Importing necesssary libraries

In [202…

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from scipy.stats import pearsonr, spearmanr
from sklearn.preprocessing import MinMaxScaler
```

Loading the datasets

In [203…

```
ghg_data = pd.read_csv('CO2_Emissions.csv')
temp_data = pd.read_csv('combined_temperature.csv')
```

Inspecting Data

In [204…

```
ghg_data.head(5)
```

Out[204]:

	Substance	EDGAR Country Code	Country	1970	1971	1972	1973	1974
0	CO2	ABW	Aruba	0.025214	0.028828	0.039472	0.044289	0.043469
1	CO2	AFG	Afghanistan	1.734053	1.733842	1.693672	1.733883	2.190254
2	CO2	AGO	Angola	8.948153	8.533779	10.383704	11.367327	11.827856
3	CO2	AIA	Anguilla	0.002178	0.002178	0.002273	0.002118	0.002360
4	CO2	AIR	International Aviation	168.602515	168.602515	178.806050	186.473958	179.454492

5 rows × 56 columns

In [205…

```
ghg_data.describe()
```

Out[205]:

	1970	1971	1972	1973	1974	1975	
count	212.000000	212.000000	212.000000	212.000000	212.000000	212.000000	212
mean	165.752027	165.496738	173.755583	183.853609	182.901639	182.006745	192
std	1160.922736	1156.625257	1215.246888	1284.799377	1276.815696	1269.564284	1343
min	0.000782	0.000808	0.000830	0.000849	0.000866	0.000881	0
25%	0.286510	0.286949	0.318343	0.343056	0.365620	0.385924	0
50%	3.387922	3.598651	3.848362	4.342237	4.418085	4.935483	5
75%	28.487826	29.131811	30.802970	33.876035	33.399006	35.772616	38
max	15825.423580	15791.976650	16595.630760	17571.709300	17501.566840	17450.413830	18448

8 rows × 53 columns

In [206…

```
ghg_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 214 entries, 0 to 213
Data columns (total 56 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Substance           212 non-null    object
 1   EDGAR Country Code  212 non-null    object
 2   Country             212 non-null    object
 3   1970                212 non-null    float64
 4   1971                212 non-null    float64
 5   1972                212 non-null    float64
 6   1973                212 non-null    float64
 7   1974                212 non-null    float64
 8   1975                212 non-null    float64
 9   1976                212 non-null    float64
 10  1977                212 non-null    float64
 11  1978                212 non-null    float64
 12  1979                212 non-null    float64
 13  1980                212 non-null    float64
 14  1981                212 non-null    float64
 15  1982                212 non-null    float64
 16  1983                212 non-null    float64
 17  1984                212 non-null    float64
 18  1985                212 non-null    float64
 19  1986                212 non-null    float64
 20  1987                212 non-null    float64
 21  1988                212 non-null    float64
 22  1989                212 non-null    float64
 23  1990                212 non-null    float64
 24  1991                212 non-null    float64
 25  1992                212 non-null    float64
 26  1993                212 non-null    float64
 27  1994                212 non-null    float64
 28  1995                212 non-null    float64
 29  1996                212 non-null    float64
 30  1997                212 non-null    float64
 31  1998                212 non-null    float64
 32  1999                212 non-null    float64
 33  2000                212 non-null    float64
 34  2001                212 non-null    float64
 35  2002                212 non-null    float64
 36  2003                212 non-null    float64
 37  2004                212 non-null    float64
 38  2005                212 non-null    float64
 39  2006                212 non-null    float64
 40  2007                212 non-null    float64
 41  2008                212 non-null    float64
 42  2009                212 non-null    float64
 43  2010                212 non-null    float64
 44  2011                212 non-null    float64
 45  2012                212 non-null    float64
 46  2013                212 non-null    float64
 47  2014                212 non-null    float64
 48  2015                212 non-null    float64
 49  2016                212 non-null    float64
 50  2017                212 non-null    float64
 51  2018                212 non-null    float64
 52  2019                212 non-null    float64
 53  2020                212 non-null    float64
 54  2021                212 non-null    float64
 55  2022                212 non-null    float64
dtypes: float64(53), object(3)
memory usage: 93.8+ KB
```

In [207…
```
temp_data.head(5)
```

Out[207]:

	Country	Year	Annual Mean	5-yr smooth	Code
0	Dominican Republic	1901	23.71	23.65	DOM
1	Dominican Republic	1902	23.40	23.52	DOM
2	Dominican Republic	1903	23.45	23.38	DOM
3	Dominican Republic	1904	23.03	23.26	DOM
4	Dominican Republic	1905	23.44	23.16	DOM

In [208…
```
temp_data.describe()
```

Out[208]:

	Year	Annual Mean	5-yr smooth
count	20740.000000	20740.000000	20740.000000
mean	1961.500000	18.938327	18.938337
std	35.218032	8.310990	8.303583
min	1901.000000	-6.810000	-5.770000
25%	1931.000000	11.270000	11.280000
50%	1961.500000	22.400000	22.355000
75%	1992.000000	25.720000	25.710000
max	2022.000000	29.780000	29.420000

In [209…
```
temp_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20740 entries, 0 to 20739
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   Country      20740 non-null  object 
 1   Year         20740 non-null  int64  
 2   Annual Mean  20740 non-null  float64
 3   5-yr smooth  20740 non-null  float64
 4   Code         20740 non-null  object 
dtypes: float64(2), int64(1), object(2)
memory usage: 810.3+ KB
```

Data preparation

In [210…
```
ghg_data = pd.melt(ghg_data, id_vars=['Country', 'Substance', 'EDGAR Country Code']
                   var_name='Year', value_name='GHG_Emissions')
```

In [211…
```
ghg_data.head(5)
```

Out[211]:

	Country	Substance	EDGAR Country Code	Year	GHG_Emissions
0	Aruba	CO2	ABW	1970	0.025214
1	Afghanistan	CO2	AFG	1970	1.734053
2	Angola	CO2	AGO	1970	8.948153
3	Anguilla	CO2	AIA	1970	0.002178
4	International Aviation	CO2	AIR	1970	168.602515

In [212…

```
ghg_data.shape
```

Out[212]:

```
(11342, 5)
```

Handling missing Values

In [213…

```
ghg_data.isnull().sum()
```

Out[213]:

```
Country               106
Substance             106
EDGAR Country Code    106
Year                    0
GHG_Emissions         106
dtype: int64
```

In [214…

```
ghg_data.dropna(inplace = True)
```

In [215…

```
temp_data.isnull().sum()
```

Out[215]:

```
Country        0
Year           0
Annual Mean    0
5-yr smooth    0
Code           0
dtype: int64
```

In [216…

```
temp_data.shape
```

Out[216]:

```
(20740, 5)
```

Converting data types

In [217…

```
ghg_data.dtypes
```

Out[217]:

```
Country                object
Substance              object
EDGAR Country Code     object
Year                   object
GHG_Emissions         float64
dtype: object
```

In [218…

```
temp_data.dtypes
```

Out[218]:
```
Country         object
Year             int64
Annual Mean    float64
5-yr smooth    float64
Code            object
dtype: object
```

In [219…
```
ghg_data['Year'] = pd.to_numeric(ghg_data['Year'])
```

In [220…
```
ghg_data.isnull().sum()
```

Out[220]:
```
Country               0
Substance             0
EDGAR Country Code    0
Year                  0
GHG_Emissions         0
dtype: int64
```

Merging the dataframes

In [221…
```
merged_data = pd.merge(ghg_data, temp_data, on=['Country', 'Year'])
```

In [222…
```
merged_data.head()
```

Out[222]:

	Country	Substance	EDGAR Country Code	Year	GHG_Emissions	Annual Mean	5-yr smooth	Code
0	Afghanistan	CO2	AFG	1970	1.734053	13.17	12.45	AFG
1	Angola	CO2	AGO	1970	8.948153	21.51	21.43	AGO
2	Albania	CO2	ALB	1970	4.848780	11.61	11.56	ALB
3	United Arab Emirates	CO2	ARE	1970	19.429211	27.80	27.34	ARE
4	Argentina	CO2	ARG	1970	87.006875	15.09	14.91	ARG

Renaming the columns

In [223…
```
merged_data = merged_data.rename(columns={'Annual Mean':'Mean_Temperature'})
```

In [224…
```
merged_data.head()
```

Out[224]:

	Country	Substance	EDGAR Country Code	Year	GHG_Emissions	Mean_Temperature	5-yr smooth	Code
0	Afghanistan	CO2	AFG	1970	1.734053	13.17	12.45	AFG
1	Angola	CO2	AGO	1970	8.948153	21.51	21.43	AGO
2	Albania	CO2	ALB	1970	4.848780	11.61	11.56	ALB
3	United Arab Emirates	CO2	ARE	1970	19.429211	27.80	27.34	ARE
4	Argentina	CO2	ARG	1970	87.006875	15.09	14.91	ARG

In [225…
```
merged_data.shape
```

Out[225]:
```
(7261, 8)
```

In [226…
```
merged_data.isnull().sum()
```

Out[226]:
```
Country               0
Substance             0
EDGAR Country Code    0
Year                  0
GHG_Emissions         0
Mean_Temperature      0
5-yr smooth           0
Code                  0
dtype: int64
```

Dropping unnecesary columns

In [227…
```
merged_data.drop(['Substance','EDGAR Country Code', '5-yr smooth'], inplace=True, a
```

In [228…
```
merged_data.head()
```

Out[228]:

	Country	Year	GHG_Emissions	Mean_Temperature	Code
0	Afghanistan	1970	1.734053	13.17	AFG
1	Angola	1970	8.948153	21.51	AGO
2	Albania	1970	4.848780	11.61	ALB
3	United Arab Emirates	1970	19.429211	27.80	ARE
4	Argentina	1970	87.006875	15.09	ARG

In [229…
```
merged_data.describe()
```

Out[229]:

	Year	GHG_Emissions	Mean_Temperature
count	7261.000000	7261.000000	7261.000000
mean	1996.000000	104.900315	19.608873
std	15.298112	461.098080	7.989158
min	1970.000000	0.007584	-6.810000
25%	1983.000000	1.443863	12.330000
50%	1996.000000	9.019200	22.870000
75%	2009.000000	55.392762	25.940000
max	2022.000000	5929.188398	29.780000

Exploratory Data Analysis

Calculation of mean GHG Emissions

In [230…
```
mean_emissions_by_year = merged_data.groupby('Year')['GHG_Emissions'].mean().reset_
```

In [231…
```
mean_emissions_by_year.head()
```

Out[231]:

	Year	GHG_Emissions
0	1970	80.281293
1	1971	79.588737
2	1972	83.602262
3	1973	88.494256
4	1974	87.399182

Calculation of mean Temperature

In [232…
```
mean_temperature_by_year = merged_data.groupby('Year')['Mean_Temperature'].mean().r
```

In [233…
```
mean_temperature_by_year.head()
```

Out[233]:

	Year	Mean_Temperature
0	1970	19.189124
1	1971	18.963066
2	1972	19.019708
3	1973	19.220949
4	1974	18.973431

Combining the mean data

In [234…
```
combined_data = pd.merge(mean_emissions_by_year, mean_temperature_by_year, on='Year
```

In [235…
```
combined_data.head()
```

Out[235]:

	Year	GHG_Emissions	Mean_Temperature
0	1970	80.281293	19.189124
1	1971	79.588737	18.963066
2	1972	83.602262	19.019708
3	1973	88.494256	19.220949
4	1974	87.399182	18.973431

Normalizing the data

In [236…
```
scaler = MinMaxScaler()
combined_data[['Normalized_GHG_Emissions', 'Normalized_Mean_Temperature']] = scaler
```

```
    combined_data[['GHG_Emissions', 'Mean_Temperature']]
)
```

Plotting of the relationship between mean GHG emissions and mean temperature over time

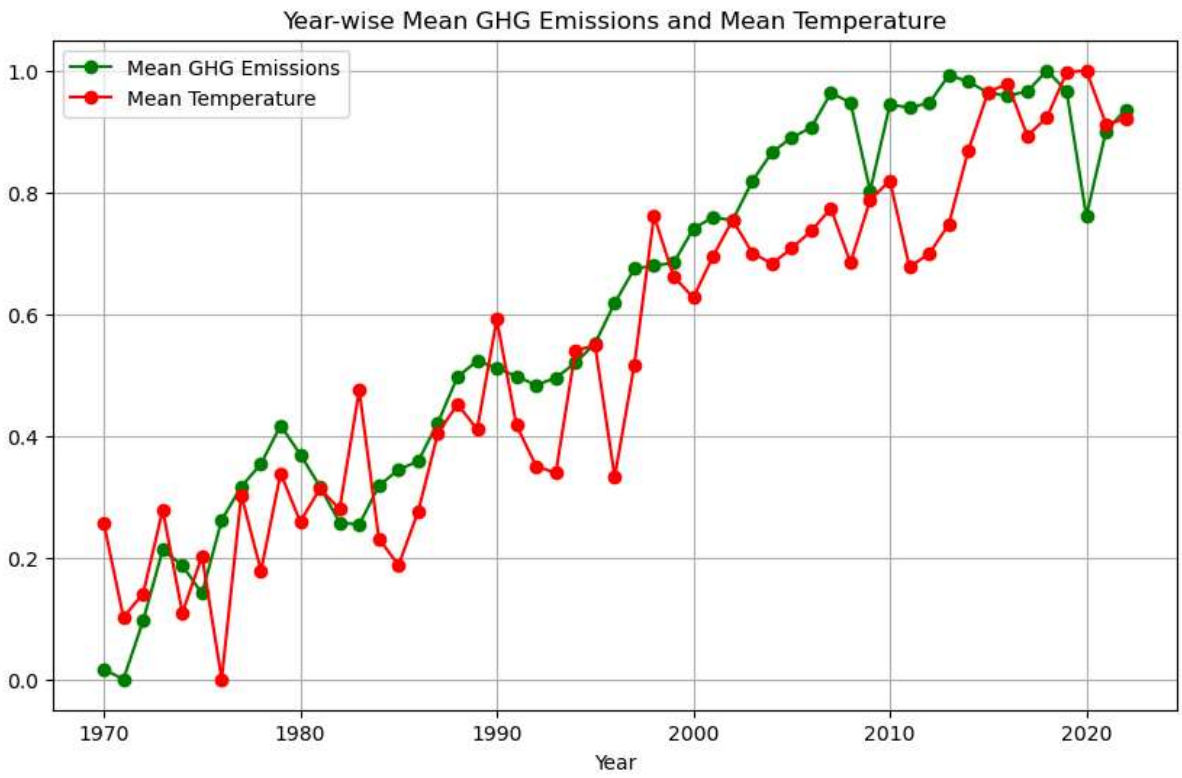
In [237…

```
# Plot the data
plt.figure(figsize=(10, 6))

# Plot normalized GHG emissions
plt.plot(combined_data['Year'], combined_data['Normalized_GHG_Emissions'], marker='

# Plot normalized temperature
plt.plot(combined_data['Year'], combined_data['Normalized_Mean_Temperature'], marke

plt.xlabel('Year')
plt.title('Year-wise Mean GHG Emissions and Mean Temperature')
plt.grid(True)
plt.legend()
plt.show()
```



Pearson Correlation Test

Hypotheses:

- **Null hypothesis:** There is no correlation between GHG emissions and mean temperature(correlation coefficient r = 0)
- **Alternative hypothesis:** There is a linear correlation between GHG emissions and mean temperature(correlation coefficient r != 0) ## Test statistic: The Pearson correlation

coefficient(r) measures the strength and direction of the linear relationship between two variables.

P-value: The p-value indicates the probability of observing a correlation as extreme as the one in the sample data, assuming the null hypothesis is true.

In [238…

```
# Pearson correlation
pearson_corr, pearson_p_value = pearsonr(combined_data['GHG_Emissions'], combined_d
print(f'Pearson Correlation Coefficient: {pearson_corr}')
print(f'Pearson P-value: {pearson_p_value}')
```

```
Pearson Correlation Coefficient: 0.9118557739084416
Pearson P-value: 2.3120841321506484e-21
```

Result

Pearson Correlation Coefficient: 0.9118557739084416

Pearson p-value: 2.3120841321506484e-21

Interpretation:

- The p-value is extremely small (much smaller than the typical significance level of 0.05)
- The correlation coefficient value indicates a strong positive linear relationship between GHG emissions and mean temperature. ## Conclusion:
- Reject the null hypothesis.
- There is strong evidence of a significant linear correlation between GHG emissions and mean temperature

Spearman Correlation Test

Hypotheses:

- **Null hypothesis:** There is no correlation between GHG emissions and mean temperature(correlation coefficient ρ = 0)
- **Alternative hypothesis:** There is a linear correlation between GHG emissions and mean temperature(correlation coefficient ρ != 0) ## Test Statistic:
- The Spearman correlation coefficient(p) measures the strength and direction of the monotonic relationship between two variables

P-value: The p-value indicates the probability of observing a correlation as extreme as the one in the sample data, assuming the null hypothesis is true

In [239…

```
# Spearman correlation
spearman_corr, spearman_p_value = spearmanr(combined_data['GHG_Emissions'], combine
```

```
print(f'Spearman Correlation Coefficient: {spearman_corr}')
print(f'Spearman P-value: {spearman_p_value}')
```

```
Spearman Correlation Coefficient: 0.9063054346073213
Spearman P-value: 1.0247373867415534e-20
```

Result

Spearman Correlation Coefficient: 0.9063054346073213

Spearman P-value: 1.0247373867415534e-20

Interpretation:

- The p-value is extremely small (much smaller than the typical significance level of 0.05)
- The correlation coefficient value indicates a strong positive monotonic relationship between GHG emissions and mean temperature. ## Conclusion:
- Reject the null hypothesis.
- There is strong evidence of a significant positive monotonic correlation between GHG emissions and mean temperature

Calculation of Correlation Coefficient of a particular country

In [240…

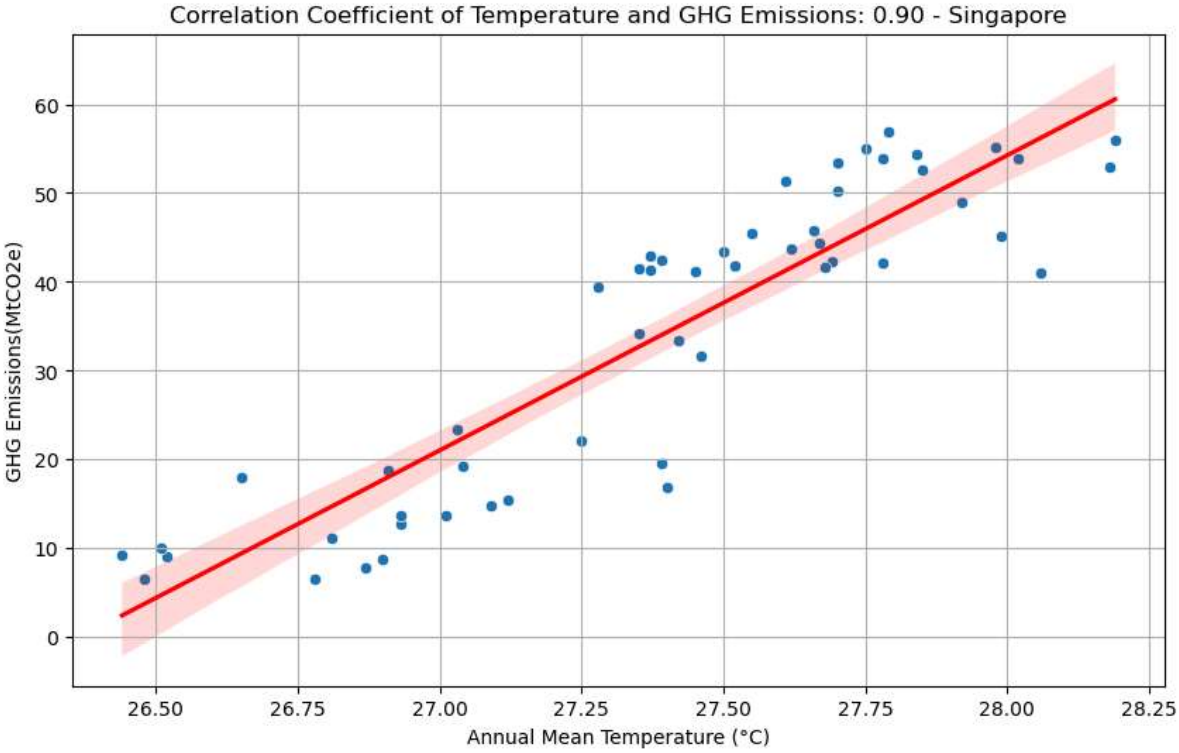
```
# Filtering the data for a particular country
country_data = merged_data[merged_data['Country'] == 'Singapore']

# Selecting the relevant columns
country_data = country_data[['Year', 'Mean_Temperature', 'GHG_Emissions']]

# Calculating the correlation coefficient
correlation_coefficient = country_data['Mean_Temperature'].corr(country_data['GHG_E
```

In [241…

```
# Creatinng a scatter plot with a regression line
plt.figure(figsize=(10, 6))
sns.scatterplot(data=country_data, x='Mean_Temperature', y='GHG_Emissions')
sns.regplot(data=country_data, x='Mean_Temperature', y='GHG_Emissions', scatter=Fal
plt.title(f'Correlation Coefficient of Temperature and GHG Emissions: {correlation_
plt.xlabel('Annual Mean Temperature (°C)')
plt.ylabel('GHG Emissions(MtCO2e)')
plt.grid(True)
plt.show()
```



In [242…

```
# Print the correlation coefficient
print(f'Correlation Coefficient: {correlation_coefficient:.2f}')
```

```
Correlation Coefficient: 0.90
```

Calculation of correlation coefficient of every country

In [243…

```
# Grouping the data by country and calculate the correlation coefficient for each c
correlation_by_country = merged_data.groupby('Country').apply(lambda x: x['Mean_Tem

# Removing countries with NaN correlation coefficients
correlation_by_country = correlation_by_country.dropna()

# Sorting the correlation coefficients in descending order
sorted_correlations = correlation_by_country.sort_values(ascending=False)

# Looping through all countries and sorting their correlation coefficients in desce
for country, correlation in sorted_correlations.items():
    print(f"Country: {country} - Correlation Coefficient: {correlation}")
```

```
Country: Singapore - Correlation Coefficient: 0.9043910686261053
Country: Brazil - Correlation Coefficient: 0.876129585196912
Country: Tunisia - Correlation Coefficient: 0.8682285712305596
Country: Mauritius - Correlation Coefficient: 0.8621834664355271
Country: Bahrain - Correlation Coefficient: 0.8586840221843093
Country: Egypt - Correlation Coefficient: 0.8569307086389132
Country: Libya - Correlation Coefficient: 0.8467679074389589
Country: Saudi Arabia - Correlation Coefficient: 0.8467406706262249
Country: Madagascar - Correlation Coefficient: 0.8439713905021096
Country: Lebanon - Correlation Coefficient: 0.8372925272457432
Country: United Arab Emirates - Correlation Coefficient: 0.8250151480532483
Country: Mexico - Correlation Coefficient: 0.8235785073059004
Country: Jordan - Correlation Coefficient: 0.821061605964699
Country: Guatemala - Correlation Coefficient: 0.8191816779423076
Country: Indonesia - Correlation Coefficient: 0.8086596033113622
Country: Kuwait - Correlation Coefficient: 0.8044523978587473
Country: Qatar - Correlation Coefficient: 0.7986132114433657
Country: Sri Lanka - Correlation Coefficient: 0.7888410922323448
Country: Guinea - Correlation Coefficient: 0.7823896649015479
Country: Malaysia - Correlation Coefficient: 0.7821702680916723
Country: Oman - Correlation Coefficient: 0.775239393862731
Country: Philippines - Correlation Coefficient: 0.7730827239770125
Country: Algeria - Correlation Coefficient: 0.7642106726744271
Country: Belize - Correlation Coefficient: 0.7610039419044732
Country: Cameroon - Correlation Coefficient: 0.7561027973542567
Country: Iraq - Correlation Coefficient: 0.7552287835251872
Country: South Africa - Correlation Coefficient: 0.7550915591965659
Country: Pakistan - Correlation Coefficient: 0.7488364573363221
Country: Kenya - Correlation Coefficient: 0.7488291703528133
Country: Honduras - Correlation Coefficient: 0.7475835606853958
Country: Barbados - Correlation Coefficient: 0.7389571070721295
Country: Comoros - Correlation Coefficient: 0.7381977307745975
Country: Colombia - Correlation Coefficient: 0.7287906635643028
Country: Samoa - Correlation Coefficient: 0.7278060527985254
Country: Seychelles - Correlation Coefficient: 0.7222714415687274
Country: Togo - Correlation Coefficient: 0.7157030456000859
Country: Dominica - Correlation Coefficient: 0.7148095133299404
Country: Tonga - Correlation Coefficient: 0.7044410711032039
Country: Grenada - Correlation Coefficient: 0.7043894605296734
Country: Central African Republic - Correlation Coefficient: 0.7004281727177126
Country: Benin - Correlation Coefficient: 0.6988226451589875
Country: Senegal - Correlation Coefficient: 0.6956497948823569
Country: Ghana - Correlation Coefficient: 0.6912086425038428
Country: Fiji - Correlation Coefficient: 0.6868679283825033
Country: Cyprus - Correlation Coefficient: 0.6853107580248099
Country: Malawi - Correlation Coefficient: 0.6737082497397658
Country: Botswana - Correlation Coefficient: 0.6731435292390175
Country: Sierra Leone - Correlation Coefficient: 0.6686712442997078
Country: Lesotho - Correlation Coefficient: 0.6605213079268107
Country: Nicaragua - Correlation Coefficient: 0.6550138675183799
Country: Dominican Republic - Correlation Coefficient: 0.6488865968440035
Country: Iceland - Correlation Coefficient: 0.6419806756754394
Country: Portugal - Correlation Coefficient: 0.6411898495800381
Country: Costa Rica - Correlation Coefficient: 0.6352292940090978
Country: Ethiopia - Correlation Coefficient: 0.6279406020375106
Country: Rwanda - Correlation Coefficient: 0.6228251081787332
Country: Haiti - Correlation Coefficient: 0.61595698940223
Country: Turkmenistan - Correlation Coefficient: 0.6128187056416184
Country: Malta - Correlation Coefficient: 0.6107428751447072
Country: Tanzania - Correlation Coefficient: 0.6023324395337378
Country: Japan - Correlation Coefficient: 0.602101701411101
Country: Thailand - Correlation Coefficient: 0.5955758504268092
Country: Vietnam - Correlation Coefficient: 0.5771823278803718
Country: Panama - Correlation Coefficient: 0.5762275181026916
```

```
Country: Nigeria - Correlation Coefficient: 0.5570077429659384
Country: Chile - Correlation Coefficient: 0.5560297431040755
Country: Suriname - Correlation Coefficient: 0.5477640771519342
Country: Bhutan - Correlation Coefficient: 0.5410552064647929
Country: Canada - Correlation Coefficient: 0.5336610828225127
Country: Australia - Correlation Coefficient: 0.5315798818347506
Country: Mauritania - Correlation Coefficient: 0.5268578576949827
Country: Maldives - Correlation Coefficient: 0.5171139344886074
Country: Argentina - Correlation Coefficient: 0.5076776748609764
Country: Eritrea - Correlation Coefficient: 0.5056944891142091
Country: Cambodia - Correlation Coefficient: 0.5050969649239403
Country: Uganda - Correlation Coefficient: 0.4982928142014223
Country: Eswatini - Correlation Coefficient: 0.4975796864514251
Country: Mongolia - Correlation Coefficient: 0.4974799138885919
Country: Nepal - Correlation Coefficient: 0.4954566322649253
Country: Burundi - Correlation Coefficient: 0.49109455332986846
Country: Niger - Correlation Coefficient: 0.48332904059978177
Country: Paraguay - Correlation Coefficient: 0.4790550922930625
Country: Mali - Correlation Coefficient: 0.478746011092791
Country: Guyana - Correlation Coefficient: 0.474510935723356
Country: Mozambique - Correlation Coefficient: 0.45037232086471335
Country: Afghanistan - Correlation Coefficient: 0.43921574329657787
Country: Norway - Correlation Coefficient: 0.43720594821401004
Country: Uzbekistan - Correlation Coefficient: 0.4370278248786446
Country: Slovenia - Correlation Coefficient: 0.42498083359431466
Country: Chad - Correlation Coefficient: 0.40114232756681656
Country: Peru - Correlation Coefficient: 0.4001757753860269
Country: Ecuador - Correlation Coefficient: 0.38466559049020993
Country: Austria - Correlation Coefficient: 0.3801055645119264
Country: Angola - Correlation Coefficient: 0.33381981998299154
Country: Greece - Correlation Coefficient: 0.3076734197681575
Country: United States - Correlation Coefficient: 0.30744120626964716
Country: Kiribati - Correlation Coefficient: 0.28788977442968755
Country: Vanuatu - Correlation Coefficient: 0.27887388917578254
Country: Uruguay - Correlation Coefficient: 0.26606876262592827
Country: New Zealand - Correlation Coefficient: 0.24949105715449674
Country: Kazakhstan - Correlation Coefficient: 0.24549229519230112
Country: Gabon - Correlation Coefficient: 0.240438064942847
Country: Jamaica - Correlation Coefficient: 0.2379315892479943
Country: Bangladesh - Correlation Coefficient: 0.22515585776810743
Country: Somalia - Correlation Coefficient: 0.18718560496733913
Country: Netherlands - Correlation Coefficient: 0.1847100234403718
Country: Zimbabwe - Correlation Coefficient: 0.18338424813580018
Country: Bolivia - Correlation Coefficient: 0.11837492226022146
Country: Liberia - Correlation Coefficient: 0.016940702594655146
Country: Solomon Islands - Correlation Coefficient: -0.014661390330446302
Country: Papua New Guinea - Correlation Coefficient: -0.027224572585812647
Country: Djibouti - Correlation Coefficient: -0.044407313226224134
Country: Finland - Correlation Coefficient: -0.09996484233462755
Country: Zambia - Correlation Coefficient: -0.14029092504347265
Country: Tajikistan - Correlation Coefficient: -0.20455954761765843
Country: Cuba - Correlation Coefficient: -0.21434667863415355
Country: Estonia - Correlation Coefficient: -0.23282033012990677
Country: Albania - Correlation Coefficient: -0.32023596167035384
Country: Belarus - Correlation Coefficient: -0.34441297793627995
Country: Palau - Correlation Coefficient: -0.3567712508689859
Country: Armenia - Correlation Coefficient: -0.4431452823431364
Country: Azerbaijan - Correlation Coefficient: -0.4748451588040713
Country: Georgia - Correlation Coefficient: -0.47586512654092933
Country: Croatia - Correlation Coefficient: -0.4912883176595166
Country: Latvia - Correlation Coefficient: -0.5084746337297732
Country: Luxembourg - Correlation Coefficient: -0.5285881820225004
Country: Sweden - Correlation Coefficient: -0.5526322113347191
Country: Moldova - Correlation Coefficient: -0.5838615370327306
```

```
Country: Poland - Correlation Coefficient: -0.5886531490502439
Country: Belgium - Correlation Coefficient: -0.5899306420536067
Country: United Kingdom - Correlation Coefficient: -0.6086067095826214
Country: Ukraine - Correlation Coefficient: -0.656329696254151
Country: Denmark - Correlation Coefficient: -0.6651351029954766
Country: Romania - Correlation Coefficient: -0.720284325951142
Country: Hungary - Correlation Coefficient: -0.7347545669148446
Country: Bulgaria - Correlation Coefficient: -0.7390735448893608
Country: Germany - Correlation Coefficient: -0.7559414977825439
```

Countries with highest temperature rise in last 5 years (2018 - 2022)

In [244…

```
last_5_years = [2018, 2019, 2020, 2021, 2022]
filtered_data = merged_data[merged_data['Year'].isin(last_5_years)]
```

In [245…

```
filtered_data['Temp_Diff'] = filtered_data.groupby('Country')['Mean_Temperature'].c
```

```
C:\Users\soumy\AppData\Local\Temp\ipykernel_22084\2268163652.py:1: SettingWithCopy
Warning:
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/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_data['Temp_Diff'] = filtered_data.groupby('Country')['Mean_Temperatur
e'].diff()
```

In [246…

```
mean_temp_rise = filtered_data.groupby('Country')['Temp_Diff'].mean().reset_index()
mean_temp_rise.columns = ['Country', 'Mean_Temperature_Rise']
```

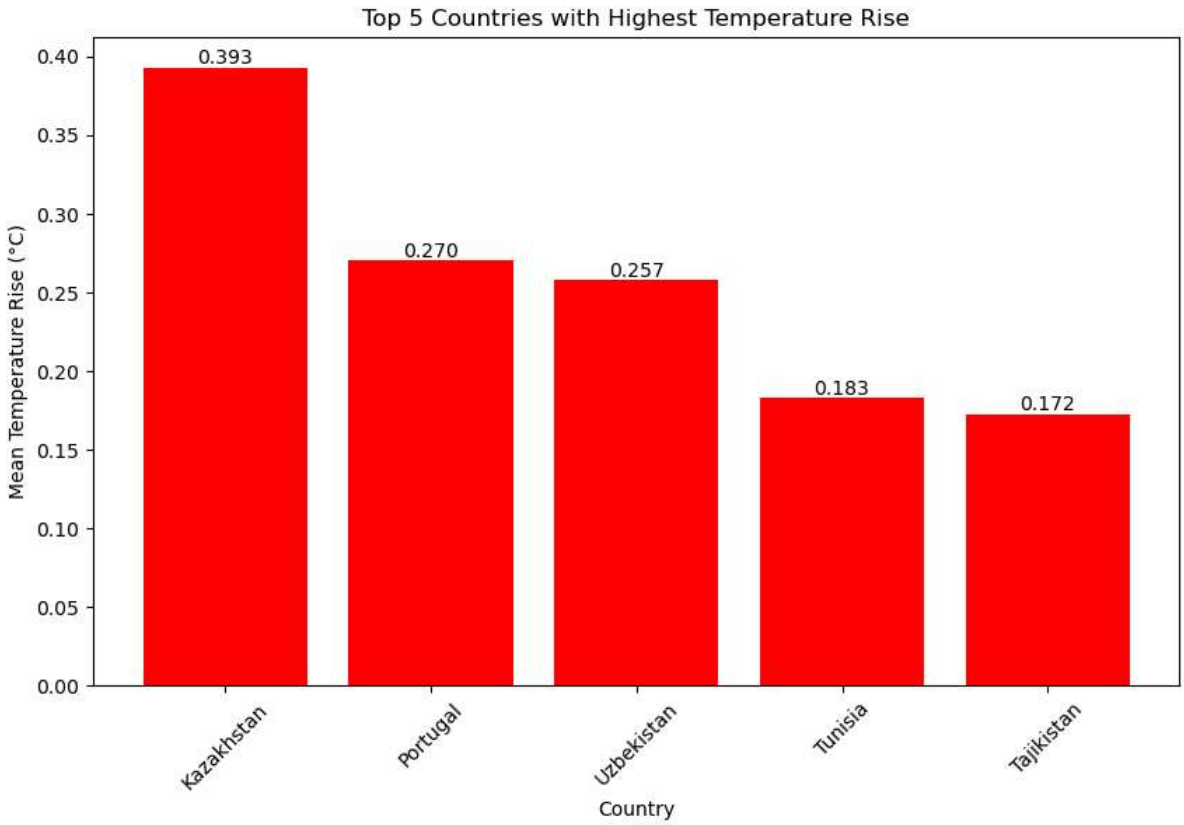
In [247…

```
# Sorting DataFrame by Mean_Temperature_Rise in descending order and selecting top
top_5_countries = mean_temp_rise.sort_values(by='Mean_Temperature_Rise', ascending=

# Plot
plt.figure(figsize=(10, 6))
bars = plt.bar(top_5_countries['Country'], top_5_countries['Mean_Temperature_Rise']
plt.xlabel('Country')
plt.ylabel('Mean Temperature Rise (°C)')
plt.title('Top 5 Countries with Highest Temperature Rise')
plt.xticks(rotation=45)

# Add data labels
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.3f}', ha='center

plt.show()
```



Conclusion:

- Kazakhstan is the country with the highest temperature rise over the last 5 years followed by Portugal, Uzbekistan, Tunisia and Tajikistan.

Countries with least temperature rise in last 5 years (2018 - 2022)

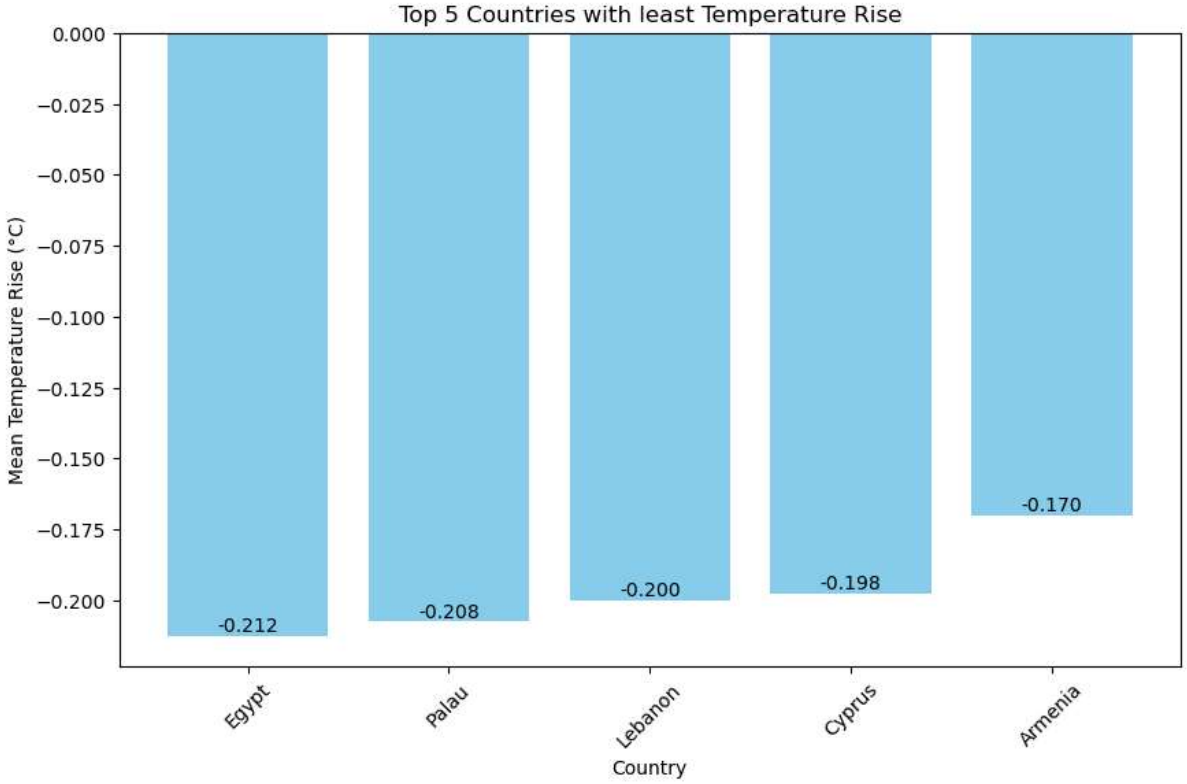
In [248…

```
# Sorting DataFrame by Mean_Temperature_Rise in descending order and selecting top
top_5_countries = mean_temp_rise.sort_values(by='Mean_Temperature_Rise', ascending=

# Plot
plt.figure(figsize=(10, 6))
bars = plt.bar(top_5_countries['Country'], top_5_countries['Mean_Temperature_Rise']
plt.xlabel('Country')
plt.ylabel('Mean Temperature Rise (°C)')
plt.title('Top 5 Countries with least Temperature Rise')
plt.xticks(rotation=45)

# Add data labels
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.3f}', ha='center

plt.show()
```



Conclusion:

- Egypt is the country with the least temperature rise in the last 5 years even it has got colder over the years.

Countries with highest GHG emissions rise over the last 5 years

In [249…

```
filtered_data['Emissions_Diff'] = filtered_data.groupby('Country')['GHG_Emissions']
mean_emissions_rise = filtered_data.groupby('Country')['Emissions_Diff'].mean().res
mean_emissions_rise.columns = ['Country', 'Mean_Emissions_Rise']
```

```
C:\Users\soumy\AppData\Local\Temp\ipykernel_22084\254088629.py:1: SettingWithCopyW
arning:
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/stabl
e/user_guide/indexing.html#returning-a-view-versus-a-copy
  filtered_data['Emissions_Diff'] = filtered_data.groupby('Country')['GHG_Emission
s'].diff()
```

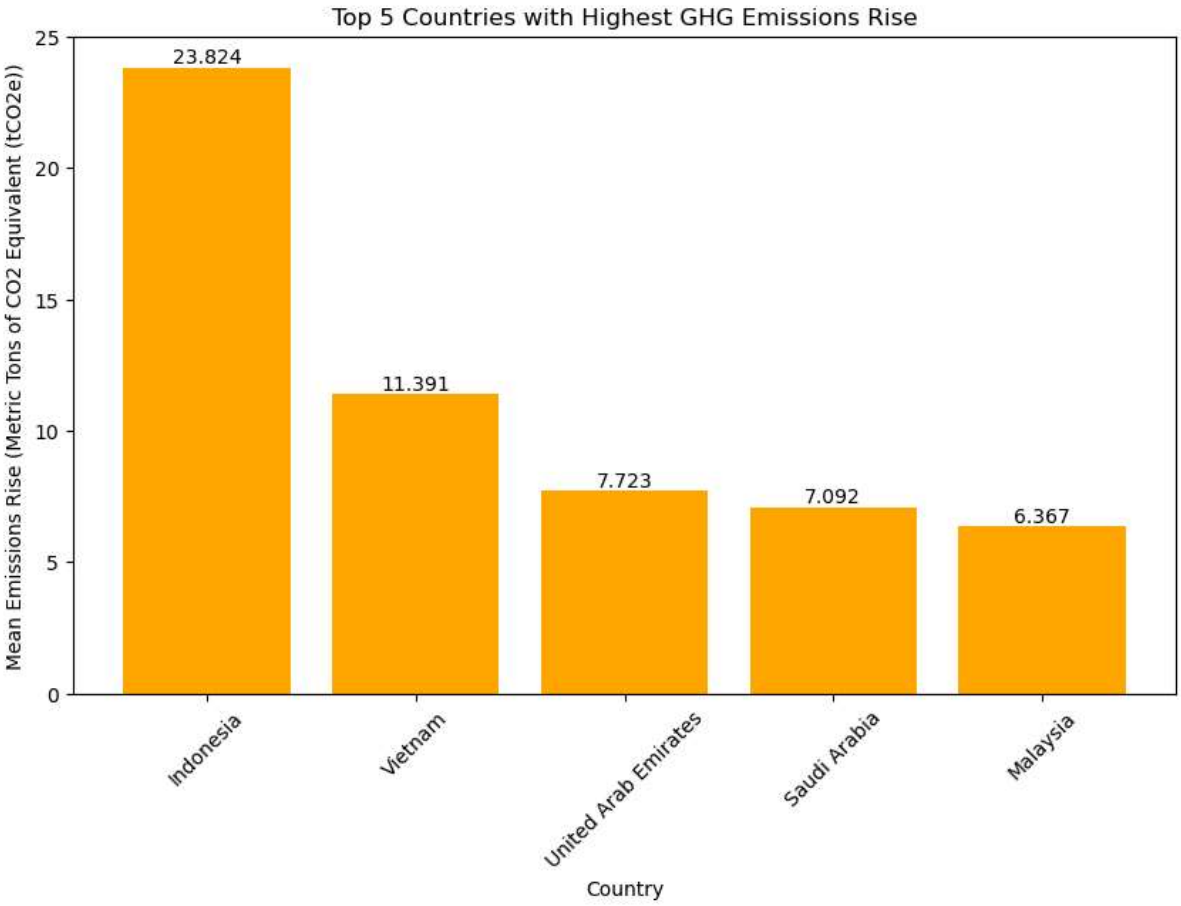
In [250…

```
# Sorting DataFrame by Mean_Emissions_Rise in descending order and selecting top 5
top_5_countries = mean_emissions_rise.sort_values(by='Mean_Emissions_Rise', ascendi

# Plot
plt.figure(figsize=(10, 6))
bars = plt.bar(top_5_countries['Country'], top_5_countries['Mean_Emissions_Rise'],
plt.xlabel('Country')
plt.ylabel('Mean Emissions Rise (Metric Tons of CO2 Equivalent (tCO2e))')
plt.title('Top 5 Countries with Highest GHG Emissions Rise')
plt.xticks(rotation=45)
```

```
# Add data labels
for bar in bars:
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.3f}', ha='center

plt.show()
```



Conclusion:

- Indonesia is the country with the highest GHG Emissions rise over the last 5 years followed by Vietnam, UAE, Saudi Arabia and Malaysia.

Countries with least GHG emissions rise over the last 5 years

In [251…

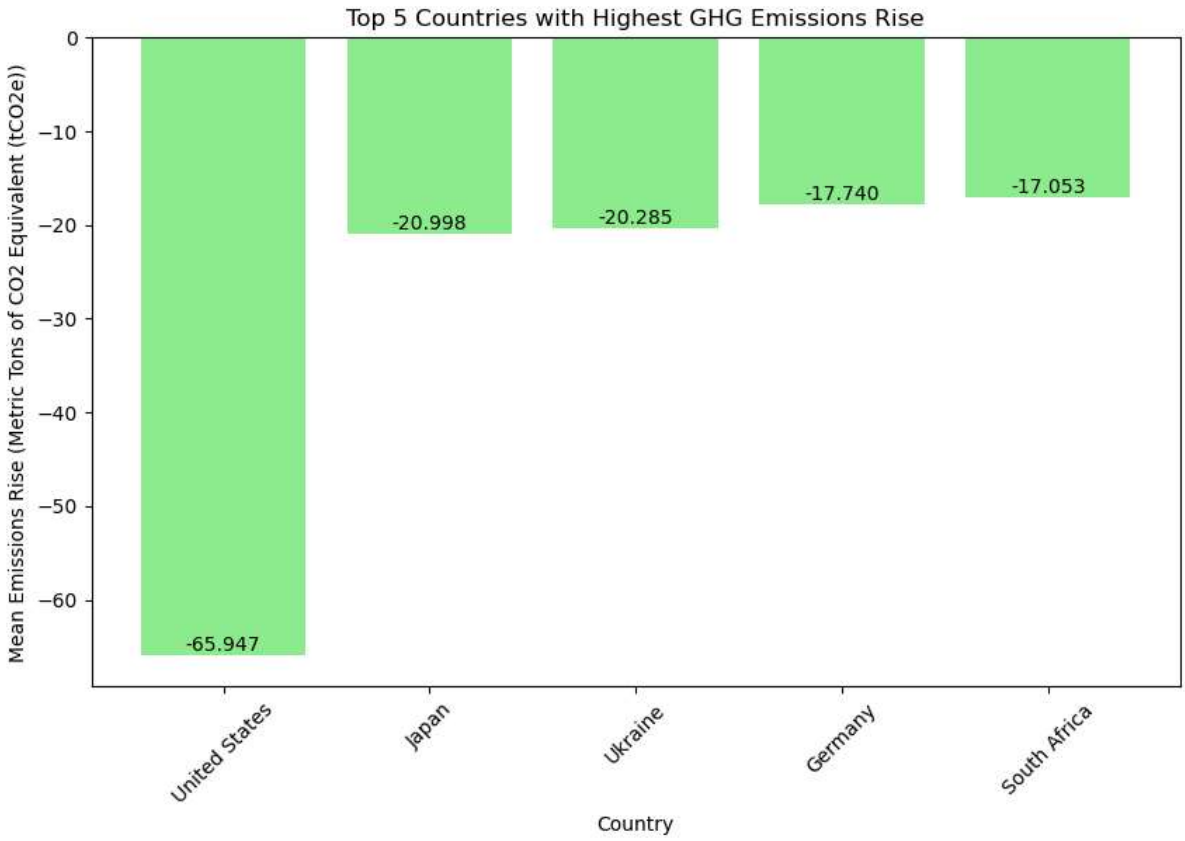
```
# Sorting DataFrame by Mean_Emissions_Rise in descending order and selecting top 5
top_5_countries = mean_emissions_rise.sort_values(by='Mean_Emissions_Rise', ascendi

# Plot
plt.figure(figsize=(10, 6))
bars = plt.bar(top_5_countries['Country'], top_5_countries['Mean_Emissions_Rise'],
plt.xlabel('Country')
plt.ylabel('Mean Emissions Rise (Metric Tons of CO2 Equivalent (tCO2e))')
plt.title('Top 5 Countries with Highest GHG Emissions Rise')
plt.xticks(rotation=45)

# Add data labels
for bar in bars:
```

```
    height = bar.get_height()
    plt.text(bar.get_x() + bar.get_width() / 2, height, f'{height:.3f}', ha='center

plt.show()
```



Conclusion:

- United States managed to decrease the average GHG emission rate over the last 5 years by 65.95 metric tons of CO2 equivalent.
- Countries like Japan, Ukraine, Germany and South Africa also managed to decrease their GHG emissions significantly.
- This demonstrates the successful adoption of green technologies and the progress toward sustainable development achieved by these countries, as evidenced by their reduced rates of GHG emissions over the years.