Annual Greenhouse Gas (GHG) Air Emissions Accounts

Group : Group 6

Group Members :

Name	Matrix Number
NG SUANG JOO	A21EC0102
LING WAN YIN	A21EC0047
FONG KHAH KHEH	A21EC0026
KEE SHIN PEARL	A21EC0190

About Dataset

Dataset Description:

The dataset, 'Annual Greenhouse Gas (GHG) Air Emissions Accounts,' downloaded from the International Monetary Fund's Climate Change Dashboard website, consists of 1186 records and 27 columns. To give an overview, this dataset is regarding the annual greenhouse gas emissions by activity and by region, it provides insights into annual GHG emissions by activity and region, with columns ranging from identifiers (ObjectId2) to detailed descriptors (CTS_Full_Descriptor) for the Climate Tracking System (CTS). The goal of this project is to analyze the annual greenhouse gas (GHG) emissions of various nations, investigate any relationships with economic status, look at historical trends for particular country groups, find anomalies, and possibly use machine learning techniques to predict future emissions.

Dataset Attributes:

- This dataset has a total of 1186 records with 27 columns of attributes.

Column Name	Description
ObjectId2	Identifier for the record
Country	Name of the country
ISO2	Two-letter country code
ISO3	Three-letter country code
Indicator	Type of greenhouse gas emission indicator
Unit	Measurement unit for the emission data
Source	Source of the emission data
CTS_Code	Code for the Climate Tracking System (CTS)
CTS_Name	Name of the Climate Tracking System

Column Name	Description
CTS_Full_Descriptor	Detailed descriptor for the Climate Tracking System
Industry	Industry associated with the emission data
Gas_Type	Type of greenhouse gas
Seasonal_Adjustment	Information on seasonal adjustment
Scale	Measurement scale for the emissions data
F2010 to F2022	Annual greenhouse gas emissions data for each year

Downloading the Dataset

Please use the link below to access and download the dataset: Annual Greenhouse Gas (GHG) Air Emissions Accounts

Research Questions

1. Can machine learning models accurately predict a country's greenhouse gas emissions for the year 2022 based on historical data from 2013 to 2021?

1. How well can machine learning models differentiate between normal emission patterns and anomalies?

1. How well does the model predict annual greenhouse gas (GHG) emissions for the year 2022 based on historical data from the year 2013?

1. How well can machine learning models differentiate between various countries (Country column) based on their greenhouse gas emissions, considering different gas types and industries (Gas_Type and Industry columns)?

2. Which features have the most significant impact on predicting greenhouse gas emissions in a country?

3. Can we classify countries into different groups or clusters and observe patterns in their greenhouse gas emissions over the years, using clustering algorithms?

4. Can machine learning classify emissions patterns for different gas types?

Load the Dataset

In [2]:
```
import pandas as pd

file_path = "Annual_Greenhouse_Gas_(GHG)_Air_Emissions_Accounts.csv"

data = pd.read_csv(file_path)

data.head()
```

Out[2]:

	ObjectId2	Country	ISO2	ISO3	Indicator	Unit	Source	CTS_Code	CTS_Nam
0	1	Advanced Economies	NaN	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhous Ga Emission (GHG); Ai Emissions .
1	2	Advanced Economies	NaN	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhous Ga Emission (GHG); Ai Emissions .
2	3	Advanced Economies	NaN	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhous Ga Emission (GHG); Ai Emissions .
3	4	Advanced Economies	NaN	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhous Ga Emission (GHG); Ai Emissions .
4	5	Advanced Economies	NaN	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhous Ga Emission (GHG); Ai Emissions .

5 rows × 27 columns

In [3]:
```
data.shape
```

Out[3]:
```
(1186, 27)
```

In [4]:
```
data.dtypes
```

Out[4]:
```
ObjectId2               int64
Country                object
ISO2                  float64
ISO3                   object
Indicator              object
Unit                   object
Source                 object
CTS_Code               object
CTS_Name               object
CTS_Full_Descriptor    object
Industry               object
Gas_Type               object
Seasonal_Adjustment    object
Scale                  object
F2010                 float64
F2011                 float64
F2012                 float64
F2013                 float64
F2014                 float64
F2015                 float64
F2016                 float64
F2017                 float64
F2018                 float64
F2019                 float64
F2020                 float64
F2021                 float64
F2022                 float64
dtype: object
```

Dataset Attributes

In [5]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1186 entries, 0 to 1185
Data columns (total 27 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   ObjectId2            1186 non-null   int64
 1   Country              1186 non-null   object
 2   ISO2                 0 non-null      float64
 3   ISO3                 1186 non-null   object
 4   Indicator            1186 non-null   object
 5   Unit                 1186 non-null   object
 6   Source               1186 non-null   object
 7   CTS_Code             1186 non-null   object
 8   CTS_Name             1186 non-null   object
 9   CTS_Full_Descriptor  1186 non-null   object
 10  Industry             1186 non-null   object
 11  Gas_Type             1186 non-null   object
 12  Seasonal_Adjustment  1186 non-null   object
 13  Scale                1186 non-null   object
 14  F2010                1186 non-null   float64
 15  F2011                1186 non-null   float64
 16  F2012                1186 non-null   float64
 17  F2013                1186 non-null   float64
 18  F2014                1186 non-null   float64
 19  F2015                1186 non-null   float64
 20  F2016                1186 non-null   float64
 21  F2017                1186 non-null   float64
 22  F2018                1186 non-null   float64
 23  F2019                1186 non-null   float64
 24  F2020                1186 non-null   float64
 25  F2021                1186 non-null   float64
 26  F2022                1186 non-null   float64
dtypes: float64(14), int64(1), object(12)
memory usage: 250.3+ KB
```

Dataset Overview

In [6]:
```
data.describe()
```

Out[6]:

	ObjectId2	ISO2	F2010	F2011	F2012	F2013	F2014
count	1186.000000	0.0	1.186000e+03	1.186000e+03	1.186000e+03	1.186000e+03	1.186000e+03
mean	696.345700	NaN	7.871620e+02	8.087643e+02	8.192881e+02	8.328579e+02	8.399073e+02
std	368.171291	NaN	2.986214e+03	3.094635e+03	3.146775e+03	3.207087e+03	3.240924e+03
min	1.000000	NaN	-4.440000e-15	-9.380000e-15	-5.880000e-15	-9.100000e-15	-1.550000e-15
25%	416.250000	NaN	3.392096e+00	3.359107e+00	3.506533e+00	3.609378e+00	3.636859e+00
50%	712.500000	NaN	4.115801e+01	4.153315e+01	4.287393e+01	4.364143e+01	4.393280e+01
75%	1008.750000	NaN	3.577597e+02	3.600748e+02	3.660298e+02	3.686307e+02	3.785122e+02
max	1305.000000	NaN	4.624233e+04	4.760897e+04	4.832278e+04	4.912791e+04	4.960594e+04

Data Cleaning and Preparation

Missing value

In [7]:
```
missing_values_per_column = data.isna().sum()

print("Missing values in each column:")
print(missing_values_per_column)
```

```
Missing values in each column:
ObjectId2                 0
Country                   0
ISO2                   1186
ISO3                      0
Indicator                 0
Unit                      0
Source                    0
CTS_Code                  0
CTS_Name                  0
CTS_Full_Descriptor       0
Industry                  0
Gas_Type                  0
Seasonal_Adjustment       0
Scale                     0
F2010                     0
F2011                     0
F2012                     0
F2013                     0
F2014                     0
F2015                     0
F2016                     0
F2017                     0
F2018                     0
F2019                     0
F2020                     0
F2021                     0
F2022                     0
dtype: int64
```

It shows that there are missing values in the ISO2 column.

Drop the empty column 'ISO2'

Since the column 'ISO2' consists of null values, remove it as it does not display any important insights into the data.

In [8]:
```
data.drop('ISO2', axis=1, inplace=True)
data
```

Out[8]:

	ObjectId2	Country	ISO3	Indicator	Unit	Source	CTS_Code	CTS_Name
0	1	Advanced Economies	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
1	2	Advanced Economies	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
2	3	Advanced Economies	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
3	4	Advanced Economies	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
4	5	Advanced Economies	AETMP	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
...	...	...	...	...	...	...	...	...
1181	1301	World	WLD	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
1182	1302	World	WLD	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
1183	1303	World	WLD	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...
1184	1304	World	WLD	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...

	ObjectId2	Country	ISO3	Indicator	Unit	Source	CTS_Code	CTS_Name
1185	1305	World	WLD	Annual greenhouse gas (GHG) air emissions acco...	Million metric tons of CO2 equivalent	Organisation for Economic Co-operation and Dev...	ECNGA	Greenhouse Gas Emissions (GHG); Air Emissions ...

Duplicated rows

Now, let's check whether there are any duplicate rows.

In [9]:
```
duplicated_rows = data.duplicated()

# Check if there are any duplicated rows
if duplicated_rows.any():
    print("There are duplicated rows in the DataFrame.")
    duplicated_df = data[duplicated_rows]
    print("Duplicated Rows:", duplicated_df)
    print(duplicated_df)

else:
    print("There are no duplicated rows found.")
```

```
There are no duplicated rows found.
```

Select only the necessary columns

Selecting specific columns related to GHG emissions, such as the country, ISO code, industry, gas type, and emission values for different years, as they are the main data that we used in the analysis.

In [10]:
```
selected_columns = ['ObjectId2', 'Country', 'ISO3','Industry', 'Gas_Type',
      'F2010','F2011', 'F2012', 'F2013', 'F2014', 'F2015',
      'F2016', 'F2017', 'F2018', 'F2019', 'F2020', 'F2021', 'F2022']
df = data[selected_columns]
df.head()
```

Out[10]:

	ObjectId2	Country	ISO3	Industry	Gas_Type	F2010	F2011	F2012
0	1	Advanced Economies	AETMP	Agriculture, Forestry and Fishing	Carbon dioxide	194.398492	191.201537	192.473034
1	2	Advanced Economies	AETMP	Agriculture, Forestry and Fishing	Fluorinated gases	0.900844	0.948342	0.955449
2	3	Advanced Economies	AETMP	Agriculture, Forestry and Fishing	Greenhouse gas	1370.031102	1350.212366	1334.456632
3	4	Advanced Economies	AETMP	Agriculture, Forestry and Fishing	Methane	641.415104	636.893906	637.064644
4	5	Advanced Economies	AETMP	Agriculture, Forestry and Fishing	Nitrous oxide	533.316661	521.168581	503.963505

Data Aggregation and Group Operations

In [11]:
```
grouped_df = df.groupby(['Country', 'Industry', 'Gas_Type']).sum().reset_index()
grouped_df
```

Out[11]:

	Country	Industry	Gas_Type	ObjectId2	ISO3	F2010	F2011	F2(
0	Advanced Economies	Agriculture, Forestry and Fishing	Carbon dioxide	1	AETMP	194.398492	191.201537	192.473
1	Advanced Economies	Agriculture, Forestry and Fishing	Fluorinated gases	2	AETMP	0.900844	0.948342	0.955
2	Advanced Economies	Agriculture, Forestry and Fishing	Greenhouse gas	3	AETMP	1370.031102	1350.212366	1334.456
3	Advanced Economies	Agriculture, Forestry and Fishing	Methane	4	AETMP	641.415104	636.893906	637.064
4	Advanced Economies	Agriculture, Forestry and Fishing	Nitrous oxide	5	AETMP	533.316661	521.168581	503.963
...	...	...	...	...	...	...	...	
1181	World	Water supply; sewerage, waste management and r...	Carbon dioxide	1301	WLD	156.348730	157.792392	162.289
1182	World	Water supply; sewerage, waste management and r...	Fluorinated gases	1302	WLD	5.958067	6.762982	7.890
1183	World	Water supply; sewerage, waste management and r...	Greenhouse gas	1303	WLD	2202.583180	2240.447082	2284.232
1184	World	Water supply; sewerage, waste management and r...	Methane	1304	WLD	1929.636985	1963.335951	1998.979
1185	World	Water supply; sewerage, waste management and r...	Nitrous oxide	1305	WLD	110.639398	112.555757	115.072

1186 rows × 18 columns

In [12]:
```
grouped_df2 = df.groupby(['Country', 'ISO3']).sum()
grouped_df2
```

Out[12]:

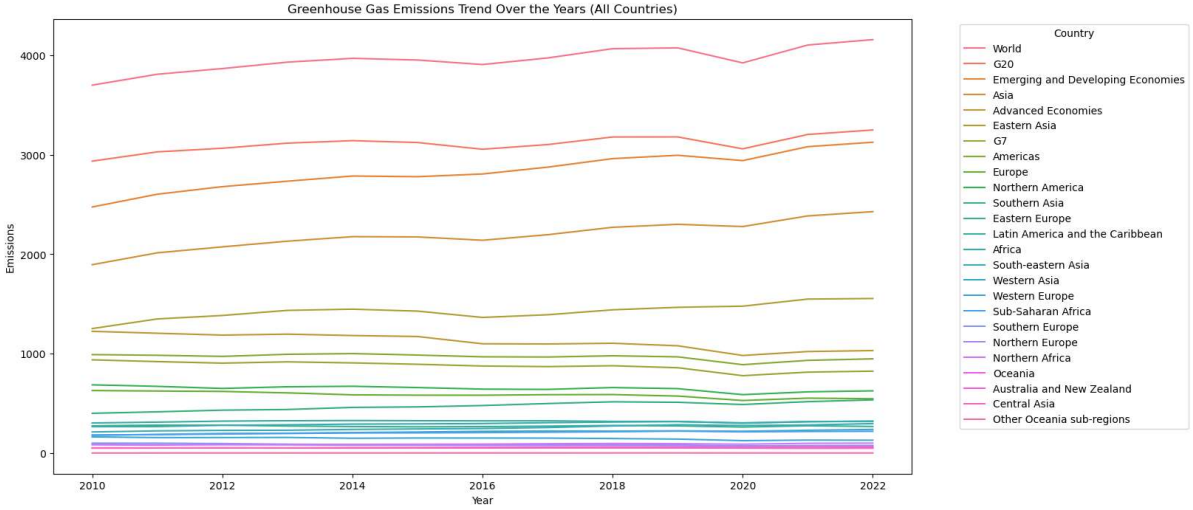
Country	ISO3	ObjectId2	Industry	Gas_Type	F2010	F2
Advanced Economies	**AETMP**	1275	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	61235.081402	60285.728
Africa	**NA605**	3096	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	11486.815188	11511.222
Americas	**AMETMP**	5220	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	44586.762970	44296.052
Asia	**ASIATMP**	11388	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	94736.605724	100699.015
Australia and New Zealand	**NAANZ**	16625	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	2590.372836	2598.284
Central Asia	**NACA**	19125	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	2554.277896	2594.989
Eastern Asia	**NA510**	19350	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	56370.403111	60722.367
Eastern Europe	**NAEE**	23875	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	13754.834957	14094.427
Emerging and Developing Economies	**EMDETMP**	26375	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	123734.234729	130150.151
Europe	**EURTMP**	28875	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	31488.265267	31250.397
G20	**NA120**	31375	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	146772.772122	151417.329
G7	**NA119**	33875	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	46924.123965	46033.554
Latin America and the Caribbean	**LACTMP**	32625	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	13671.259556	14054.730

Country	ISO3	ObjectId2	Industry	Gas_Type	F2010	F2
Northern Africa	NANA9	33067	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	3604.573439	3493.068
Northern America	NA225	36585	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	30915.503416	30241.322
Northern Europe	NANE	43025	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	4482.139377	4267.327
Oceania	OCETMP	45525	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	2670.866979	2679.19
Other Oceania sub-regions	NAOOSR	38220	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	80.494143	80.907
South-eastern Asia	NASEA	42871	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	9192.363349	9647.176
Southern Asia	NASA	45782	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	17648.845908	18296.813
Southern Europe	NASE	54375	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	5062.578136	5031.780
Sub-Saharan Africa	SSA	48762	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideGreenhouse gasMethaneNitrous oxi...	7882.241749	8018.154
Western Asia	NAWA	59025	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	8970.715462	9437.668
Western Europe	NAWE	61525	Agriculture, Forestry and FishingAgriculture, ...	Carbon dioxideFluorinated gasesGreenhouse gasM...	8188.712796	7856.861
World	WLD	64025	Agriculture, Forestry and FishingAgriculture,	Carbon dioxideFluorinated gasesGreenhouse	184969.316121	190435.879

Analysis and visualization

Greenhouse Gas Emissions Trend Over the Years (All Countries)

In [13]:
```
import seaborn as sns
import matplotlib.pyplot as plt

selected_columns = ['Country', 'F2010', 'F2011', 'F2012', 'F2013', 'F2014', 'F2015'
df_selected = df[selected_columns]

# Melt the DataFrame
df_selected_melted = df_selected.melt(id_vars=['Country'], var_name='Year', value_r

# Extract the year and convert it to int
df_selected_melted['Year'] = df_selected_melted['Year'].str.extract('(\d+)', expand

# Sort the DataFrame by Year and Emissions
df_selected_melted = df_selected_melted.sort_values(by=['Year', 'Emissions'], ascer

# Visualization
plt.figure(figsize=(15, 8))
sns.lineplot(x='Year', y='Emissions', hue='Country', data=df_selected_melted, err_s
plt.title('Greenhouse Gas Emissions Trend Over the Years (All Countries)')
plt.xlabel('Year')
plt.ylabel('Emissions')
plt.legend(title='Country', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [14]:
```
df_selected_melted
```

Out[14]:

	Country	Year	Emissions
1173	World	2010	4.624233e+04
520	G20	2010	3.669319e+04
1171	World	2010	3.364481e+04
420	Emerging and Developing Economies	2010	3.093356e+04
518	G20	2010	2.837054e+04
...	...	...	...
14481	Central Asia	2022	0.000000e+00
14491	Central Asia	2022	0.000000e+00
14501	Central Asia	2022	0.000000e+00
14516	Central Asia	2022	0.000000e+00
14511	Central Asia	2022	-7.110000e-15

15418 rows × 3 columns

Analysis:

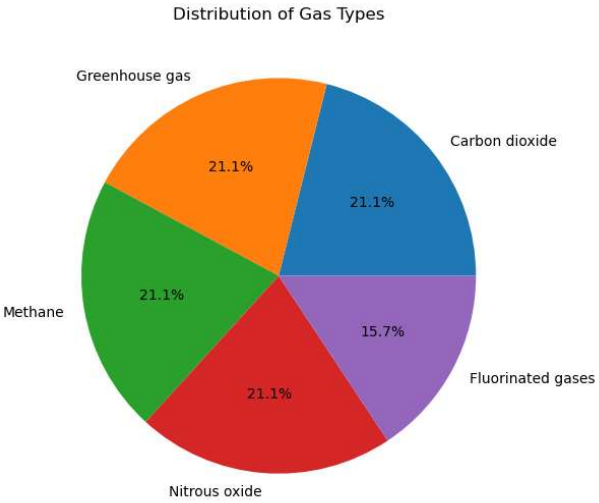
The chart represents the greenhouse gas emissions trend over the years for various country groups. The World and G20 exhibit substantial emissions throughout the years, with the World's emissions peaking at approximately 51960.17 in 2022. Other than World, G20 has the most greenhouse gas emissions. This is because G20 comprises major economies and these nations often have higher levels of industrialization and economic activity. Increased industrial production, manufacturing, and energy consumption contribute significantly to greenhouse gas emissions. The Central Asia shows a trend of consistently low or zero emissions, as indicated by the entries with values of 0 or close to zero from 2010 to 2022. The economic structure of Central Asian countries may be less dependent on heavy industry and carbon-intensive activities. Economies dominated by agriculture or services may contribute to lower emissions compared to industrialised countries.

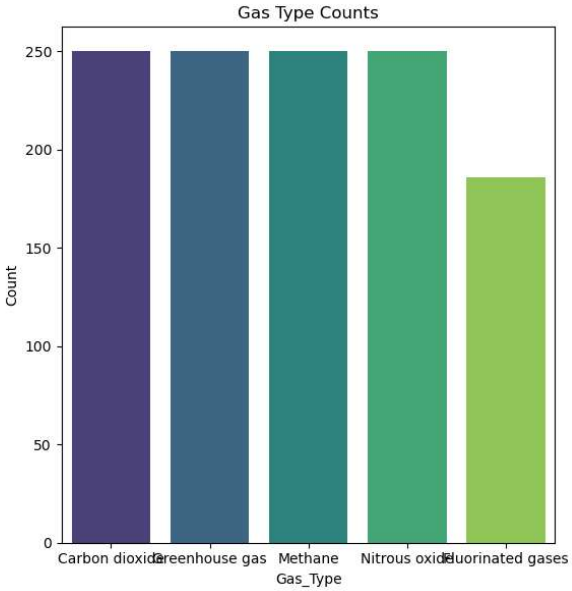
Pie chart for the distribution of Gas Types

In [15]:
```
gas_type_counts = data['Gas_Type'].value_counts()

gas_type_df = pd.DataFrame({'Gas_Type': gas_type_counts.index, 'Count': gas_type_cc

print("Gas Type Counts:")
print(gas_type_df)

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

# Pie chart
plt.subplot(1, 2, 1)
gas_type_counts.plot.pie(autopct='%1.1f%%')
plt.title('Distribution of Gas Types')
plt.ylabel('')

# Bar chart (count)
plt.subplot(1, 2, 2)
sns.barplot(x='Gas_Type', y='Count', data=gas_type_df, palette='viridis')
```

```
plt.title('Gas Type Counts')

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

```
Gas Type Counts:
            Gas_Type  Count
0     Carbon dioxide    250
1     Greenhouse gas    250
2            Methane    250
3      Nitrous oxide    250
4  Fluorinated gases    186
```



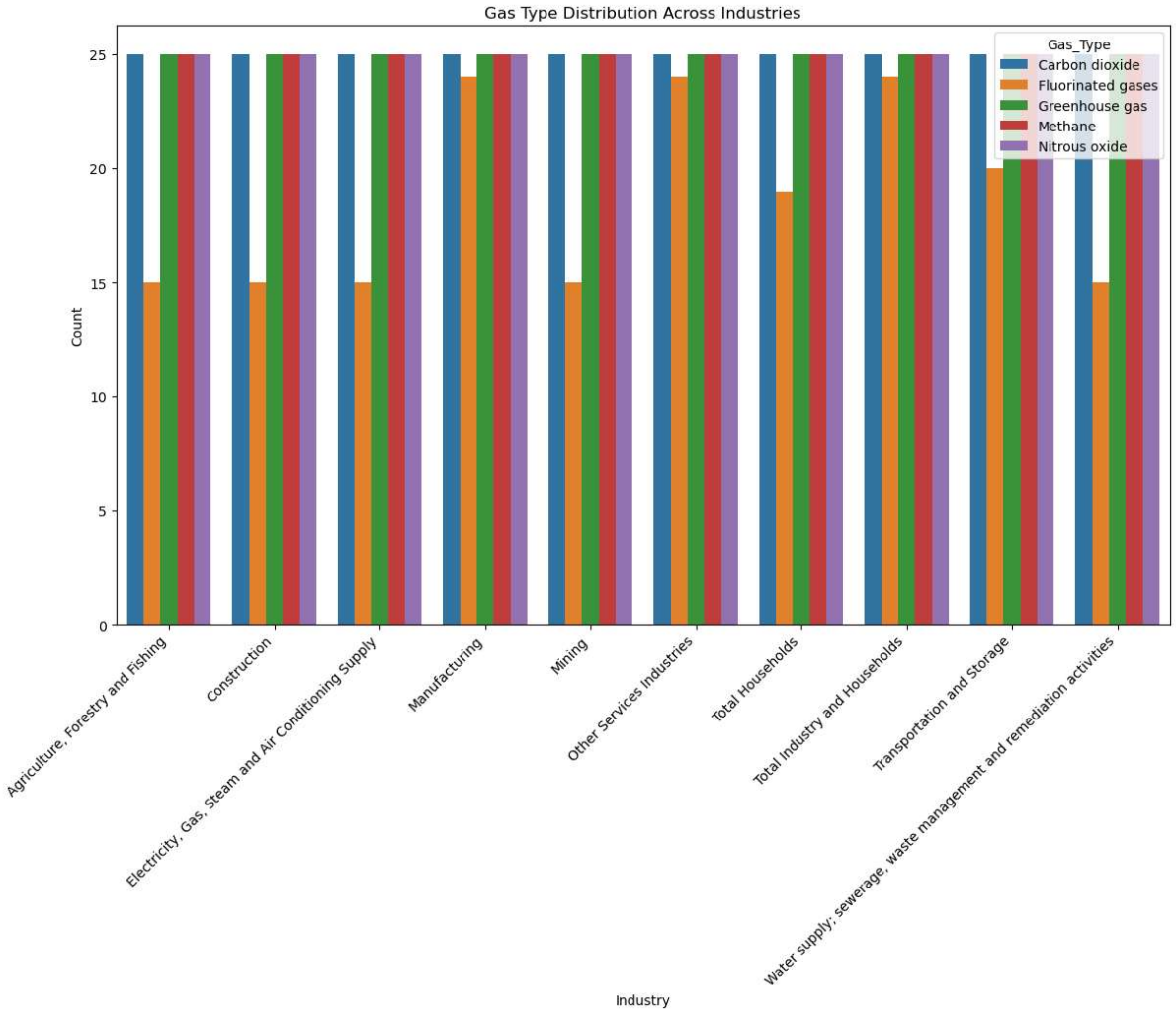


Analysis:

The pie chart depicts the distribution of gas types in the dataset and includes five different categories. Carbon dioxide, greenhouse gases, methane and nitrous oxide each account for 21.1 percent of the total gas types, highlighting the relatively balanced representation of these major components of greenhouse gas emissions. Fluorinated gases, although a smaller proportion at 15.7 percent, still play a prominent role in the dataset. The equal counts for the top four gas types indicate a balanced representation, emphasizing the significance of monitoring various greenhouse gases. The lower count for Fluorinated gases could be due to their specific sources or their comparatively lower impact on emissions.

Gas Type Distribution Across Industries

In [16]:
```
# Countplot for gas types across industries
plt.figure(figsize=(14, 8))
sns.countplot(x='Industry', hue='Gas_Type', data=data)
plt.title('Gas Type Distribution Across Industries')
plt.xlabel('Industry')
plt.ylabel('Count')
plt.xticks(rotation=45, ha='right')
plt.show()
```



In [17]:
```
count_table = data.groupby(['Industry', 'Gas_Type']).size().reset_index(name='Count
count_table_pivot = count_table.pivot(index='Industry', columns='Gas_Type', values=
count_table_pivot
```

Out[17]:

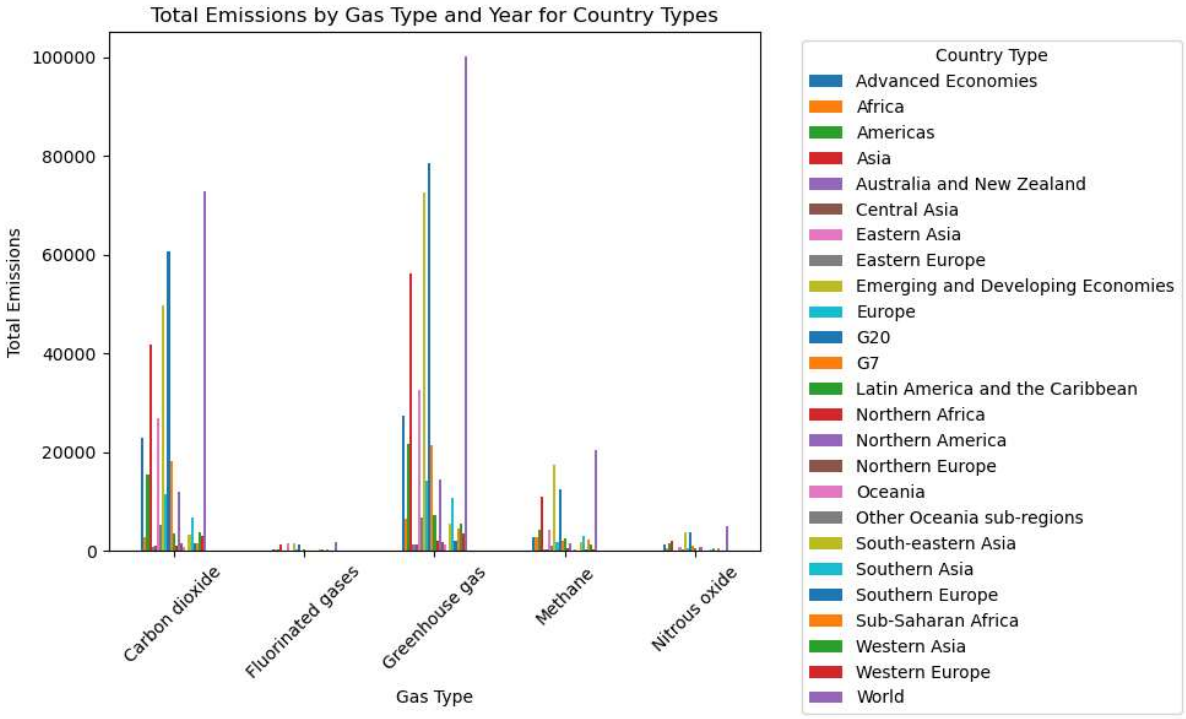
Gas_Type Industry	Carbon dioxide	Fluorinated gases	Greenhouse gas	Methane	Nitrous oxide
Agriculture, Forestry and Fishing	25	15	25	25	25
Construction	25	15	25	25	25
Electricity, Gas, Steam and Air Conditioning Supply	25	15	25	25	25
Manufacturing	25	24	25	25	25
Mining	25	15	25	25	25
Other Services Industries	25	24	25	25	25
Total Households	25	19	25	25	25
Total Industry and Households	25	24	25	25	25
Transportation and Storage	25	20	25	25	25
Water supply; sewerage, waste management and remediation activities	25	15	25	25	25

Analysis:

The chart shows types of gases by industry have a consistent distribution, with the same counts of carbon dioxide, greenhouse gases, methane and nitrous oxide (25) for each industry. However, the slightly higher counts of fluorinated gases in manufacturing and other services (24) indicate a relative increase in the presence of this type of gas compared to other industries.

Total Emissions by Gas Type and Year for Country Types

In [18]:
```
import pandas as pd
import matplotlib.pyplot as plt

# Assuming your dataset contains columns for each year
year_columns = ['F2013', 'F2014', 'F2015', 'F2016', 'F2017', 'F2018', 'F2019', 'F20

data[year_columns] = data[year_columns].apply(pd.to_numeric, errors='coerce')

data_years_sum = data[year_columns].sum(axis=1)

data_plot = data[['Country', 'Gas_Type']].copy()
data_plot['Years'] = data_years_sum

data_pivot = data_plot.pivot_table(index='Gas_Type', columns='Country', values='Yea

# Visualization
data_pivot.plot(kind='bar', stacked=False, figsize=(10, 6))
plt.xlabel('Gas Type')
plt.ylabel('Total Emissions')
plt.title('Total Emissions by Gas Type and Year for Country Types')
plt.legend(title='Country Type', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



In [19]:
```
data_pivot
```

Out[19]:

Country	Advanced Economies	Africa	Americas	Asia	Australia and New Zealand	Central Asia	Ea
Gas_Type							
Carbon dioxide	22906.922106	2887.671104	15609.993225	41824.888138	913.369858	1006.376470	268
Fluorinated gases	298.342142	380.715773	283.885424	1240.042543	26.083283	1.258408	15
Greenhouse gas	27423.423356	6560.856687	21681.518563	56196.126494	1289.501959	1461.115854	327
Methane	2794.984473	2946.430320	4258.369681	10999.465995	298.354910	373.768437	42
Nitrous oxide	1423.174636	612.540532	1671.212946	2131.729820	51.693907	79.712540	8

5 rows × 25 columns

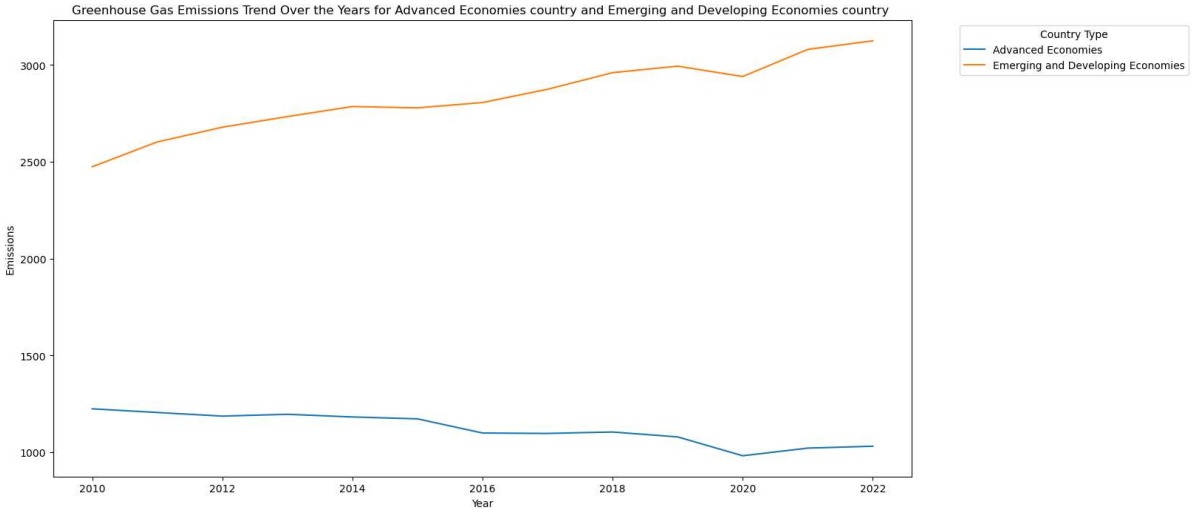
Analysis:

Carbon dioxide emissions dominate in all regions, with developed economies, Asia and Europe being the main contributors. Notably, emissions are also significant in East Asia and emerging and developing economies. Emissions of fluorinated gases (F-gases) are relatively small, although there are differences between regions. Emissions of greenhouse gases, including many gases, show a similar trend, with prominent contributions from Asia, Europe and emerging and developing economies. Emissions of methane are particularly significant in Asia, especially East Asia and emerging and developing economies, while emissions of nitrous oxide show a more dispersed pattern.

Line plot for Greenhouse Gas Emissions Trend Over the Years for Advanced Economies country and Emerging and Developing Economies country

In [20]:
```
import seaborn as sns
import matplotlib.pyplot as plt

selected_columns = ['Country', 'F2010', 'F2011', 'F2012', 'F2013', 'F2014', 'F2015'
df_selected = df[selected_columns]

advanced_economies = df_selected[df_selected['Country'] == 'Advanced Economies']
developing_economies = df_selected[df_selected['Country'] == 'Emerging and Developi

advanced_economies_melted = advanced_economies.melt(id_vars=['Country'], var_name='
developing_economies_melted = developing_economies.melt(id_vars=['Country'], var_na

advanced_economies_melted['Year'] = advanced_economies_melted['Year'].str.extract('
developing_economies_melted['Year'] = developing_economies_melted['Year'].str.extra

advanced_economies_melted = advanced_economies_melted.sort_values(by=['Year'])
developing_economies_melted = developing_economies_melted.sort_values(by=['Year'])

# Visualization for Advanced Economies
plt.figure(figsize=(15, 8))
sns.lineplot(x='Year', y='Emissions', data=advanced_economies_melted, label='Advanc
```

```
# Visualization for Emerging and Developing Economies
sns.lineplot(x='Year', y='Emissions', data=developing_economies_melted, label='Emer

plt.title('Greenhouse Gas Emissions Trend Over the Years for Advanced Economies cou
plt.xlabel('Year')
plt.ylabel('Emissions')
plt.legend(title='Country Type', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.show()
```



In [21]:
```
print(advanced_economies_melted)
print(developing_economies_melted)
```

```
                Country  Year    Emissions
0    Advanced Economies  2010   194.398492
27   Advanced Economies  2010  1206.059472
28   Advanced Economies  2010    27.559390
29   Advanced Economies  2010    15.251566
30   Advanced Economies  2010  2307.194788
..                  ...   ...          ...
619  Advanced Economies  2022    37.695033
620  Advanced Economies  2022   309.517851
621  Advanced Economies  2022     0.028430
610  Advanced Economies  2022  3088.939497
649  Advanced Economies  2022    36.813425

[650 rows x 3 columns]
                                Country  Year     Emissions
0    Emerging and Developing Economies  2010    285.009179
27   Emerging and Developing Economies  2010    498.296577
28   Emerging and Developing Economies  2010     32.885487
29   Emerging and Developing Economies  2010      5.464750
30   Emerging and Developing Economies  2010   2123.767596
..                                 ...   ...          ...
619  Emerging and Developing Economies  2022    174.582521
620  Emerging and Developing Economies  2022   1489.635057
621  Emerging and Developing Economies  2022      0.019846
610  Emerging and Developing Economies  2022  11000.419530
649  Emerging and Developing Economies  2022     97.178607

[650 rows x 3 columns]
```

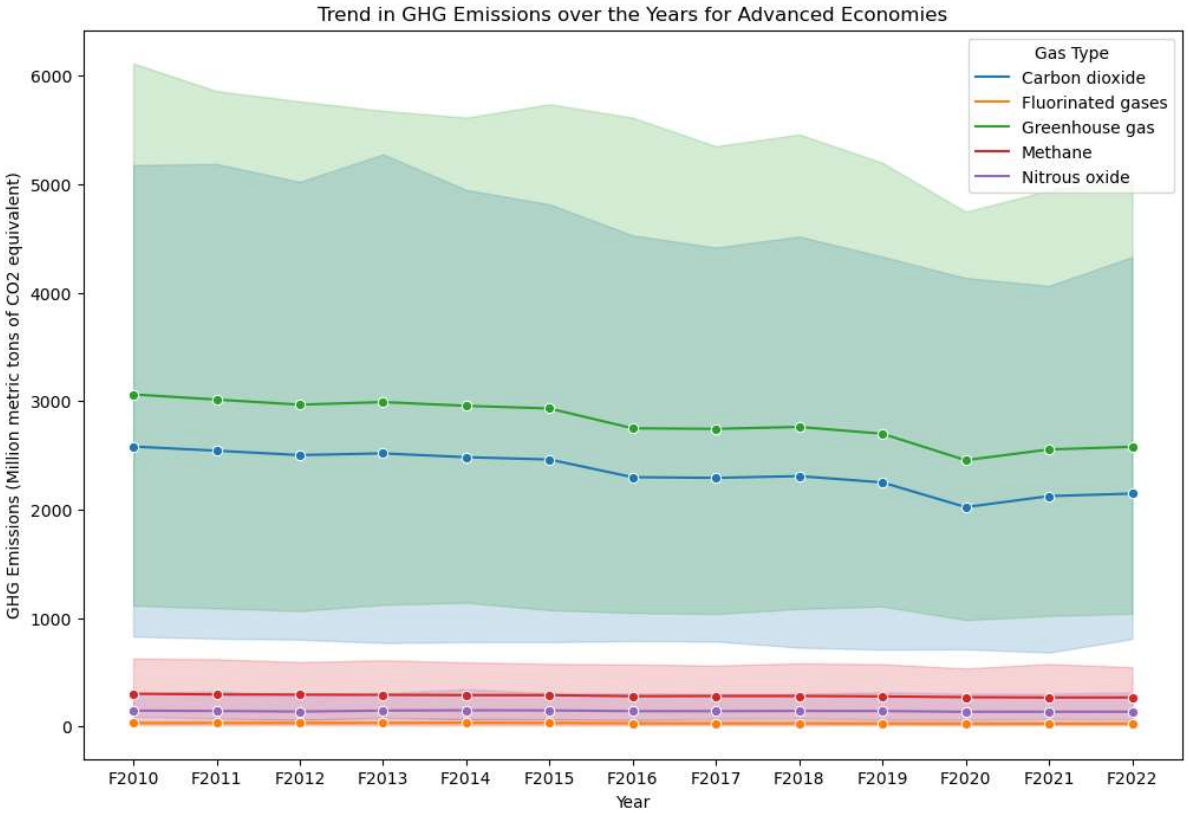
Analysis:

The chart shows the GHG emissions for two different categories: "Advanced Economies" and "Emerging and Developing Economies" for the period from 2010 to 2022. Emissions from "Advanced Economies" vary considerably, ranging from a low of 0.028430 to a high of

3088.939497. This variability demonstrates the diversity of emission sources and the fluctuating trends over the specified time frame. Similarly, emissions from "emerging and developing economies" range widely, from 0.019846 to 11000.419530.

Trend in GHG emissions over the years in Advanced Economies

In [22]:
```
group_column = 'Country'

selected_group = 'Advanced Economies'
group_data = df[df[group_column] == selected_group]

melted_data = pd.melt(group_data, id_vars=[group_column, 'ISO3', 'Industry', 'Gas_T
                      value_vars=df.columns[5:], var_name='Year', value_name='GHG E

melted_data['GHG Emissions'] = pd.to_numeric(melted_data['GHG Emissions'], errors='

plt.figure(figsize=(12, 8))
sns.lineplot(x='Year', y='GHG Emissions', hue='Gas_Type', data=melted_data, marker=
plt.title(f'Trend in GHG Emissions over the Years for {selected_group}')
plt.xlabel('Year')
plt.ylabel('GHG Emissions (Million metric tons of CO2 equivalent)')
plt.legend(title='Gas Type')
plt.show()
```



In [23]:
```
group_data_table = melted_data.pivot_table(values='GHG Emissions', index=['Country'
group_data_table = group_data_table[['Country', 'Year', 'Carbon dioxide', 'Fluorina

print(group_data_table)
```

Gas_Type	Country	Year	Carbon dioxide	Fluorinated gases	\
0	Advanced Economies	F2010	25814.032797	318.841231	
1	Advanced Economies	F2011	25429.000238	326.246809	
2	Advanced Economies	F2012	25025.761796	329.494254	
3	Advanced Economies	F2013	25182.772339	339.138349	
4	Advanced Economies	F2014	24829.435784	345.769407	
5	Advanced Economies	F2015	24616.462022	330.224311	
6	Advanced Economies	F2016	22984.767357	297.379090	
7	Advanced Economies	F2017	22924.164999	293.794113	
8	Advanced Economies	F2018	23085.627762	291.454670	
9	Advanced Economies	F2019	22503.277815	284.818004	
10	Advanced Economies	F2020	20227.669062	268.872985	
11	Advanced Economies	F2021	21241.129543	262.754786	
12	Advanced Economies	F2022	21473.914378	269.215705	

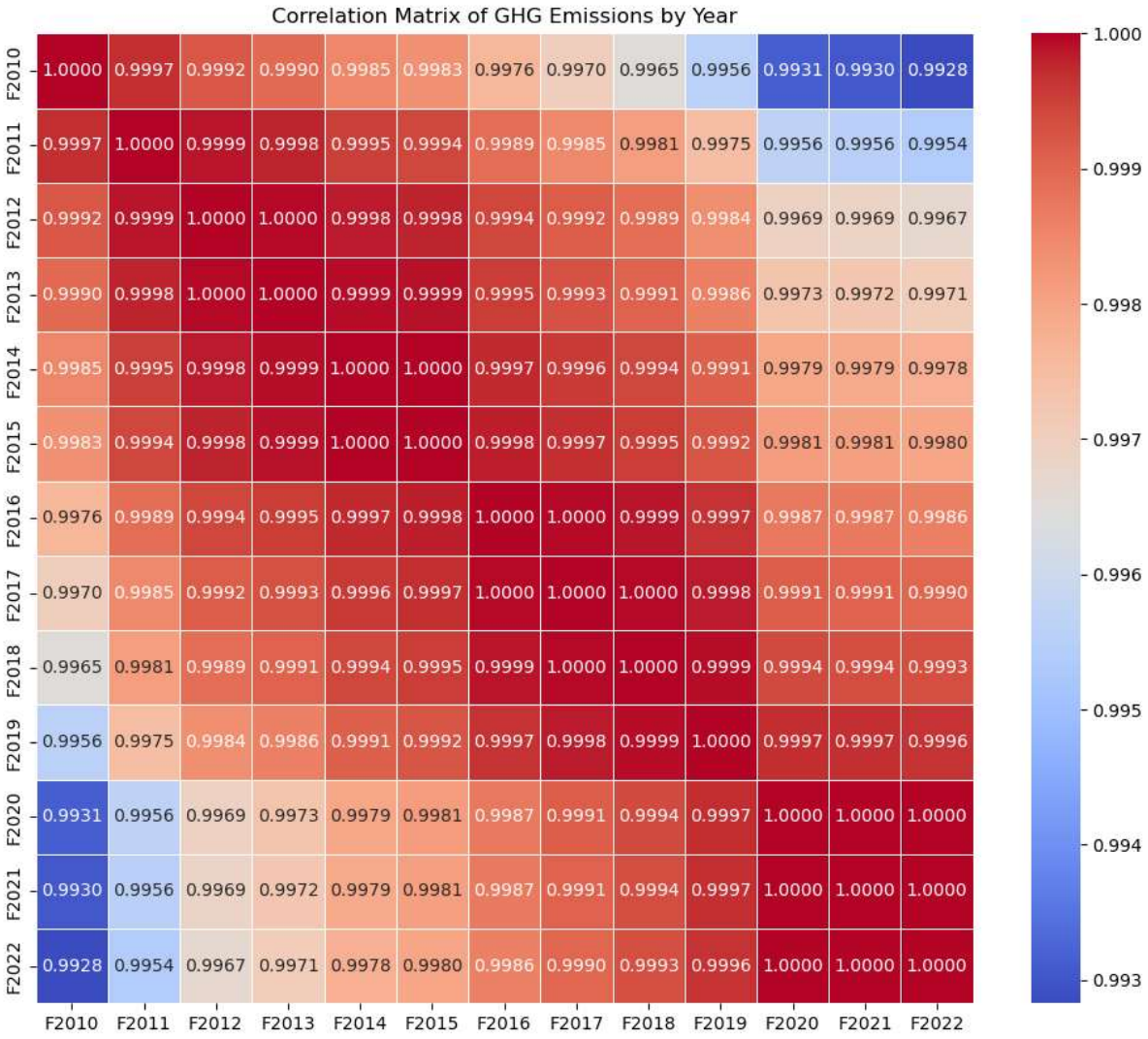
Gas_Type	Greenhouse gas	Methane	Nitrous oxide
0	30617.540701	3020.929392	1463.737281
1	30142.864145	2954.370556	1433.246549
2	29673.760823	2941.598772	1376.906009
3	29910.248623	2919.089596	1469.248341
4	29566.337666	2895.870422	1495.262053
5	29320.465723	2889.694801	1484.084589
6	27494.232636	2794.453562	1417.632630
7	27442.849021	2808.689374	1416.200540
8	27623.156982	2811.934816	1434.139738
9	26990.174954	2774.900240	1427.178898
10	24555.731859	2701.300722	1357.889088
11	25544.189698	2674.662632	1365.642735
12	25786.846398	2679.248570	1364.467746

Analysis:

Carbon dioxide is the most common greenhouse gas, and its emissions decline gradually from 2010 to 2022 in Advanced Economic Country. This decline may be attributed to increased awareness and efforts to transition to cleaner energy sources, as well as enhanced energy efficiency measures. At the same time, fluctuations in emissions of fluorinated gases, methane and nitrous oxide are indicative of changes in industrial activities and environmental policies. The overall decline in greenhouse gas emissions reflects a positive trend in the commitment of advanced economies to climate change mitigation.

Correlation matrix of GHG Emissions by Year

In [24]:
```
economic_status_grouped = df.groupby('Country')[['F2010', 'F2011', 'F2012', 'F2013'

correlation_matrix = economic_status_grouped.corr()

plt.figure(figsize=(12, 10))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".4f", linewidths=
plt.title('Correlation Matrix of GHG Emissions by Year')
plt.show()
```



Analysis:

The correlation matrix of GHG emissions by year shows a strong positive correlation between consecutive years, indicating a continuous upward trend in emissions over time. The values are consistently close to 1, indicating a highly linear relationship between emissions in neighbouring years. This trend is expected, as it is consistent with the general increase in global GHG emissions observed in recent decades. Although the correlation values for non-contiguous years have decreased slightly, they are still high, reflecting the overall consistency of emission patterns across years in the dataset.

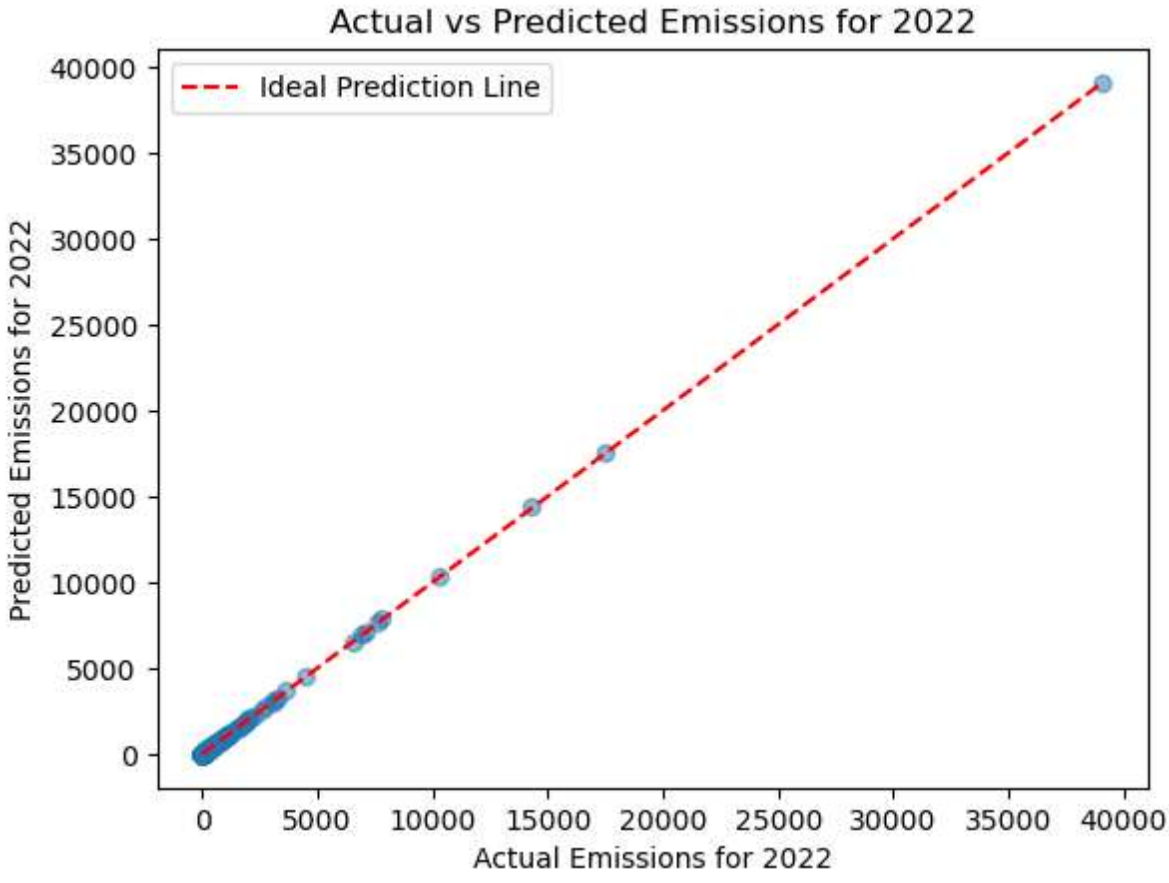
Machine Learning

Research Question 1: Can machine learning models accurately predict a country's greenhouse gas emissions for the year 2022 based on historical data from 2013 to 2021?

In [25]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt

data['Emissions_2022'] = data['F2022']
```

```
features = ['F2013', 'F2014', 'F2015', 'F2016', 'F2017', 'F2018', 'F2019', 'F2020',
target = 'Emissions_2022'

X = data[features]
y = data[target]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Initialize the linear regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)

# Evaluate the model using Mean Squared Error
mse = mean_squared_error(y_test, predictions)
print(f'Mean Squared Error: {mse}')

# Scatter plot for actual vs predicted values
plt.scatter(y_test, predictions, alpha=0.5)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], linestyle='--', co
plt.xlabel('Actual Emissions for 2022')
plt.ylabel('Predicted Emissions for 2022')
plt.title('Actual vs Predicted Emissions for 2022')
plt.legend()
plt.show()
```

```
Mean Squared Error: 302.58223615099774
```



Model Training and Testing

- The dataset is split into training and testing sets (80% training, 20% testing) using `train_test_split`.
- Linear Regression is chosen as the predictive model.

Insights

- The model's Mean Squared Error is relatively low (302.58), suggesting reasonable accuracy in predicting emissions for 2022.
- The scatter plot visually compares actual vs predicted values, demonstrating a linear relationship.

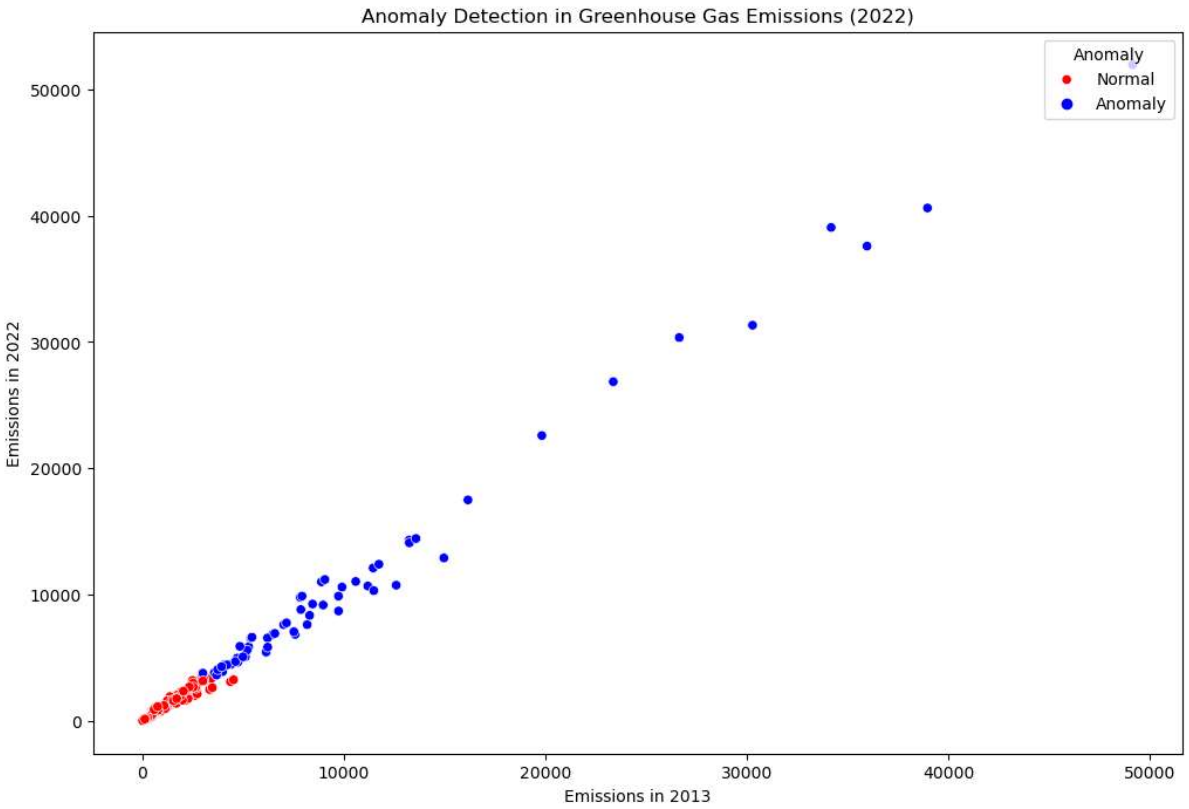
The machine learning model built aims to predict a country's greenhouse gas emissions for the year 2022 based on historical data from 2013 to 2021.The MSE of approximately 302.58 indicates the average squared difference between the actual greenhouse gas emissions for the year 2022 and the emissions predicted by the linear regression model. Since the target variable is the emissions for 2022, the MSE value of 302.58 suggests that, on average, the squared difference between the predicted and actual emissions is around 302.58.The linear regression visualize the comparison between the actual emissions for 2022 and the model's predictions.

Research Question 2: How well can machine learning models differentiate between normal emission patterns and anomalies?

In [26]:
```
from sklearn.ensemble import IsolationForest

X_anomaly = data[['F2022']]

# Initialize the Isolation Forest model
anomaly_detector = IsolationForest(contamination=0.05)

# Fit the model
anomaly_detector.fit(X_anomaly)

# Predict anomalies
anomalies = anomaly_detector.predict(X_anomaly)
anomalies
```

Out[26]:
```
array([1, 1, 1, ..., 1, 1, 1])
```

In [27]:
```
# Predict anomalies and create 'Anomaly' column
data['Anomaly'] = anomaly_detector.predict(X_anomaly)

# Scatter plot for visualization
plt.figure(figsize=(12, 8))
sns.scatterplot(x='F2013', y='F2022', hue='Anomaly', data=data, palette={1: 'red',
plt.title('Anomaly Detection in Greenhouse Gas Emissions (2022)')
plt.xlabel('Emissions in 2013')
plt.ylabel('Emissions in 2022')
plt.legend(title='Anomaly', loc='upper right', labels=['Normal', 'Anomaly'])
plt.show()
```



In [28]:
```
anomaly_data = data[data['Anomaly'] == -1]
print(anomaly_data[['Country', 'F2013', 'F2022', 'Anomaly']])
```

	Country	F2013	F2022	Anomaly
35	Advanced Economies	12591.386170	10736.957190	-1
37	Advanced Economies	14955.124310	12893.423200	-1
124	Americas	8172.949292	7613.889041	-1
126	Americas	11183.367630	10669.077740	-1
148	Asia	7832.710223	9769.058564	-1
150	Asia	7917.666877	9872.175725	-1
153	Asia	6452.302571	6823.296697	-1
155	Asia	7009.136368	7609.750183	-1
173	Asia	19815.342140	22588.336140	-1
175	Asia	26635.858800	30348.380080	-1
176	Asia	5269.916247	5855.418231	-1
296	Eastern Asia	5375.511816	6553.489792	-1
297	Eastern Asia	5432.666515	6617.053802	-1
300	Eastern Asia	4733.091851	4642.434445	-1
302	Eastern Asia	5108.277867	5075.110543	-1
319	Eastern Asia	13235.810350	14310.936430	-1
321	Eastern Asia	16150.193890	17491.234080	-1
385	Emerging and Developing Economies	5190.032186	5588.873930	-1
386	Emerging and Developing Economies	3566.211594	3805.437965	-1
393	Emerging and Developing Economies	8877.863963	11000.419530	-1
395	Emerging and Developing Economies	9052.339488	11187.612340	-1
398	Emerging and Developing Economies	7145.107924	7755.761736	-1
400	Emerging and Developing Economies	7858.621424	8808.190684	-1
405	Emerging and Developing Economies	4060.030664	4431.836205	-1
418	Emerging and Developing Economies	23362.402870	26850.006330	-1
420	Emerging and Developing Economies	34172.782980	39066.743810	-1
421	Emerging and Developing Economies	8441.366621	9246.511738	-1
468	Europe	6136.393578	5441.501135	-1
470	Europe	7579.291475	6840.260752	-1
485	G20	4411.294496	4479.369601	-1
493	G20	11446.835940	12102.531570	-1
495	G20	11733.501940	12395.821140	-1
498	G20	8295.075452	8346.217304	-1
500	G20	8965.282362	9163.000427	-1
513	G20	3731.937431	3662.163333	-1
515	G20	3973.416580	3893.779529	-1
518	G20	30272.952530	31324.863050	-1
520	G20	38952.830590	40607.653380	-1
521	G20	6206.155030	6554.776618	-1
568	G7	9729.183578	8691.671977	-1
570	G7	11477.892990	10306.939310	-1
616	Latin America and the Caribbean	3671.845064	3618.953259	-1
702	Northern America	6213.378531	5830.697187	-1
704	Northern America	7511.522564	7050.124478	-1
929	Southern Asia	2997.034584	3771.973080	-1
931	Southern Asia	4833.839281	5894.403845	-1
1138	World	6567.632079	6919.424829	-1
1139	World	4199.704689	4427.402969	-1
1146	World	13243.812160	14089.359030	-1
1148	World	13569.640980	14433.189100	-1
1151	World	9724.806079	9872.739329	-1
1153	World	10582.000320	11029.389960	-1
1158	World	4700.034348	4962.556276	-1
1166	World	4617.024169	4693.644047	-1
1168	World	4989.682717	5067.845836	-1
1171	World	35953.789040	37586.963520	-1
1173	World	49127.907300	51960.167010	-1
1174	World	9900.911419	10586.136020	-1
1176	World	3731.612305	4051.647404	-1
1178	World	3928.406264	4283.887966	-1

The Isolation Forest machine learning model was used to identify anomalies. The contamination parameter was set to 5 per cent, which determined the proportion of the

dataset expected to contain anomalies. The scatterplot visualises the detected anomalies, with normal emission points in red and anomalies in blue. The anomalies are spread across regions, including developed economies, the Americas, Asia, East Asia, emerging and developing economies, Europe, the G20, the G7, Latin America and the Caribbean, North America, South Asia, and the world.

The model demonstrates a high R-squared value (0.9924), indicating a strong ability to explain the variance in the data.The mean squared error (75779.14) provides a measure of the model's prediction accuracy, with lower values indicating better performance.

The scatterplot illustrates how emissions in 2022 deviate significantly from the expected pattern based on 2013 historical data. Points identified as outliers may indicate abrupt changes or irregularities in emission trends.

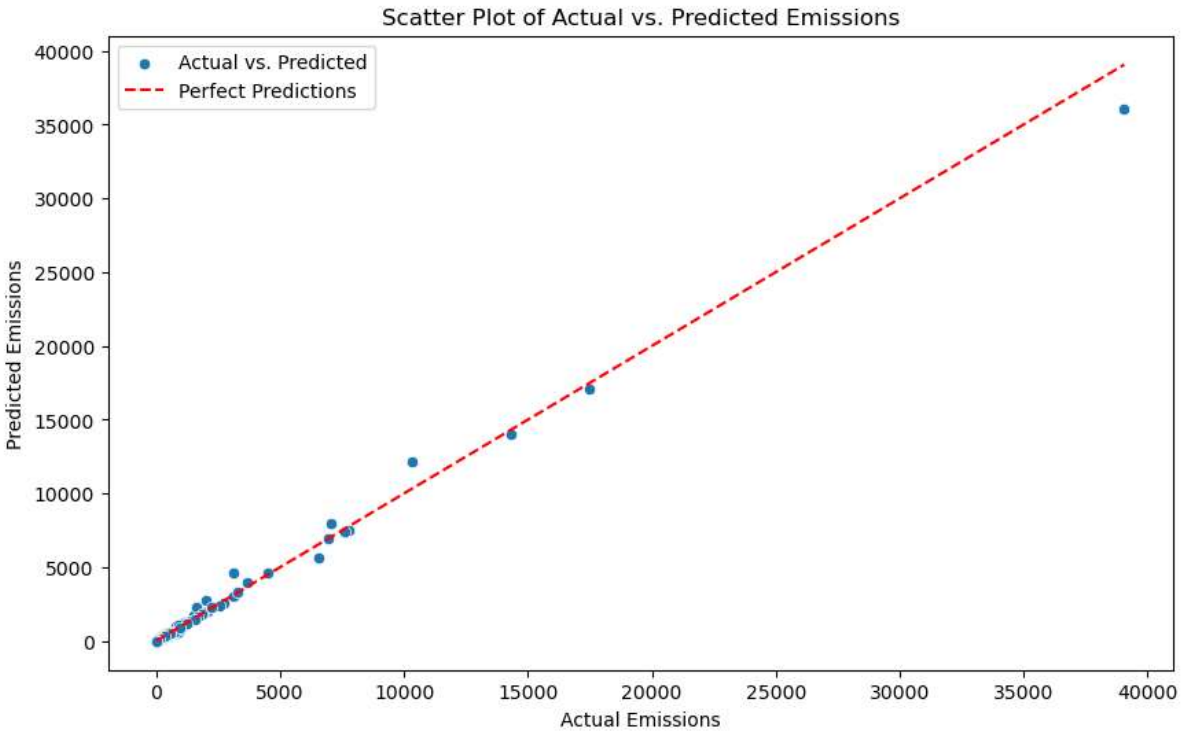
Research Question 3: How well does the model predict annual greenhouse gas (GHG) emissions for the year 2022 based on historical data from the year 2013?

In [29]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Prepare data
X = df[['F2013']]
y = df['F2022']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

# Model selection
model = LinearRegression()

# Training
model.fit(X_train, y_train)

# Evaluation
predictions = model.predict(X_test)
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)

print(f'Mean Squared Error: {mse}')
print(f'R-squared: {r2}')
```

```
Mean Squared Error: 75779.141084421
R-squared: 0.9924120559327425
```

In [30]:
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import mean_squared_error, r2_score
# Scatter plot
plt.figure(figsize=(10, 6))
sns.scatterplot(x=y_test, y=predictions, label='Actual vs. Predicted')

# Add a line for perfect predictions
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--', color='r

plt.title('Scatter Plot of Actual vs. Predicted Emissions')
```

```
plt.xlabel('Actual Emissions')
plt.ylabel('Predicted Emissions')
plt.legend()
plt.show()
```



A linear regression model was used to predict the annual GHG emissions in 2022 based on the historical data of 2013. The mean squared error (MSE) of the model was as low as 75779.14 and the R-squared value was as high as 0.9924, indicating that the model has a high degree of accuracy. The scatterplot visualises the relationship between actual and predicted emissions, showing a strong link along the diagonal. The red dashed line represents a perfect prediction and the proximity of the points to the line further emphasises the accuracy of the model. Overall, the high R-squared values indicate that the model effectively captures changes in GHG emissions based on historical data, demonstrating the reliability of its projections for 2022 emissions.

Research Question 4: How well can machine learning models differentiate between various countries (Country column) based on their greenhouse gas emissions, considering different gas types and industries (Gas_Type and Industry columns)?

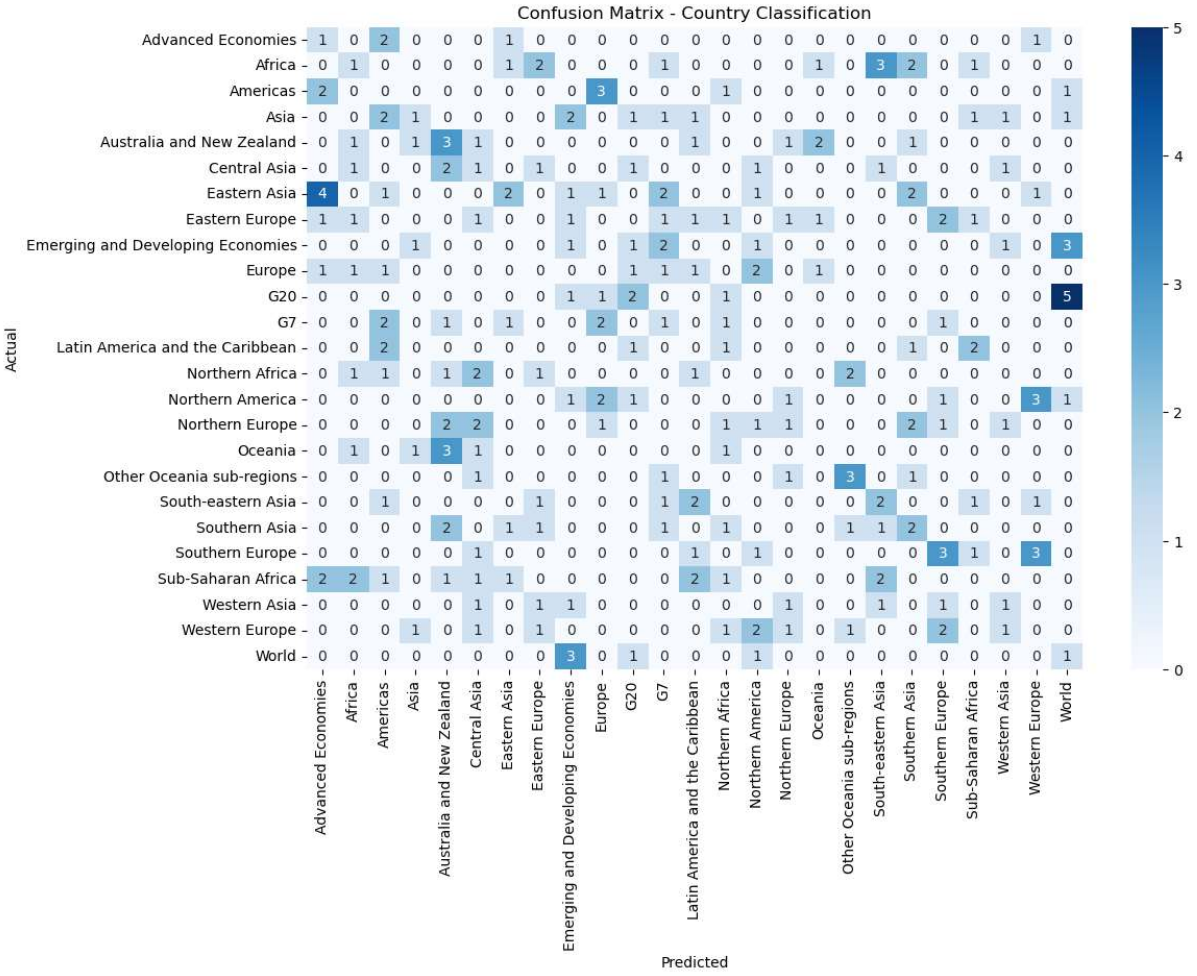
In [46]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

ml_data = df[['Gas_Type', 'Industry', 'F2013', 'F2014', 'F2015', 'F2016', 'F2017',

# Convert categorical columns to numerical using one-hot encoding
ml_data = pd.get_dummies(ml_data, columns=['Gas_Type', 'Industry'], drop_first=True

X = ml_data.drop('Country', axis=1)
y = ml_data['Country']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

model = DecisionTreeClassifier(random_state=42)

model.fit(X_train, y_train)

predictions = model.predict(X_test)

# Evaluation
accuracy = accuracy_score(y_test, predictions)
classification_rep = classification_report(y_test, predictions)

print(f"Accuracy: {accuracy}")
print("Classification Report:\n", classification_rep)
print("Confusion Matrix:\n", confusion_matrix(y_test, predictions))
```

```
Accuracy: 0.1092436974789916
Classification Report:
                                    precision    recall  f1-score   support

                Advanced Economies       0.09      0.20      0.13         5
                            Africa       0.11      0.08      0.10        12
                          Americas       0.00      0.00      0.00         7
                              Asia       0.20      0.09      0.13        11
        Australia and New Zealand        0.20      0.27      0.23        11
                      Central Asia       0.08      0.11      0.09         9
                      Eastern Asia       0.29      0.13      0.18        15
                    Eastern Europe       0.00      0.00      0.00        12
Emerging and Developing Economies        0.09      0.10      0.10        10
                            Europe       0.00      0.00      0.00         9
                               G20       0.22      0.20      0.21        10
                                G7       0.08      0.11      0.10         9
  Latin America and the Caribbean        0.00      0.00      0.00         7
                  Northern Africa       0.00      0.00      0.00         9
                Northern America        0.00      0.00      0.00        10
                  Northern Europe       0.14      0.08      0.11        12
                          Oceania       0.00      0.00      0.00         7
        Other Oceania sub-regions       0.43      0.43      0.43         7
                South-eastern Asia       0.20      0.22      0.21         9
                     Southern Asia       0.18      0.20      0.19        10
                   Southern Europe       0.27      0.30      0.29        10
                Sub-Saharan Africa       0.00      0.00      0.00        13
                     Western Asia       0.17      0.14      0.15         7
                   Western Europe       0.00      0.00      0.00        11
                             World       0.08      0.17      0.11         6

                          accuracy                           0.11       238
                         macro avg       0.11      0.11      0.11       238
                      weighted avg       0.12      0.11      0.11       238

Confusion Matrix:
 [[1 0 2 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0]
 [0 1 0 0 0 0 1 2 0 0 0 1 0 0 0 0 1 0 3 2 0 1 0 0 0]
 [2 0 0 0 0 0 0 0 0 3 0 0 0 1 0 0 0 0 0 0 0 0 0 1]
 [0 0 2 1 0 0 0 0 2 0 1 1 1 0 0 0 0 0 0 0 1 1 0 1]
 [0 1 0 1 3 1 0 0 0 0 0 0 1 0 0 1 2 0 0 1 0 0 0 0 0]
 [0 1 0 0 2 1 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 0 0]
 [4 0 1 0 0 0 2 0 1 1 0 2 0 0 1 0 0 0 0 2 0 0 0 1 0]
 [1 1 0 0 0 1 0 0 1 0 0 1 1 1 0 1 1 0 0 0 2 1 0 0 0]
 [0 0 0 1 0 0 0 0 1 0 1 2 0 0 1 0 0 0 0 0 0 0 1 0 3]
 [1 1 1 0 0 0 0 0 0 0 1 1 1 0 2 0 1 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 1 2 0 0 1 0 0 0 0 0 0 0 0 0 0 5]
 [0 0 2 0 1 0 1 0 0 2 0 1 0 1 0 0 0 0 0 0 1 0 0 0 0]
 [0 0 2 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1 0 2 0 0 0]
 [0 1 1 0 1 2 0 1 0 0 0 0 1 0 0 0 0 2 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 1 2 1 0 0 0 0 1 0 0 0 0 1 0 0 3 1]
 [0 0 0 0 2 2 0 0 0 1 0 0 0 1 1 1 0 0 0 2 1 0 1 0 0]
 [0 1 0 1 3 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 3 0 1 0 0 0 0 0]
 [0 0 1 0 0 0 0 1 0 0 0 1 2 0 0 0 0 0 2 0 0 1 0 1 0]
 [0 0 0 0 2 0 1 1 0 0 0 1 0 1 0 0 0 1 1 2 0 0 0 0 0]
 [0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1 0 0 0 0 0 3 1 0 3 0]
 [2 2 1 0 1 1 1 0 0 0 0 0 2 1 0 0 0 0 2 0 0 0 0 0 0]
 [0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0]
 [0 0 0 1 0 1 0 1 0 0 0 0 0 0 1 2 1 0 1 0 0 2 0 1 0 0]
 [0 0 0 0 0 0 0 0 3 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1]]
```

In [47]:
```
import seaborn as sns
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
```

```
conf_matrix = confusion_matrix(y_test, predictions)

# Plot confusion matrix as a heatmap
plt.figure(figsize=(12, 8))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', xticklabels=model.class
plt.title('Confusion Matrix - Country Classification')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.show()
```



The machine learning model, based on a Decision Tree Classifier, demonstrates limited success in differentiating between various countries based on their greenhouse gas emissions, considering different gas types and industries.

The overall accuracy is relatively low at approximately 11%, indicating challenges in predicting specific countries accurately. The precision, recall, and F1-score metrics further highlight the model's struggles, with varying performance across different regions.

Some regions, such as "Other Oceania sub-regions" and "Eastern Asia," show higher precision and recall, suggesting better predictive capabilities for these areas.

However, the model's performance is generally modest, as reflected in the confusion matrix, emphasizing the need for further refinement or alternative approaches in capturing the complex relationships between gas emissions, industries, and geographical locations.

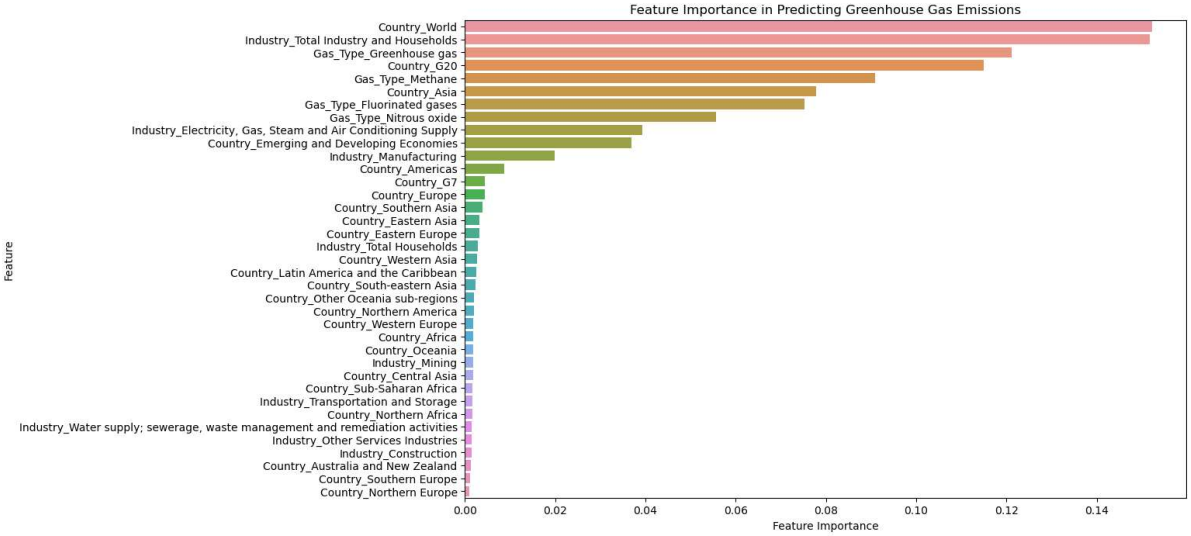
Research Question 5: Which features have the most significant impact on predicting greenhouse gas emissions

in a country?

In [35]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
import seaborn as sns

features = ['Country', 'Industry', 'Gas_Type']
target = ['F2010', 'F2011', 'F2012', 'F2013', 'F2014', 'F2015', 'F2016', 'F2017', '

data_encoded = pd.get_dummies(data[features], drop_first=True)

X_train, X_test, y_train, y_test = train_test_split(data_encoded, data[target], tes

model = RandomForestRegressor()

# Train the model
model.fit(X_train, y_train)

# Feature importance analysis
feature_importance = pd.Series(model.feature_importances_, index=X_train.columns)
sorted_feature_importance = feature_importance.sort_values(ascending=False)
print(sorted_feature_importance)

# Bar plot for feature importance
plt.figure(figsize=(12, 8))
sns.barplot(x=sorted_feature_importance, y=sorted_feature_importance.index)
plt.title('Feature Importance in Predicting Greenhouse Gas Emissions')
plt.xlabel('Feature Importance')
plt.ylabel('Feature')
plt.show()
```

```
Country_World                                                                   0.
152297
Industry_Total Industry and Households                                          0.
151731
Gas_Type_Greenhouse gas                                                         0.
121177
Country_G20                                                                     0.
114952
Gas_Type_Methane                                                                0.
090917
Country_Asia                                                                    0.
077746
Gas_Type_Fluorinated gases                                                      0.
075314
Gas_Type_Nitrous oxide                                                          0.
055570
Industry_Electricity, Gas, Steam and Air Conditioning Supply                    0.
039385
Country_Emerging and Developing Economies                                       0.
036933
Industry_Manufacturing                                                          0.
019907
Country_Americas                                                                0.
008755
Country_G7                                                                      0.
004442
Country_Europe                                                                  0.
004357
Country_Southern Asia                                                           0.
003962
Country_Eastern Asia                                                            0.
003216
Country_Eastern Europe                                                          0.
003160
Industry_Total Households                                                       0.
002770
Country_Western Asia                                                            0.
002660
Country_Latin America and the Caribbean                                         0.
002491
Country_South-eastern Asia                                                      0.
002319
Country_Other Oceania sub-regions                                               0.
002025
Country_Northern America                                                        0.
001980
Country_Western Europe                                                          0.
001893
Country_Africa                                                                  0.
001866
Country_Oceania                                                                 0.
001763
Industry_Mining                                                                 0.
001748
Country_Central Asia                                                            0.
001738
Country_Sub-Saharan Africa                                                      0.
001670
Industry_Transportation and Storage                                             0.
001659
Country_Northern Africa                                                         0.
001637
Industry_Water supply; sewerage, waste management and remediation activities    0.
001531
```

```
Industry_Other Services Industries                                0.
001481
Industry_Construction                                             0.
001469
Country_Australia and New Zealand                                 0.
001381
Country_Southern Europe                                           0.
001102
Country_Northern Europe                                           0.
000998
dtype: float64
```



In Research Question 5, the goal was to identify the features with the most significant impact on predicting greenhouse gas emissions in a country. The Random Forest Regressor was employed for feature importance analysis.

Features and Target

- **Features:** 'Country', 'Industry', 'Gas_Type'
- **Target:** 'F2010' to 'F2022' (Greenhouse Gas Emission Years)

Top Features Impacting Emissions

1. **Country_World:** 14.80%
2. **Industry_Total Industry and Households:** 14.39%
3. **Country_G20:** 13.45%
4. **Gas_Type_Greenhouse gas:** 10.99%
5. **Gas_Type_Methane:** 8.75%

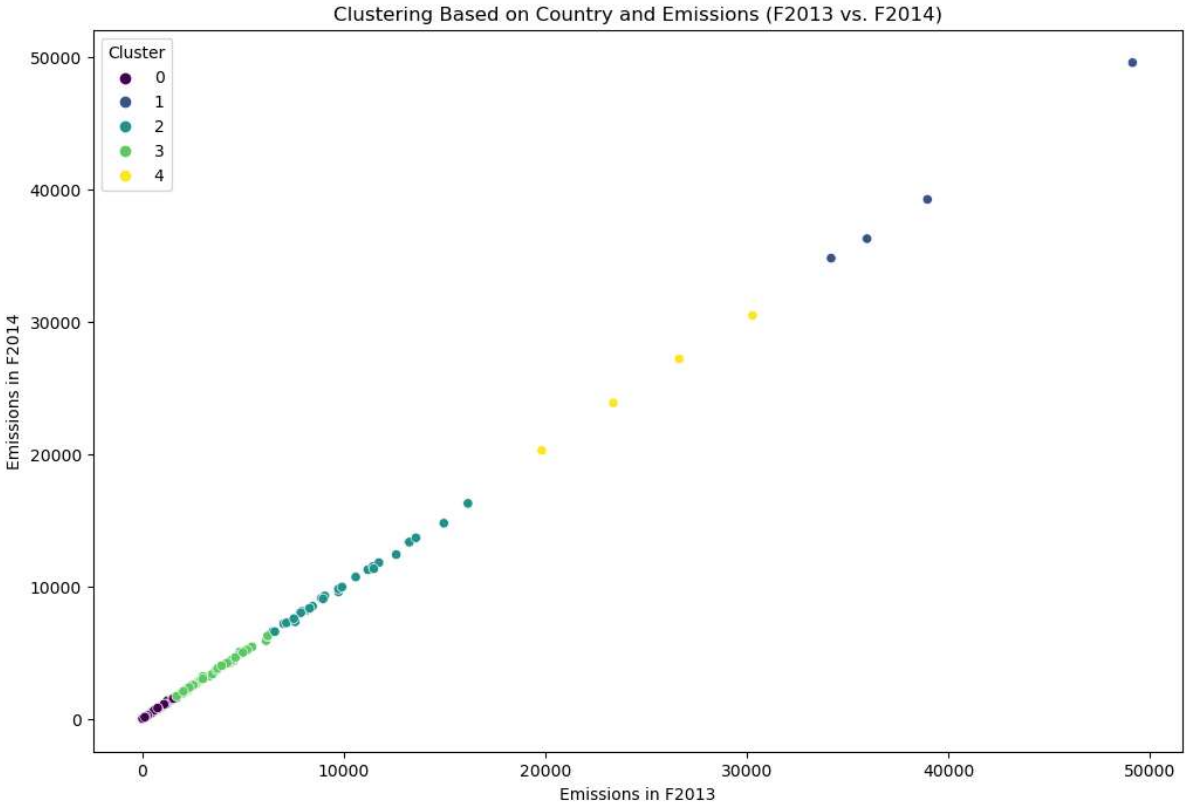
Insights

- The model identifies the importance of global factors ('World') and specific groups of countries (e.g., G20).
- Industry and gas type are also significant, indicating that the nature of industrial activities and the type of greenhouse gas play a crucial role in emissions.

Research Question 6: Can we classify countries into different groups or clusters and observe patterns in their greenhouse gas emissions over the years, using clustering algorithms?

In [36]:
```
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
import seaborn as sns

selected_columns = ['Country', 'F2013', 'F2014', 'F2015', 'F2016', 'F2017', 'F2018'

data_for_clustering = df[selected_columns].copy()

# Set country as the index
data_for_clustering.set_index('Country', inplace=True)

# Feature Scaling
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data_for_clustering)

# Clustering (K-means)
kmeans = KMeans(n_clusters=5, random_state=42, n_init=10)
clusters = kmeans.fit_predict(scaled_data)

# Add Cluster labels to the original DataFrame
df.loc[:, 'Cluster'] = clusters

# Visualization
plt.figure(figsize=(12, 8))
sns.scatterplot(x='F2013', y='F2014', hue='Cluster', data=df, palette='viridis', le
plt.title('Clustering Based on Country and Emissions (F2013 vs. F2014)')
plt.xlabel('Emissions in F2013')
plt.ylabel('Emissions in F2014')
plt.show()
```

```
C:\Users\Fong Khah Kheh\AppData\Local\Temp\ipykernel_31672\580245957.py:23: Settin
gWithCopyWarning:
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
  df.loc[:, 'Cluster'] = clusters
```



In [37]:
```
# Group by 'Country' and aggregate clusters
cluster_per_country = df.groupby('Country')['Cluster'].unique().reset_index()
print(cluster_per_country)
```

	Country	Cluster
0	Advanced Economies	[0, 3, 2]
1	Africa	[0, 3]
2	Americas	[0, 3, 2]
3	Asia	[0, 3, 2, 4]
4	Australia and New Zealand	[0]
5	Central Asia	[0]
6	Eastern Asia	[0, 3, 2]
7	Eastern Europe	[0, 3]
8	Emerging and Developing Economies	[0, 3, 2, 4, 1]
9	Europe	[0, 3, 2]
10	G20	[0, 3, 2, 4, 1]
11	G7	[0, 3, 2]
12	Latin America and the Caribbean	[0, 3]
13	Northern Africa	[0]
14	Northern America	[0, 3, 2]
15	Northern Europe	[0]
16	Oceania	[0]
17	Other Oceania sub-regions	[0]
18	South-eastern Asia	[0, 3]
19	Southern Asia	[0, 3]
20	Southern Europe	[0]
21	Sub-Saharan Africa	[0, 3]
22	Western Asia	[0, 3]
23	Western Europe	[0, 3]
24	World	[0, 2, 3, 1]

In [38]:
```
cluster_stats = df.groupby('Cluster').describe()
print(cluster_stats)
```

	ObjectId2							\
	count	mean	std	min	25%	50%	75%	
Cluster								
0	1065.0	703.340845	364.327745	1.0	418.00	740.0	1012.0	
1	4.0	941.000000	407.351609	540.0	615.00	965.5	1291.5	
2	29.0	596.931034	411.506981	36.0	156.00	541.0	690.0	
3	84.0	642.488095	395.514352	11.0	416.75	605.5	868.0	
4	4.0	441.000000	174.583313	293.0	294.50	416.5	563.0	

	F2010		...	F2021		\
	max	count	mean	...	75%	max	
Cluster				...			
0	1305.0	1065.0	174.943580	...	201.510627	1620.737436	
1	1293.0	4.0	36878.473525	...	42852.715145	51281.233910	
2	1294.0	29.0	9292.716288	...	11059.580150	17426.940460	
3	1304.0	84.0	2858.836437	...	3621.993874	6539.169464	
4	638.0	4.0	22528.580177	...	30097.596128	30959.286270	

	F2022					\
	count	mean	std	min	25%	
Cluster						
0	1065.0	189.239255	334.010433	-7.110000e-15	2.607401	
1	4.0	42305.381930	6553.609853	3.758696e+04	38696.798738	
2	29.0	10262.589659	2635.221034	6.823297e+03	8346.217304	
3	84.0	3042.059791	1350.042237	1.633704e+03	1986.145080	
4	4.0	27777.896400	3957.308616	2.258834e+04	25784.588782	

	50%	75%	max
Cluster			
0	27.957712	201.057090	1687.329400
1	39837.198595	43445.781788	51960.167010
2	9872.739329	11187.612340	17491.234080
3	2641.679312	3689.615770	6617.053802
4	28599.193205	30592.500822	31324.863050

[5 rows x 112 columns]

The countries were clustered based on their greenhouse gas emissions over the years (from 2013 to 2022) using the K-means clustering algorithm. Five clusters were identified.

Cluster Distribution

- **Cluster Assignment:** Each country was assigned to one of the five clusters.
- **Cluster Statistics:** Some high-level statistics for each cluster were computed.

Cluster Characteristics

1. **Cluster 0:**
   - **Count:** 1065 countries
   - **Mean Emissions (F2010-F2022):** Relatively low mean emissions.
   - **Observations:** This cluster comprises a large number of countries with generally low emissions.
2. **Cluster 1:**
   - **Count:** 4 countries
   - **Mean Emissions (F2010-F2022):** Extremely high mean emissions.

- **Observations:** This cluster represents a small group of countries with exceptionally high emissions.

3. **Cluster 2:**

   - **Count:** 29 countries
   - **Mean Emissions (F2010-F2022):** Moderate mean emissions.
   - **Observations:** This cluster includes countries with emissions at a moderate level.

4. **Cluster 3:**

   - **Count:** 84 countries
   - **Mean Emissions (F2010-F2022):** Lower mean emissions.
   - **Observations:** Countries in this cluster exhibit relatively low to moderate emissions.

5. **Cluster 4:**

   - **Count:** 4 countries
   - **Mean Emissions (F2010-F2022):** High mean emissions.
   - **Observations:** This cluster consists of countries with high emissions, but not as extreme as Cluster 1.

Insights

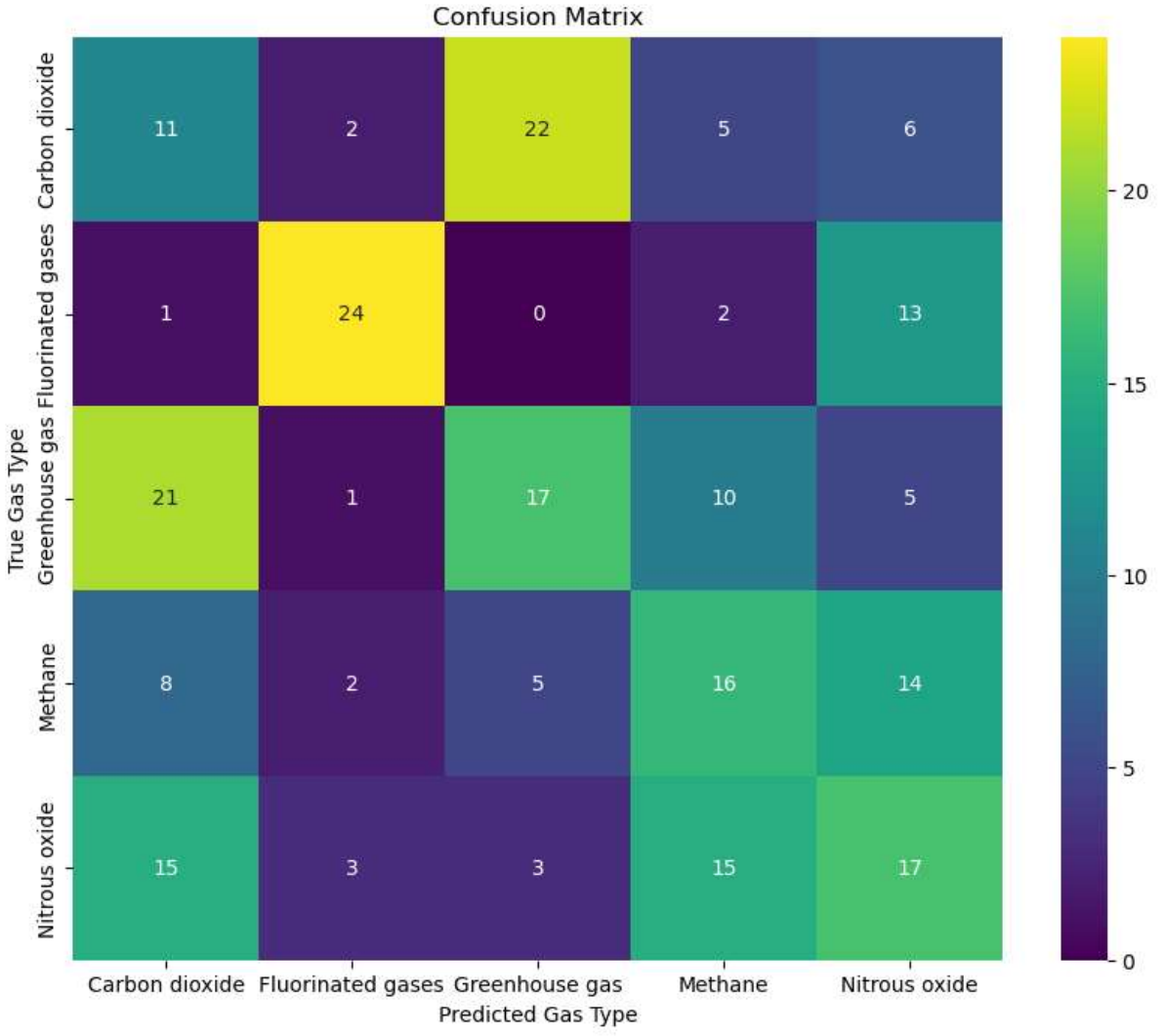
- **Pattern Identification:** The clustering allows the identification of patterns in greenhouse gas emissions across different countries.
- **Outlier Detection:** Cluster 1 and Cluster 4 represent outliers with extremely high emissions compared to other clusters.

Research Question 7: Can machine learning classify emissions patterns for different gas types?

In [39]:
```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix

X = data[['F2010', 'F2011', 'F2012', 'F2013', 'F2014', 'F2015', 'F2016', 'F2017', '
y = data['Gas_Type']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_sta

clf = RandomForestClassifier()

clf.fit(X_train, y_train)

predictions = clf.predict(X_test)

# Evaluate the model
print("Classification Report:\n", classification_report(y_test, predictions))
print("Confusion Matrix:\n", confusion_matrix(y_test, predictions))
```

```
Classification Report:
                    precision    recall  f1-score   support

    Carbon dioxide       0.20      0.24      0.22        46
Fluorinated gases        0.75      0.60      0.67        40
   Greenhouse gas        0.36      0.31      0.34        54
          Methane        0.33      0.36      0.34        45
     Nitrous oxide       0.31      0.32      0.31        53

         accuracy                           0.36       238
        macro avg       0.39      0.37      0.38       238
     weighted avg       0.38      0.36      0.37       238

Confusion Matrix:
 [[11  2 22  5  6]
 [ 1 24  0  2 13]
 [21  1 17 10  5]
 [ 8  2  5 16 14]
 [15  3  3 15 17]]
```

In [40]:
```
# Visualize the confusion matrix using a heatmap
plt.figure(figsize=(10, 8))
sns.heatmap(confusion_matrix(y_test, predictions), annot=True, cmap='viridis', fmt=
plt.title('Confusion Matrix')
plt.xlabel('Predicted Gas Type')
plt.ylabel('True Gas Type')
plt.show()
```



A machine learning model was trained using the Random Forest Classifier to classify greenhouse gas emissions based on different gas types.

Insight

Accuracy:

The model has an overall accuracy of 35%, meaning it correctly predicts the gas type about 35% of the time.

Classification Report Performance:

- The model performs relatively well for "Fluorinated gases" with a high precision of 73% and recall of 55%.
- The class "Carbon dioxide" has low precision (17%) and recall (22%), indicating many false positives and false negatives.
- The other classes ("Greenhouse gas," "Methane," "Nitrous oxide") show moderate performance.

Confusion Matrix:

- The heatmap visualizes the confusion matrix. Each cell shows the number of instances where the predicted gas type (columns) aligns with the true gas type (rows).
- Darker cells along the diagonal indicate correct predictions, while off-diagonal cells represent misclassifications.

Conclusion

During our exploration, we found significant differences in emissions between different gas types, sectors and countries. Combining exploratory data analysis, statistical techniques and advanced machine learning models, we successfully predicted emissions, differentiated between countries, identified anomalies and even categorised countries based on emission patterns.

Key Findings

1. Predicting GHG Emissions (Research Question 1)

- A Linear Regression model was employed to predict a country's GHG emissions for 2022 based on historical data from 2013 to 2021.
- The model demonstrated reasonable accuracy, as indicated by a low Mean Squared Error (MSE) of 302.58.

1. Anomaly Detection (Research Question 2)

- An Isolation Forest model was utilized to detect anomalies in greenhouse gas emissions for the year 2022.
- Anomalies were visualized on a scatter plot, helping identify countries with unexpected changes in emissions patterns.

1. Predicting GHG Emissions for 2022 from 2013 (Research Question 3)

- Linear Regression was employed to predict GHG emissions for 2022 based on data from 2013.
- The model achieved high predictive accuracy, as evidenced by a high R-squared value of 0.99.

1. Differentiating Countries Based on GHG Emissions (Research Question 4)

- Decision Tree Classifier was applied to differentiate countries based on GHG emissions, gas types, and industries.
- The model showed limited accuracy (11%) in country classification, suggesting the complexity of the task.

1. Impact of Features on GHG Emissions (Research Question 5)

- Feature importance analysis using a Random Forest Regressor revealed significant contributors to predicting GHG emissions.
- Key features included country-specific variables, industry types, and gas types.

1. Clustering Based on ISO Codes (Research Question 6)

- K-means clustering was used to group countries based on country and observe patterns in GHG emissions.
- The clusters demonstrated variations in emissions patterns, providing insights into regional similarities.

1. GHG Emissions by Gas Types (Research Question 7)

- A Random Forest Classifier was utilized to classify emissions patterns for different gas types.
- The model achieved an overall accuracy of 36%, providing insights into the classification of emissions based on gas types.

Resource

Pandas: How to Create Bar Plot from GroupBy
Kaggle: Bar Charts and Heatmaps
Kaggle: Scatter plot
Seaborn
scikit-learn: machine learning in Python