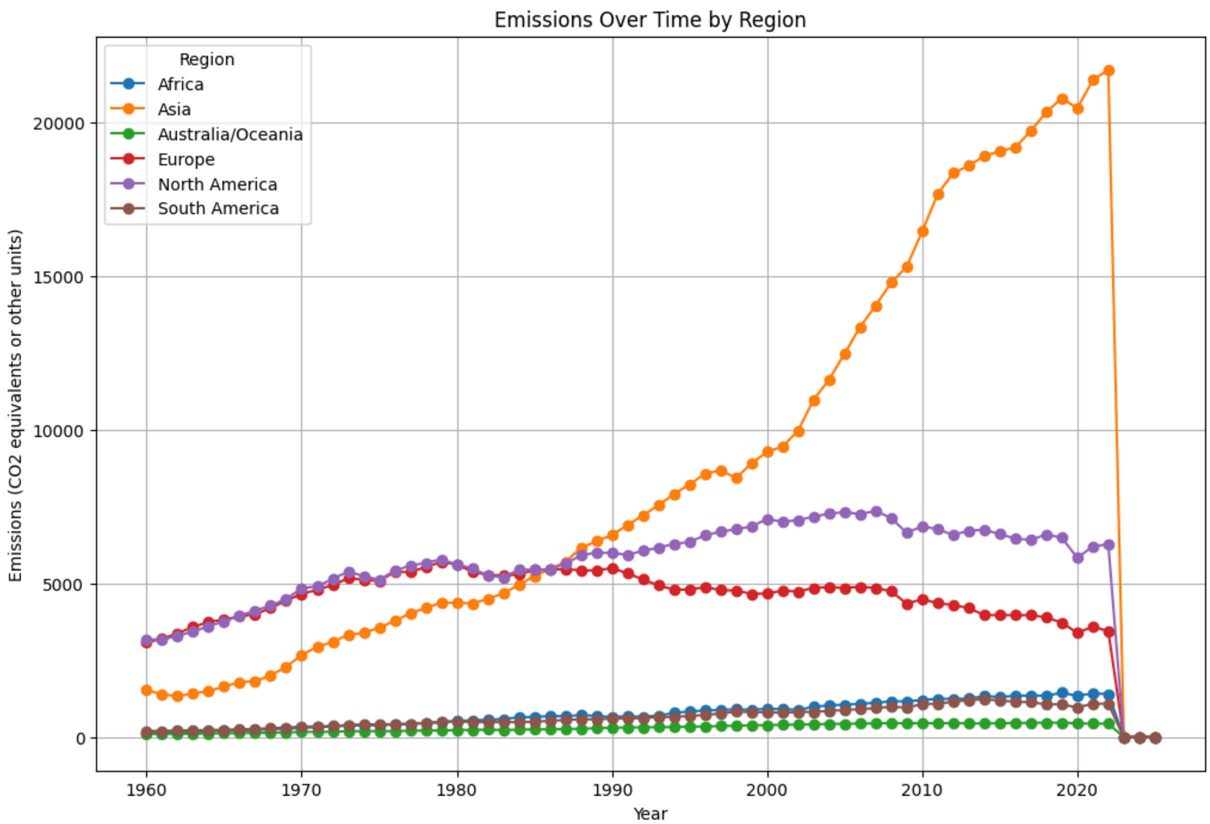
The output showing inf (infinity) for several countries suggests that these countries had years with zero emissions followed by years with non-zero emissions, leading to an infinite percentage increase calculation (any non-zero number divided by zero results in infinity in mathematical operations)

In [87]:
```python
# Replace inf/-inf with NaN to exclude them from average calculations
annual_percentage_change.replace([np.inf, -np.inf], np.nan, inplace=True)
average_annual_change = annual_percentage_change.mean()
sig_increases = average_annual_change.nlargest(5)
sig_decreases = average_annual_change.nsmallest(5)
print("Countries with the Most Significant Increases in Emissions:",sig_increases)
print("\nCountries with the Most Significant Decreases in Emissions:",sig_decreases
```

```
Countries with the Most Significant Increases in Emissions: United Arab Emirates
83.972102
Kosovo                 66.839137
Qatar                  53.285137
Equatorial Guinea      31.016393
Oman                   28.489997
dtype: float64

Countries with the Most Significant Decreases in Emissions: Moldova          -0.9799
53
United Kingdom   -0.855311
Ukraine          -0.758988
Luxembourg       -0.389190
Latvia           -0.331255
dtype: float64
```

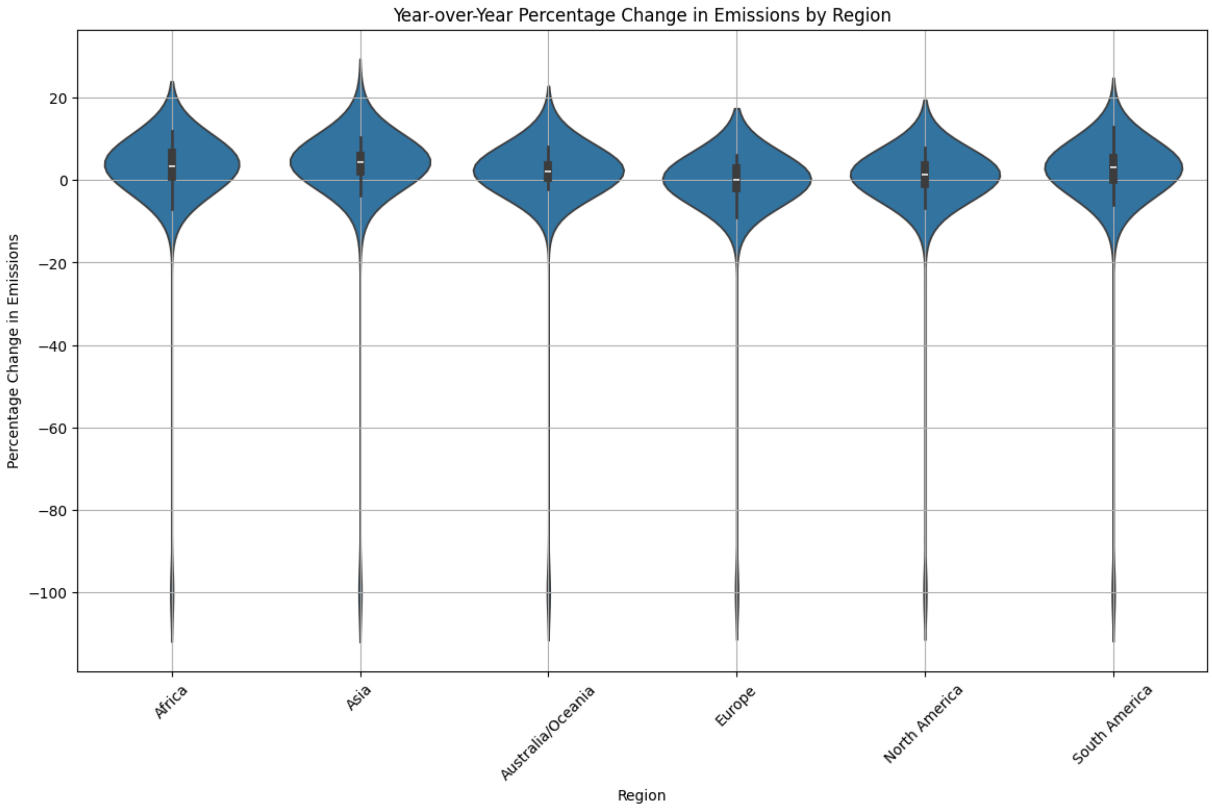
In [88]:
```python
inverted_mapping = {}
for region, countries in country_to_region.items():
    for country in countries:
        inverted_mapping[country] = region
data_transposed = data.transpose()

# Map country to its region
data_transposed['Region'] = data_transposed.index.map(lambda x: inverted_mapping.ge

# Drop rows where Region is NaN (countries not in the mapping)
```

```
data_transposed = data_transposed.dropna(subset=['Region'])

# Aggregate emissions by region, summing up for all years
region_sum = data_transposed.groupby('Region').sum()
region_sum = region_sum.drop(columns=['Region'], errors='ignore')

# Transpose back to have years as rows and regions as columns for analysis
region_sum_transposed = region_sum.transpose()

# Plot the trends for each region
region_sum_transposed.plot(figsize=(12, 8), marker='o')
plt.title('Emissions Over Time by Region')
plt.xlabel('Year')
plt.ylabel('Emissions (CO2 equivalents or other units)')
plt.legend(title='Region')
plt.grid(True)
plt.show()
```



In [89]:
```
# Percent change of emissions by region
percent_change_by_region = region_sum_transposed.pct_change().multiply(100)
percent_change_by_region = percent_change_by_region.reset_index()
violin_data = percent_change_by_region.melt(id_vars=['index'], var_name='Region', v

# Violin plot
plt.figure(figsize=(14, 8))
sns.violinplot(x='Region', y='Percent Change', data=violin_data)
plt.title('Year-over-Year Percentage Change in Emissions by Region')
plt.xlabel('Region')
plt.ylabel('Percentage Change in Emissions')
plt.xticks(rotation=45)
```

```
plt.grid(True)
plt.show()
```



Key Takeaways

1. **Descriptive Analysis Insights**: The descriptive statistics of the emissions data highlighted the variation in CO2 emissions across different countries, with significant differences in mean, median, and standard deviation values. This indicates a wide disparity in emissions levels, which could reflect differences in country sizes, industrial activity, energy consumption patterns, and the adoption of renewable energy sources.

2. **Trends in Selected Countries**: The exploratory analysis of CO2 emissions over time for selected countries (United States of America, China, India, Germany, Brazil) revealed distinct trends, underscoring the significant impact of national policies, economic development, and industrialization on emissions. Notably, countries like China and India have shown rapid increases in emissions, reflecting their economic growth and increasing energy needs.

3. **Global and Regional Emissions Trends**: Aggregated data showed a global increase in emissions over the studied period, while the regional analysis provided insights into how emissions have evolved differently across various regions. This variation could be influenced by regional policies on renewable energy, technological advancements, and shifts in industrial activities.

4. **Significant Increases and Decreases in Emissions**: Diagnostic analysis identified countries with the most significant increases and decreases in emissions, revealing how factors such as economic development, policy changes, and investment in renewable energy can dramatically influence a country's emissions profile. The appearance of "infinity" in the initial calculation highlights the challenge of measuring percentage changes from a zero base, necessitating careful data handling.

5. **Correlation of Emissions Among CO2 Emitters**: The correlation analysis among major emitters and the EU proxy indicated interlinked emissions trends, suggesting global economic activities, energy use patterns, and policy shifts can have widespread impacts beyond national borders.

6. **Variability in Regional Emissions Changes**: The violin plot visualization of year-over-year percentage change in emissions by region illustrated the variability and density of emissions changes across regions, highlighting the differing volatility in regional emissions. This could reflect the effectiveness of policies, the adoption rate of renewable energy technologies, and the resilience of regions to economic and environmental challenges.

These are the insights into the global dynamics of CO2 emissions, emphasizing the role of policy, technology, and economic factors in shaping emissions trends.

In [ ]:

```

```