```
from google.colab import files
import pandas as pd

files.upload()
df=pd.read_csv('climate_change_dataset.csv')
print(df)
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving climate_change_dataset.csv to climate_change_dataset.csv
     Year    Country  Avg Temperature (°C)  CO2 Emissions (Tons/Capita)  \
0    2006         UK                   8.9                          9.3
1    2019        USA                  31.0                          4.8
2    2014     France                  33.9                          2.8
3    2010  Argentina                   5.9                          1.8
4    2007    Germany                  26.9                          5.6
..    ...        ...                   ...                          ...
995  2019      India                  23.6                          8.0
996  2000         UK                  21.8                         10.0
997  2019  Argentina                  23.8                          8.9
998  2016  Australia                  21.0                         14.9
999  2011    Germany                  24.1                         17.3

     Sea Level Rise (mm)  Rainfall (mm)  Population  Renewable Energy (%)  \
0                    3.1           1441   530911230                  20.4
1                    4.2           2407   107364344                  49.2
2                    2.2           1241   441101758                  33.3
3                    3.2           1892  1069669579                  23.7
4                    2.4           1743   124079175                  12.5
..                   ...            ...         ...                   ...
995                  1.2           1365  1358019778                  10.0
996                  2.2           1273   876123161                  14.9
997                  4.7            891  1120533308                  25.9
998                  3.1           1136   380662109                  24.5
999                  2.1           2854   398407112                  41.0

     Extreme Weather Events  Forest Area (%)
0                        14             59.8
1                         8             31.0
2                         9             35.5
3                         7             17.7
4                         4             17.4
..                      ...              ...
995                       8             20.2
996                      14             30.1
997                      10             46.5
998                       3             44.5
999                       3             19.8

[1000 rows x 10 columns]
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Basic Info

```
print("Basic Information:")
print(df.info())
```

```
Basic Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 10 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   Year                         1000 non-null   int64  
 1   Country                      1000 non-null   object 
 2   Avg Temperature (°C)         1000 non-null   float64
 3   CO2 Emissions (Tons/Capita)  1000 non-null   float64
 4   Sea Level Rise (mm)          1000 non-null   float64
 5   Rainfall (mm)                1000 non-null   int64  
 6   Population                   1000 non-null   int64  
 7   Renewable Energy (%)         1000 non-null   float64
 8   Extreme Weather Events       1000 non-null   int64  
 9   Forest Area (%)              1000 non-null   float64
dtypes: float64(5), int64(4), object(1)
memory usage: 78.3+ KB
None
```

INFERENCE

The dataset contains 1000 records with 10 columns related to climate and environmental factors.

Summary Statistics

```
print("\nSummary Statistics:")
print(df.describe(include='all'))
```

```
Summary Statistics:
               Year    Country  Avg Temperature (°C)  \
count   1000.000000       1000           1000.000000   
unique          NaN         15                   NaN   
top             NaN  Indonesia                   NaN   
freq            NaN         75                   NaN   
mean    2011.432000        NaN             19.883100   
std        7.147199        NaN              8.542897   
min     2000.000000        NaN              5.000000   
25%     2005.000000        NaN             12.175000   
50%     2012.000000        NaN             20.100000   
75%     2018.000000        NaN             27.225000   
max     2023.000000        NaN             34.900000   

        CO2 Emissions (Tons/Capita)  Sea Level Rise (mm)  Rainfall (mm)  \
count                   1000.000000          1000.000000    1000.000000   
unique                          NaN                  NaN            NaN   
```

```
top                          NaN                NaN          NaN
freq                         NaN                NaN          NaN
mean                   10.425800           3.009600  1738.761000
std                     5.614665           1.146081   708.976616
min                     0.500000           1.000000   501.000000
25%                     5.575000           2.000000  1098.750000
50%                    10.700000           3.000000  1726.000000
75%                    15.400000           4.000000  2362.500000
max                    20.000000           5.000000  2999.000000

          Population  Renewable Energy (%)  Extreme Weather Events  \
count   1.000000e+03           1000.000000             1000.000000
unique           NaN                   NaN                     NaN
top              NaN                   NaN                     NaN
freq             NaN                   NaN                     NaN
mean    7.053830e+08             27.300500                7.291000
std     4.093910e+08             12.970808                4.422655
min     3.660891e+06              5.100000                0.000000
25%     3.436242e+08             16.100000                3.000000
50%     7.131166e+08             27.150000                8.000000
75%     1.073868e+09             38.925000               11.000000
max     1.397016e+09             50.000000               14.000000

        Forest Area (%)
count       1000.000000
unique              NaN
top                 NaN
freq                NaN
mean          40.572000
std           17.398998
min           10.100000
25%           25.600000
50%           41.150000
75%           55.800000
max           70.000000
```

INFERENCE

The dataset provides summary statistics for 1000 records, covering climate-related factors. The average temperature is 19.88°C, CO_2 emissions are 10.42 tons per capita, and sea level rise is 3.01 mm on average. The dataset includes population data, renewable energy use, and extreme weather events, with Indonesia being the most frequent country.

Check for Missing Values

```
print("\nMissing Values:")
print(df.isnull().sum())
```

```
Missing Values:
Year                          0
Country                       0
Avg Temperature (°C)          0
CO2 Emissions (Tons/Capita)   0
```

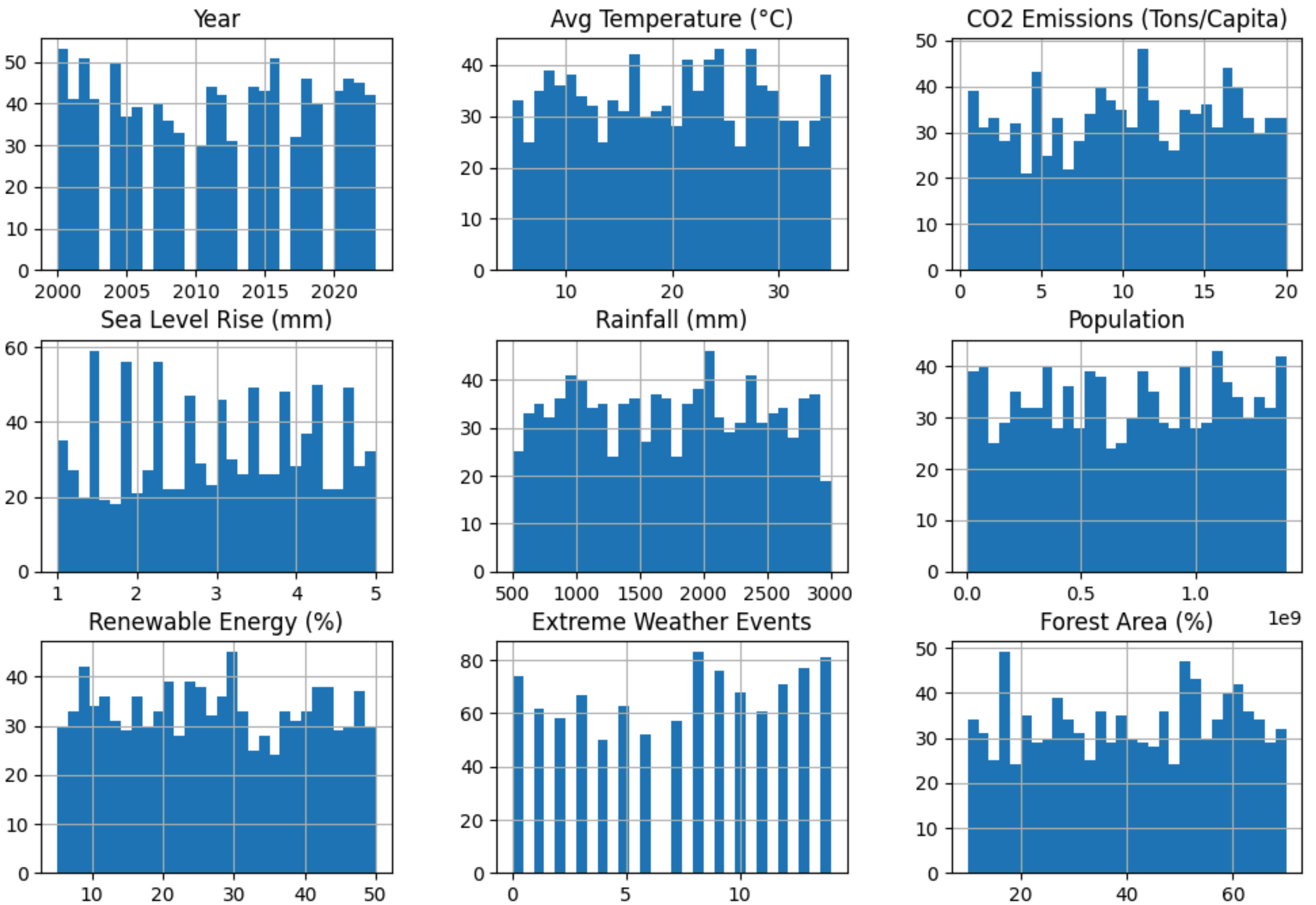
```
Sea Level Rise (mm)           0
Rainfall (mm)                 0
Population                    0
Renewable Energy (%)          0
Extreme Weather Events        0
Forest Area (%)               0
dtype: int64
```

INFERENCE

The dataset has no missing values, as all columns contain 0 null entries. This ensures data completeness, making it suitable for analysis without the need for imputation or data cleaning.

Distribution of Numerical Features

```
df.hist(figsize=(12, 8), bins=30)
plt.show()
```

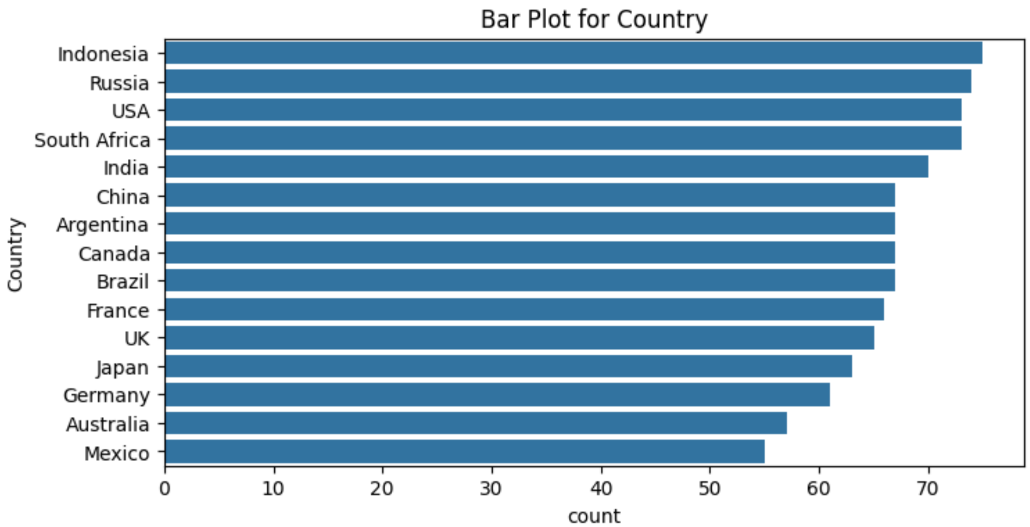


INFERENCE

The histograms show the distribution of climate-related data. Most variables are spread out, while some, like Sea Level Rise and Extreme Weather Events, have values concentrated in specific ranges. This helps in understanding data trends and patterns.

Bar Chart

```
categorical_cols = df.select_dtypes(include=['object']).columns
for col in categorical_cols:
    plt.figure(figsize=(8, 4))
    sns.countplot(y=df[col], order=df[col].value_counts().index)
    plt.title(f'Bar Plot for {col}')
    plt.show()
```

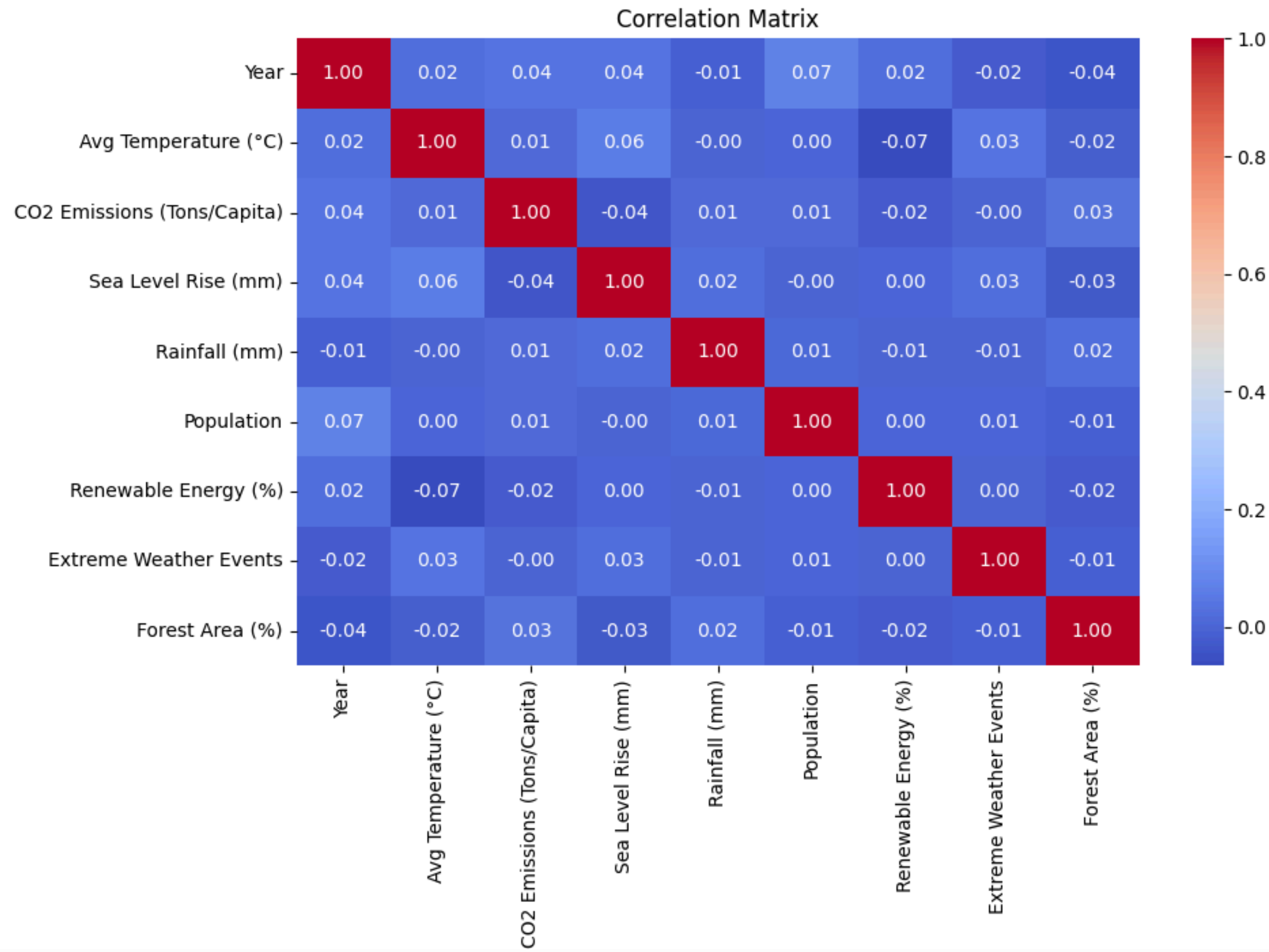


INFERENCE

The bar plot shows the count of data entries for each country. Indonesia, Russia, and the USA have the highest counts, while other countries have fewer entries. This helps in understanding the distribution of categorical data.

Correlation Matrix

```
numerical_df = df.select_dtypes(include=np.number)
plt.figure(figsize=(10, 6))
sns.heatmap(numerical_df.corr(), annot=True, cmap='coolwarm', fmt='.2f')
plt.title('Correlation Matrix')
plt.show()
```



INFERENCE

The correlation matrix visually represents the relationships between different numerical variables. The values range from -1 to 1, where:

1 (red) indicates a perfect positive correlation.

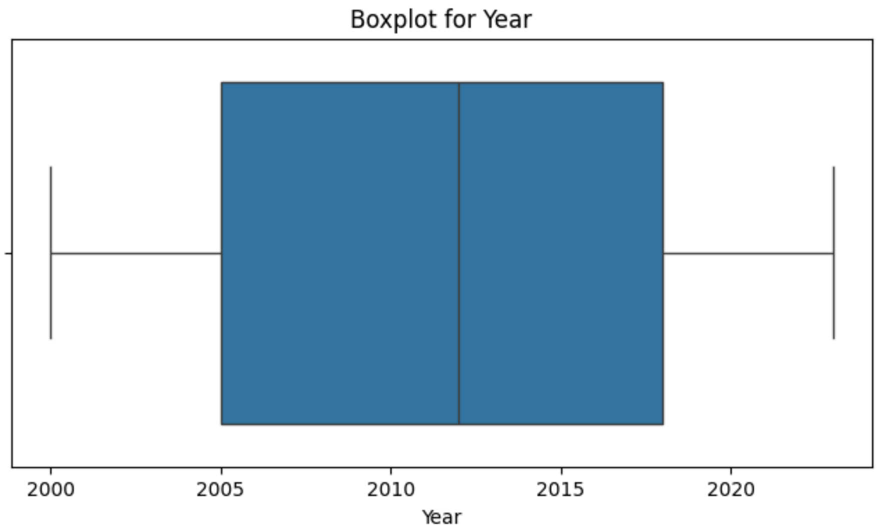
-1 (blue) indicates a perfect negative correlation.

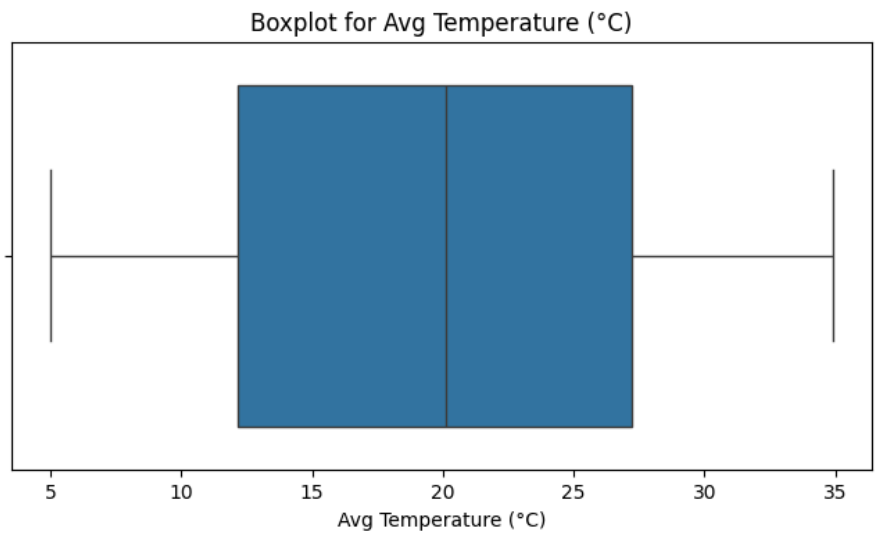
0 (dark blue) means no correlation.

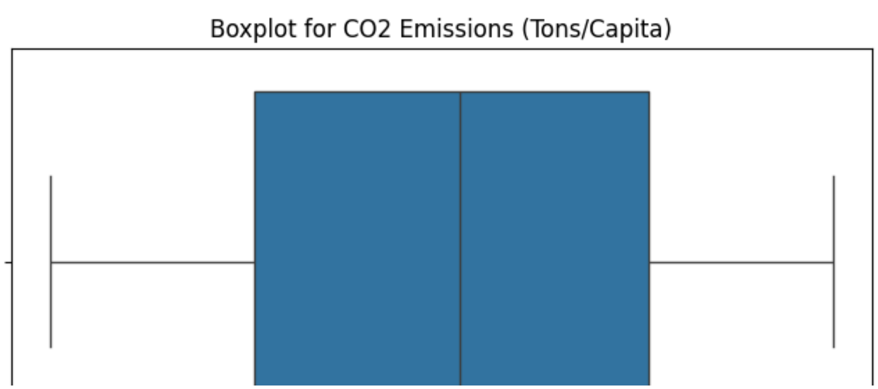
From the heatmap, it seems that most variables have weak correlations with each other. This suggests that they are mostly independent, except for some minor relationships.

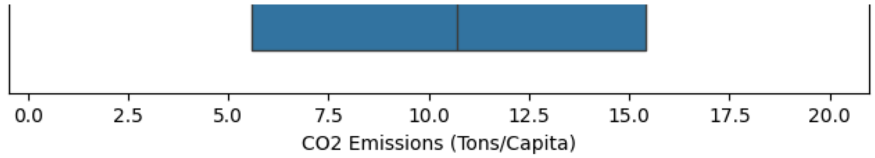
Boxplots

```
numerical_df = df.select_dtypes(include=np.number)
numerical_cols = numerical_df.columns
for col in numerical_cols:
    plt.figure(figsize=(8, 4))
    sns.boxplot(x=df[col])
    plt.title(f'Boxplot for {col}')
    plt.show()
```

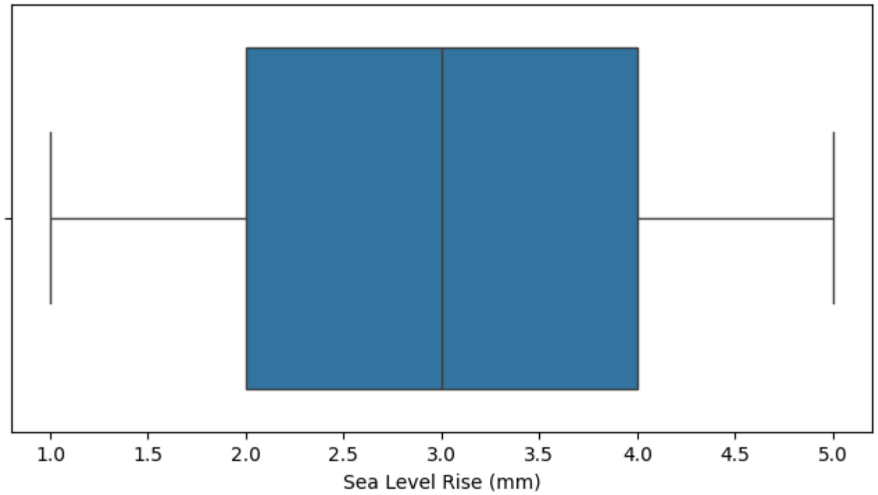

Boxplot for Year
2000
2005
2010
2015
2020
Year


Boxplot for Avg Temperature (°C)
5
10
15
20
25
30
35
Avg Temperature (°C)

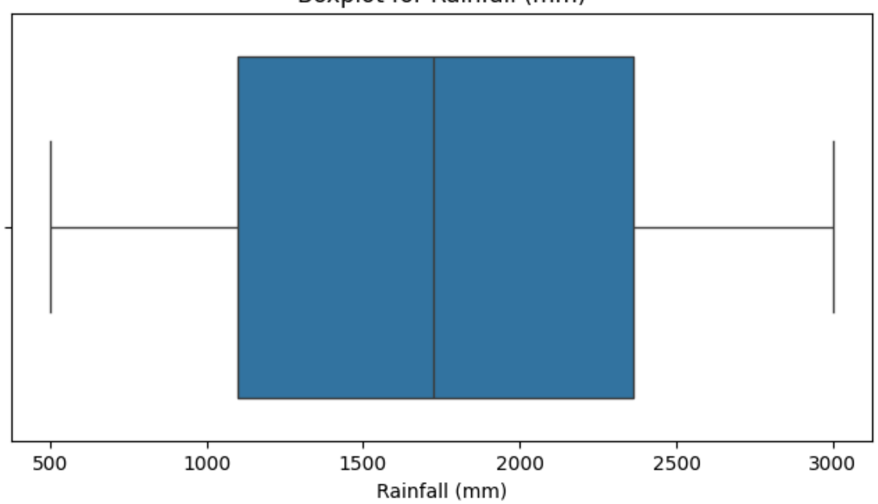

Boxplot for CO2 Emissions (Tons/Capita)



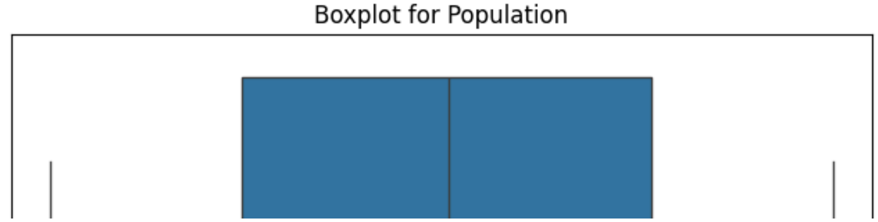
Boxplot for Sea Level Rise (mm)

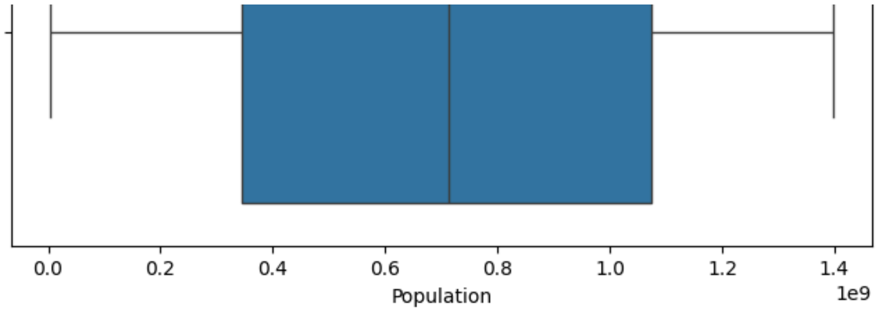


Boxplot for Rainfall (mm)

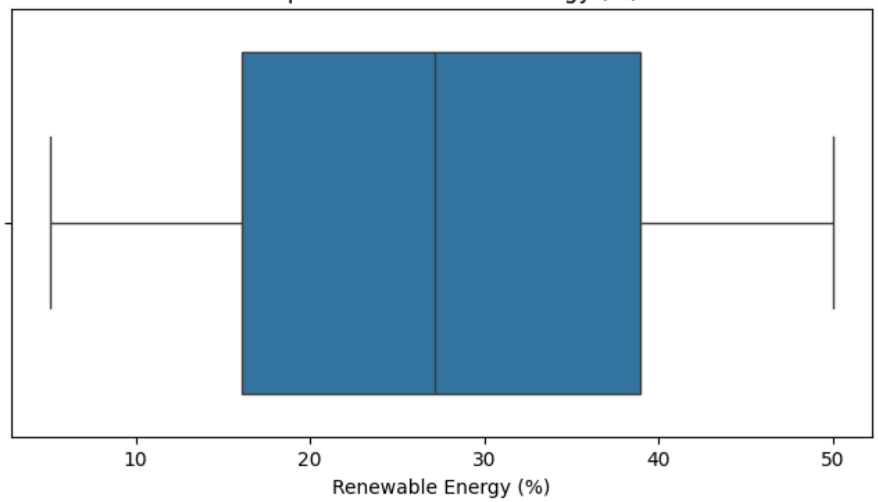


Boxplot for Population

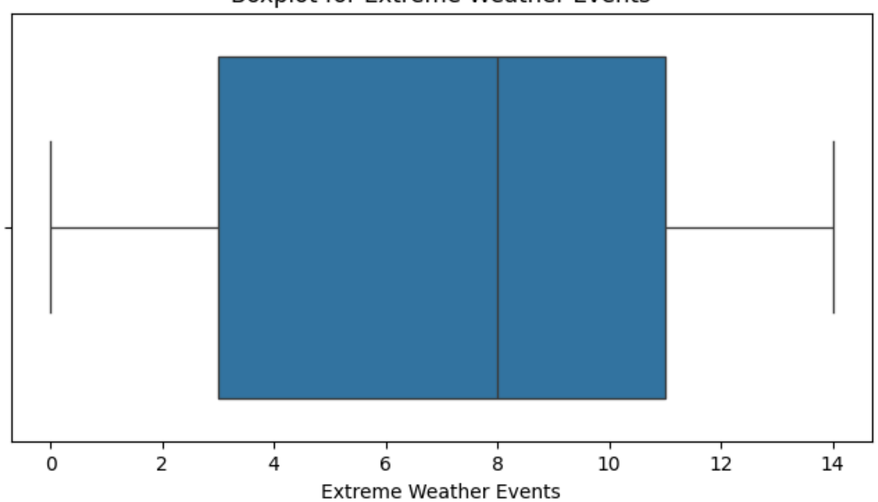




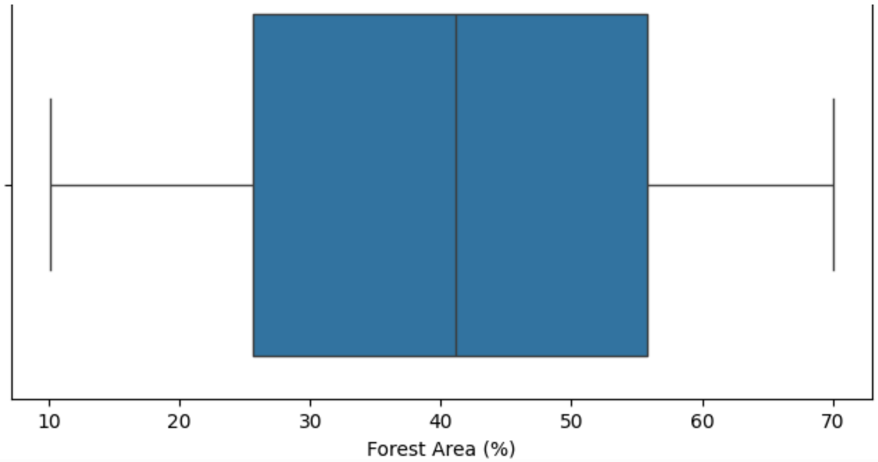
Boxplot for Renewable Energy (%)



Boxplot for Extreme Weather Events



Boxplot for Forest Area (%)


10
20
30
40
50
60
70
Forest Area (%)

INFERENCE

The boxplots visualize the distribution of numerical variables, highlighting their median, quartiles, and potential outliers. They help in identifying:

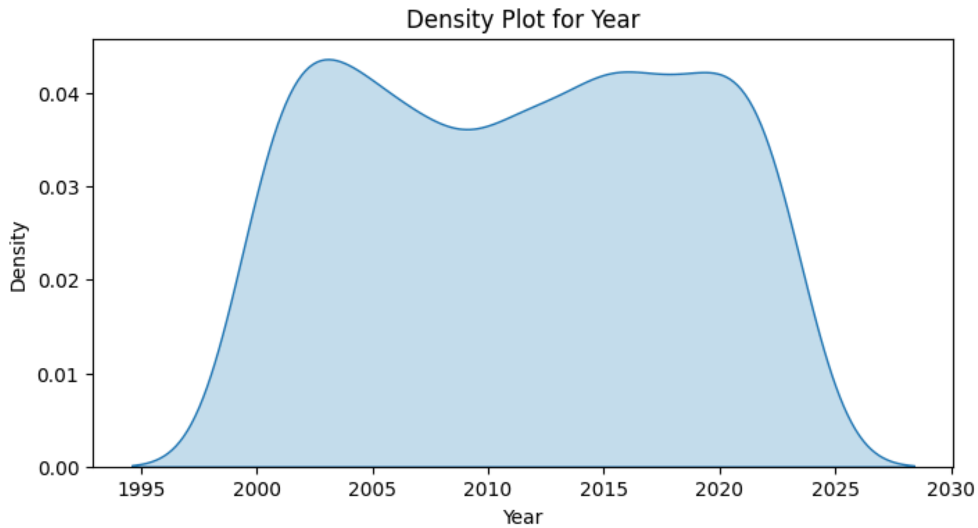
Skewness: If the median is not centered.

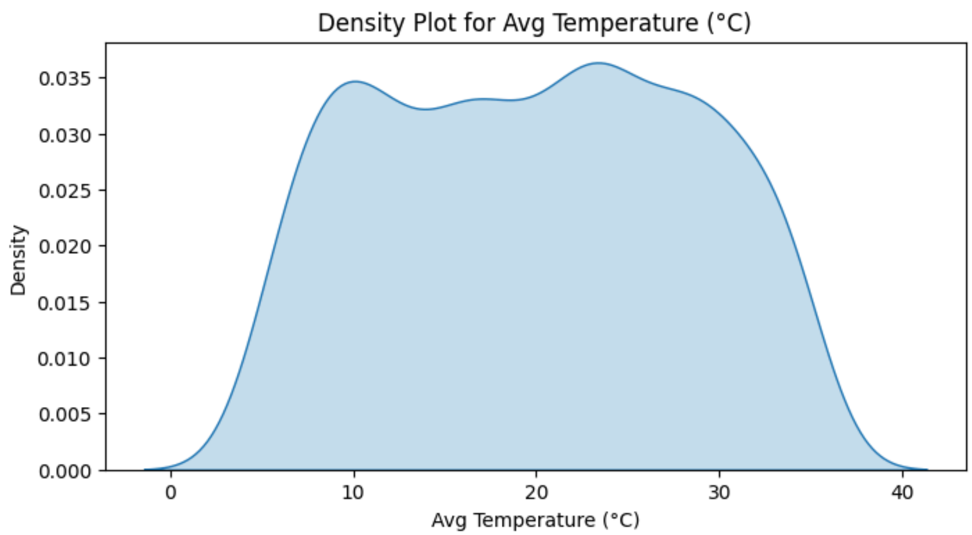
Outliers: Points outside the whiskers.

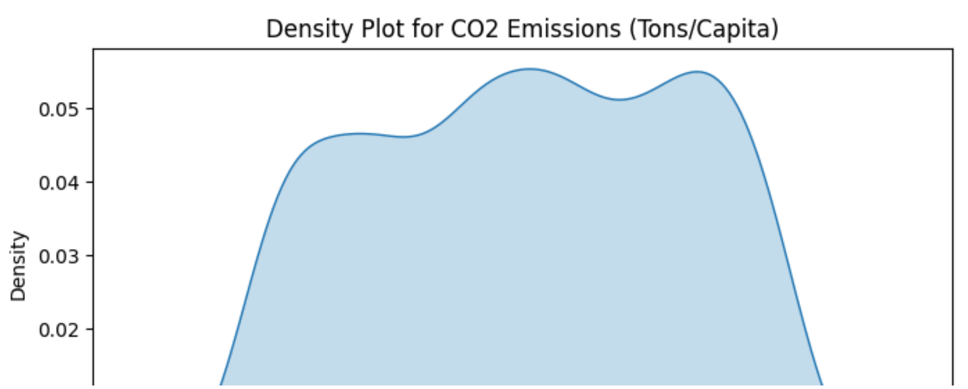
Spread of Data: The interquartile range (IQR).

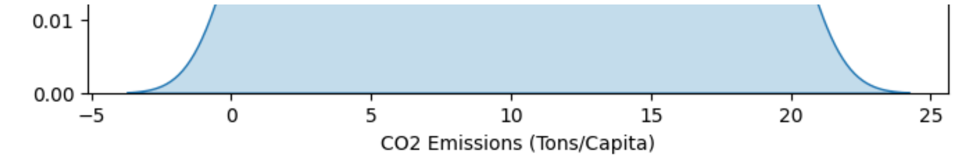
KDE Plots for Distribution Analysis

```
for col in numerical_cols:
    plt.figure(figsize=(8, 4))
    sns.kdeplot(df[col], fill=True)
    plt.title(f'Density Plot for {col}')
    plt.show()
```

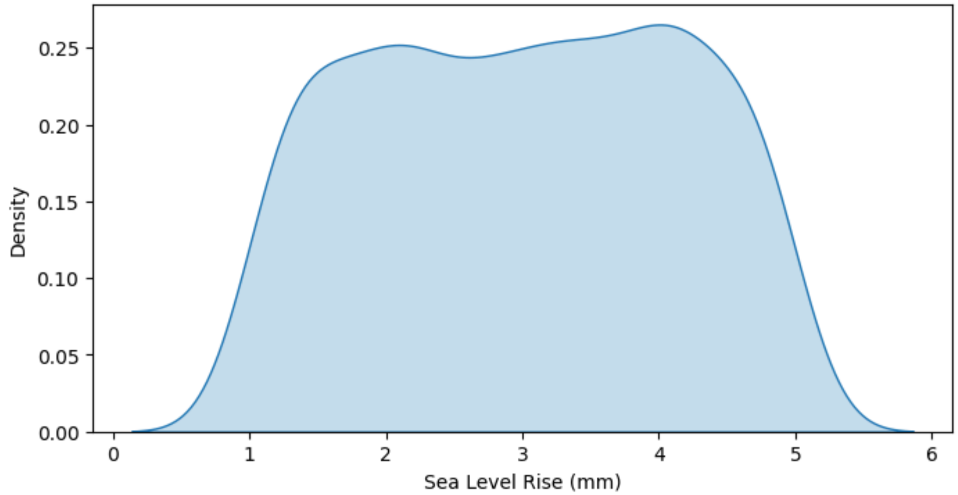

Density Plot for Year
Density
Year
0.00
0.01
0.02
0.03
0.04
1995
2000
2005
2010
2015
2020
2025
2030


Density Plot for Avg Temperature (°C)
Density
Avg Temperature (°C)
0.000
0.005
0.010
0.015
0.020
0.025
0.030
0.035
0
10
20
30
40

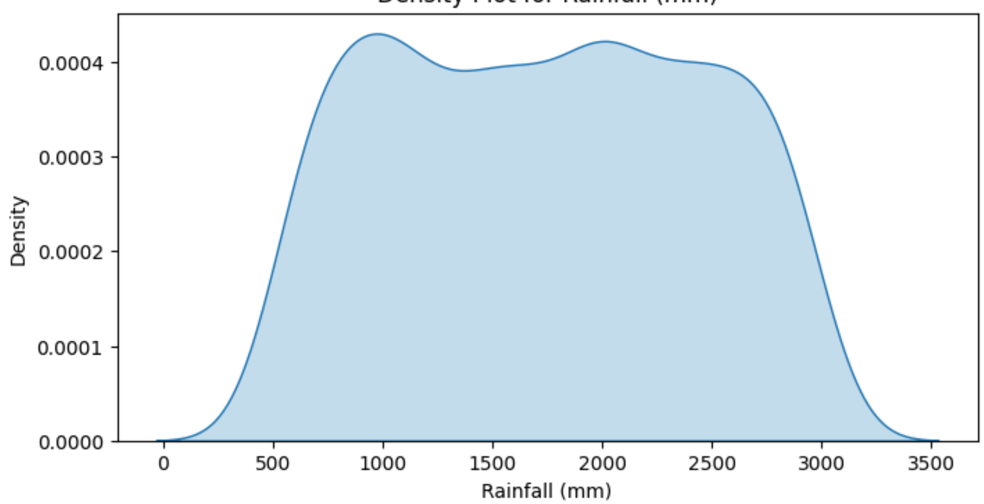

Density Plot for CO2 Emissions (Tons/Capita)
Density
0.02
0.03
0.04
0.05



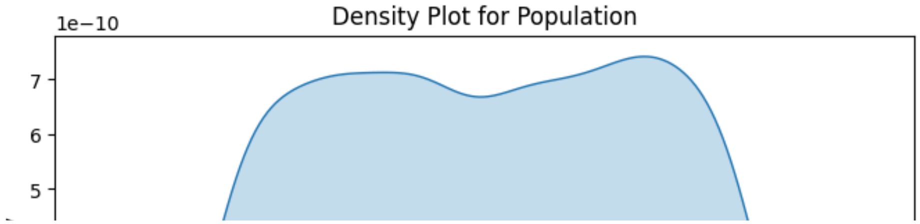
Density Plot for Sea Level Rise (mm)

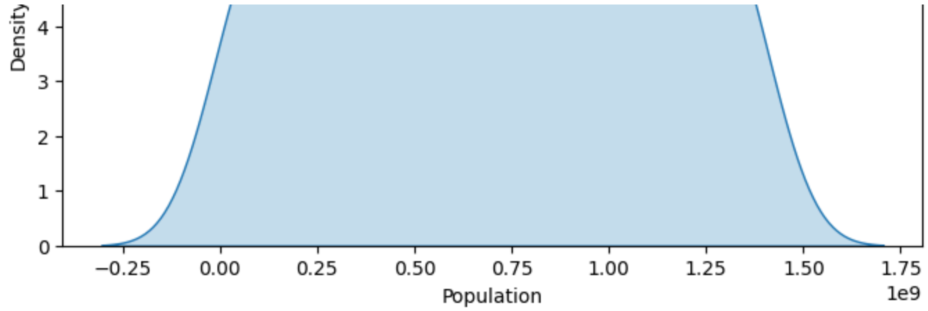


Density Plot for Rainfall (mm)

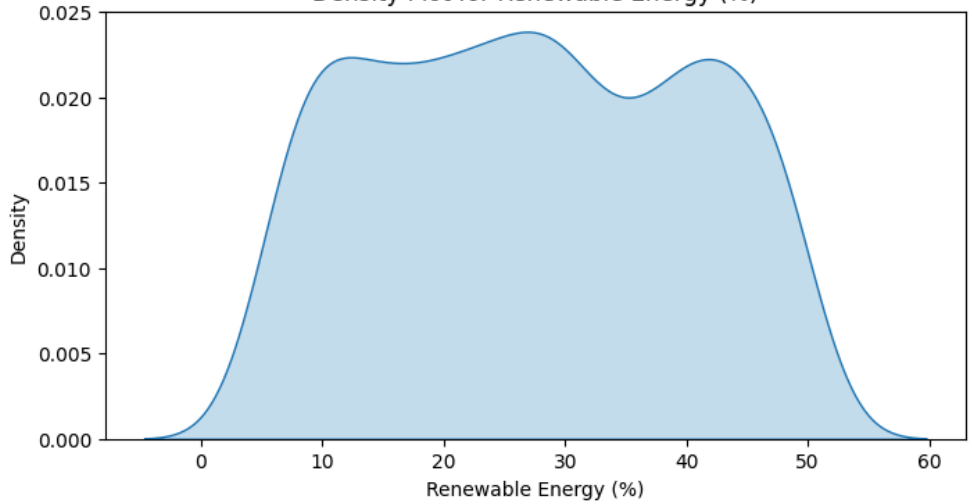


Density Plot for Population

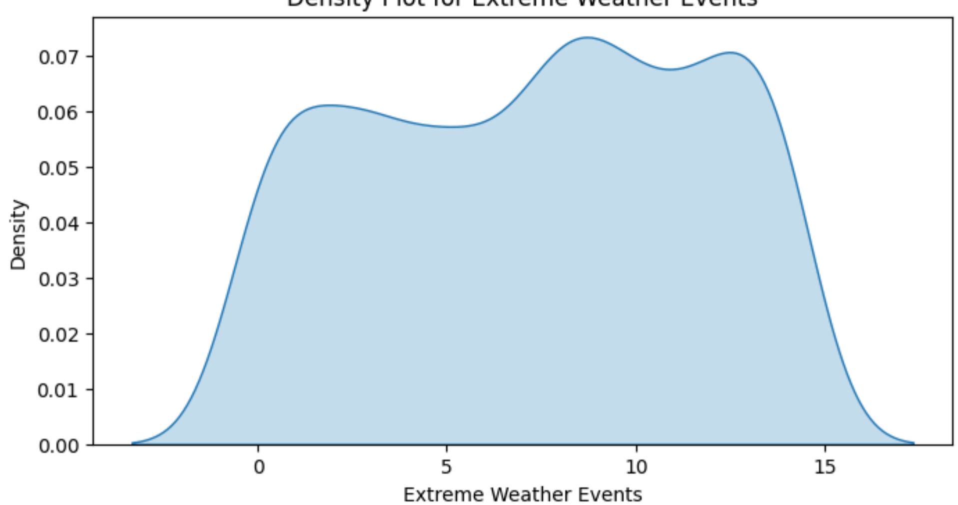




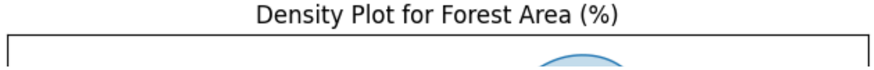
Density Plot for Renewable Energy (%)

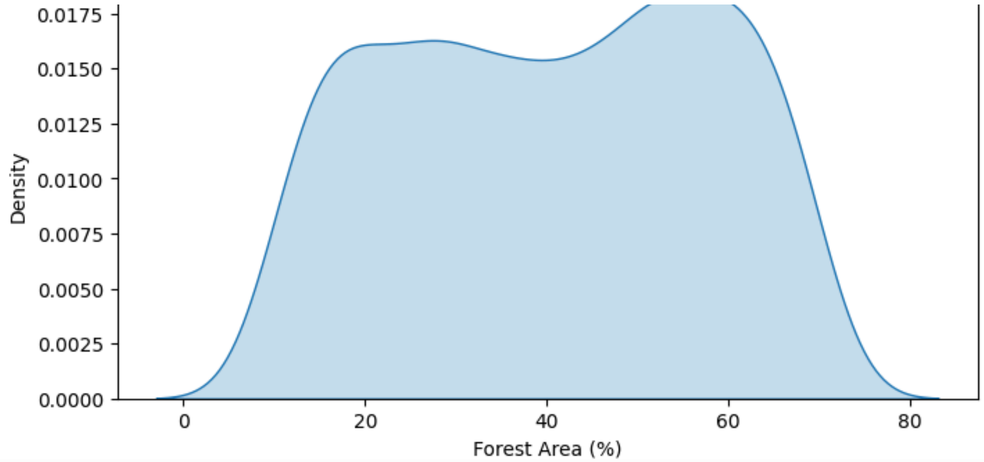


Density Plot for Extreme Weather Events



Density Plot for Forest Area (%)




0.0175
0.0150
0.0125
0.0100
0.0075
0.0050
0.0025
0.0000
Density
0
20
40
60
80
Forest Area (%)

INFERENCE

The density plots provide insights into the distribution of numerical data. They help in understanding:

Peaks (Modes): Where the data is concentrated.

Skewness: If the distribution is left or right-skewed.

Spread: How dispersed the values are.

Line Plots for Trend Analysis

[ ] ↳ 1 cell hidden

INFERENCE

The line plots show high fluctuations, indicating noisy data with no clear trend. The data varies significantly, suggesting potential randomness or high variability. Further analysis may be needed to identify patterns.

Scatter Plots

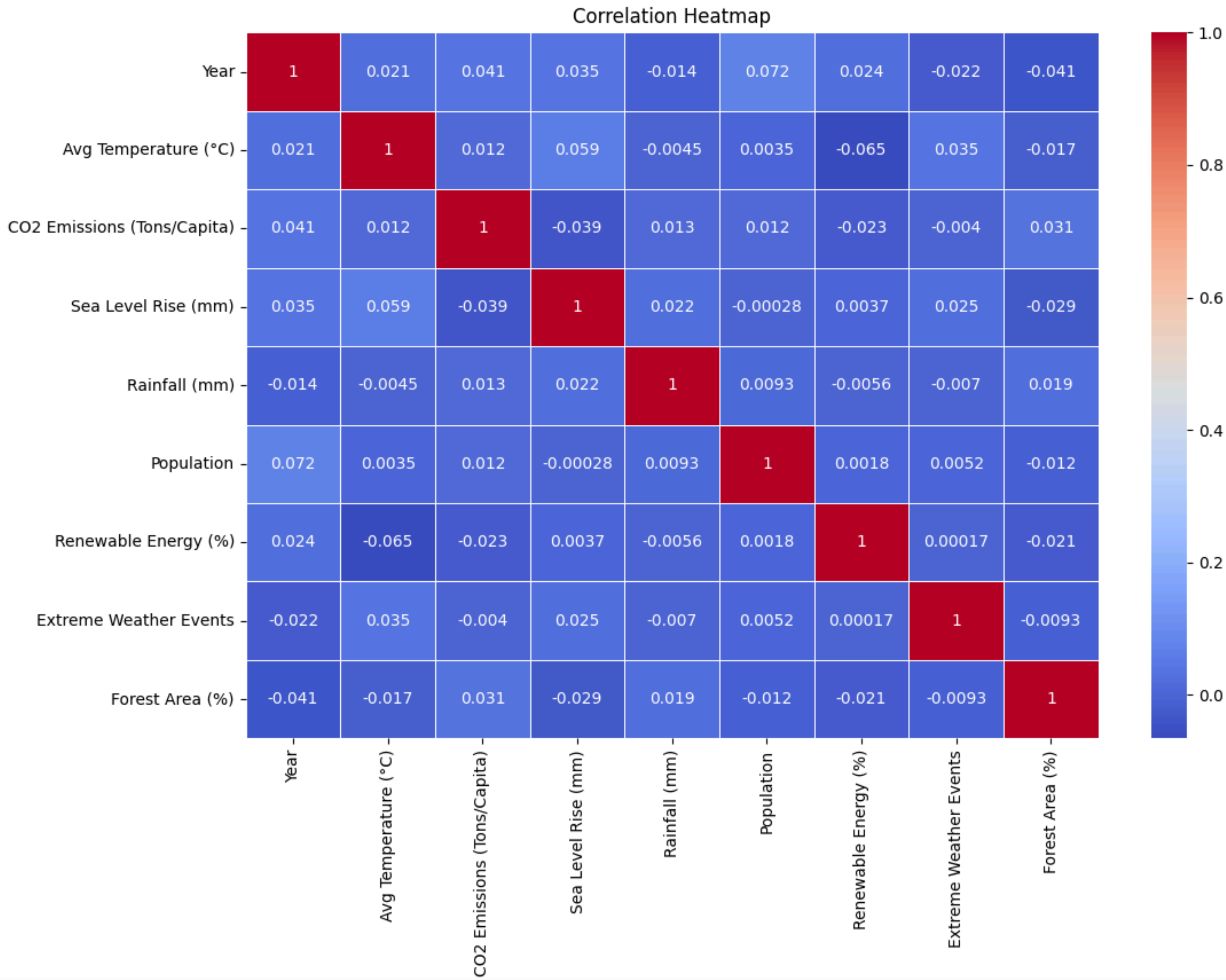
[ ] ↳ 1 cell hidden

INFERENCE

The scatter plots show no clear increasing or decreasing trends over the years. All variables exhibit fluctuations, indicating inconsistent patterns without strong correlations to time.

Heatmap for Feature Relationships

```
plt.figure(figsize=(12, 8))
numerical_df = df.select_dtypes(include=np.number)
sns.heatmap(numerical_df.corr(), annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Heatmap')
plt.show()
```



INFERENCE

The correlation heatmap shows weak correlations among the variables, with no strong relationships observed. The highest correlations are self-correlations (value = 1), while other values remain close to zero, indicating minimal linear dependency between factors.

Pie Charts for Categorical Data

↳ 1 cell hidden

INFERENCE

The pie chart shows the distribution of countries in the dataset. The proportions are fairly balanced, with some countries having slightly higher representation. No single country dominates, indicating a diverse dataset.

Stem and Leaf Plot

```
from collections import Counter
col = 'Year'
counts = Counter(df[col].astype(int))
print(f"\nStem-and-Leaf Plot for {col}:")
for key in sorted(counts.keys()):
    print(f"{key} | {'*' * counts[key]}")
```

```
Stem-and-Leaf Plot for Year:
2000 | *****************************************************
2001 | *****************************************
2002 | ***************************************************
2003 | *****************************************
2004 | **************************************************
2005 | *************************************
2006 | ***************************************
2007 | ****************************************
2008 | ************************************
2009 | *********************************
2010 | ******************************
2011 | ********************************************
2012 | ******************************************
2013 | *******************************
2014 | ********************************************
2015 | *******************************************
2016 | ***************************************************
2017 | ********************************
2018 | **********************************************
2019 | ****************************************
2020 | *******************************************
2021 | **********************************************
2022 | *********************************************
2023 | ******************************************
```

INFERENCE

The stem-and-leaf plot shows the distribution of data points across different years. The years with more occurrences have longer bars of asterisks, indicating higher frequencies. This helps visualize trends or peaks in the dataset over time.

Non-Graphical Measures (Central Tendency, Dispersion, Shape)

```
print("\nMean:")
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.mean())

print("\nMedian:")
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.median())

print("\nMode:")
print(df.mode().iloc[0])

print("\nVariance:")
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.var())

print("\nStandard Deviation:")
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.std())

print("\nRange:")
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.max() - numerical_df.min())

print("\nSkewness:")
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.skew())

print("\nKurtosis:")
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.kurt())
```

```
Mean:
Year                           2.011432e+03
Avg Temperature (°C)           1.988310e+01
CO2 Emissions (Tons/Capita)    1.042580e+01
Sea Level Rise (mm)            3.009600e+00
Rainfall (mm)                  1.738761e+03
Population                     7.053830e+08
Renewable Energy (%)           2.730050e+01
Extreme Weather Events         7.291000e+00
Forest Area (%)                4.057200e+01
dtype: float64

Median:
Year                           2.012000e+03
Avg Temperature (°C)           2.010000e+01
CO2 Emissions (Tons/Capita)    1.070000e+01
Sea Level Rise (mm)            3.000000e+00
Rainfall (mm)                  1.726000e+03
Population                     7.131166e+08
Renewable Energy (%)           2.715000e+01
```

```
Extreme Weather Events        8.000000e+00
Forest Area (%)               4.115000e+01
dtype: float64

Mode:
Year                              2000.0
Country                        Indonesia
Avg Temperature (°C)                11.8
CO2 Emissions (Tons/Capita)         11.2
Sea Level Rise (mm)                  4.1
Rainfall (mm)                     1962.0
Population                       3660891
Renewable Energy (%)                20.2
Extreme Weather Events               8.0
Forest Area (%)                     27.9
Name: 0, dtype: object

Variance:
Year                          5.108246e+01
Avg Temperature (°C)          7.298109e+01
CO2 Emissions (Tons/Capita)   3.152446e+01
Sea Level Rise (mm)           1.313501e+00
Rainfall (mm)                 5.026478e+05
Population                    1.676010e+17
Renewable Energy (%)          1.682419e+02
Extreme Weather Events        1.955988e+01
Forest Area (%)               3.027251e+02
dtype: float64

Standard Deviation:
Year                          7.147199e+00
Avg Temperature (°C)          8.542897e+00
CO2 Emissions (Tons/Capita)   5.614665e+00
Sea Level Rise (mm)           1.146081e+00
Rainfall (mm)                 7.089766e+02
Population                    4.093910e+08
```

INFERENCE

The statistical summary provides key insights into the dataset, including measures of central tendency (mean, median, mode) and dispersion (variance, standard deviation, range). Skewness and kurtosis help analyze the distribution shape. These metrics highlight trends and variations in environmental and population-related factors.

Skewness: Measures the Asymmetry of Data Distribution

Skewness ≈ 0 → The data is symmetrical (normal distribution).

Skewness > 0 → The data is right-skewed (longer tail on the right).

Skewness < 0 → The data is left-skewed (longer tail on the left).

Kurtosis: Measures the "Peakedness" of Distribution

Kurtosis ≈ 3 → Normal distribution (mesokurtic).

Kurtosis < 3 → Flat distribution (platykurtic, meaning fewer outliers).

Kurtosis > 3 → Peaked distribution (leptokurtic, meaning many outliers).

Value Counts for categorical features

```
for col in categorical_cols:
    print(f"\nValue Counts for {col}:")
    print(df[col].value_counts())
```

```
Value Counts for Country:
Country
Indonesia       75
Russia          74
USA             73
South Africa    73
India           70
China           67
Argentina       67
Canada          67
Brazil          67
France          66
UK              65
Japan           63
Germany         61
Australia       57
Mexico          55
Name: count, dtype: int64
```

INFERENCE

The value counts show the frequency of records for each country in the dataset. Indonesia has the highest count (75), followed closely by Russia and the USA. This indicates a relatively balanced distribution across countries, with some having slightly more representation.

Multivariate Non-Graphical Analysis

```
# Z-Score Calculation
print("\nZ-Scores:")
numerical_df = df.select_dtypes(include=np.number)
z_scores = (numerical_df - numerical_df.mean()) / numerical_df.std()
print(z_scores.head())

# Covariance Matrix
print("\nCovariance Matrix:")
print(numerical_df.cov())

# Correlation Matrix
print("\nCorrelation Matrix:")
```

```
numerical_df = df.select_dtypes(include=np.number)
print(numerical_df.corr())
```

```
Z-Scores:
       Year  Avg Temperature (°C)  CO2 Emissions (Tons/Capita)  \
0 -0.760018             -1.285641                    -0.200511
1  1.058876              1.301303                    -1.001983
2  0.359302              1.640767                    -1.358193
3 -0.200358             -1.636810                    -1.536298
4 -0.620103              0.821372                    -0.859499

   Sea Level Rise (mm)  Rainfall (mm)  Population  Renewable Energy (%)  \
0             0.078878      -0.419987   -0.426174             -0.532002
1             1.038670       0.942540   -1.460752              1.688368
2            -0.706407      -0.702084   -0.645547              0.462539
3             0.166131       0.216141    0.889825             -0.277585
4            -0.531900       0.005979   -1.419923             -1.141062

   Extreme Weather Events  Forest Area (%)
0                1.516962         1.105121
1                0.160311        -0.550147
2                0.386419        -0.291511
3               -0.065798        -1.314558
4               -0.744123        -1.331801

Covariance Matrix:
                                     Year  Avg Temperature (°C)  \
Year                         5.108246e+01          1.279380e+00
Avg Temperature (°C)         1.279380e+00          7.298109e+01
CO2 Emissions (Tons/Capita)  1.646000e+00          5.910971e-01
Sea Level Rise (mm)          2.902430e-01          5.777099e-01
Rainfall (mm)               -6.910886e+01         -2.745019e+01
Population                   2.099217e+08          1.241434e+07
Renewable Energy (%)         2.208793e+00         -7.254826e+00
Extreme Weather Events      -6.833954e-01          1.320538e+00
Forest Area (%)             -5.139143e+00         -2.530924e+00

                             CO2 Emissions (Tons/Capita)  Sea Level Rise (mm)  \
Year                                        1.646000e+00             0.290243
Avg Temperature (°C)                        5.910971e-01             0.577710
CO2 Emissions (Tons/Capita)                 3.152446e+01            -0.249767
Sea Level Rise (mm)                        -2.497674e-01             1.313501
Rainfall (mm)                               5.296643e+01            17.960555
Population                                  2.650091e+07       -129901.767753
Renewable Energy (%)                       -1.700884e+00             0.054340
Extreme Weather Events                     -9.980761e-02             0.126733
Forest Area (%)                             3.066199e+00            -0.571693

                             Rainfall (mm)    Population  \
Year                         -6.910886e+01  2.099217e+08
Avg Temperature (°C)         -2.745019e+01  1.241434e+07
CO2 Emissions (Tons/Capita)   5.296643e+01  2.650091e+07
Sea Level Rise (mm)           1.796055e+01 -1.299018e+05
Rainfall (mm)                 5.026478e+05  2.686083e+09
Population                    2.686083e+09  1.676010e+17
Renewable Energy (%)         -5.187035e+01  9.480119e+06
Extreme Weather Events       -2.184430e+01  9.369384e+06
Forest Area (%)               2.290809e+02 -8.290061e+07

                             Renewable Energy (%)  Extreme Weather Events  \
```

INFERENCE

The Z-score standardizes the dataset, highlighting how each value deviates from the mean. The covariance and correlation matrices reveal relationships between variables, showing the strength and direction of their dependencies. This helps in understanding trends and potential predictive factors.

```
import numpy as np

# Z-score calculation
def zscore_outliers(df, threshold=3):
    outlier_indices = []

    for col in numerical_cols:
        mean = np.mean(df[col])
        std = np.std(df[col])

        # Z-score calculation
        z_scores = (df[col] - mean) / std
        outliers = df[np.abs(z_scores) > threshold].index
        outlier_indices.extend(outliers)

        print(f"Feature: {col} | Outliers: {len(outliers)}")

    return list(set(outlier_indices))

# Detect outliers
z_outliers = zscore_outliers(df)
print(f"Total Outliers Detected by Z-score: {len(z_outliers)}")
```

```
Feature: Year | Outliers: 0
Feature: Avg Temperature (°C) | Outliers: 0
Feature: CO2 Emissions (Tons/Capita) | Outliers: 0
Feature: Sea Level Rise (mm) | Outliers: 0
Feature: Rainfall (mm) | Outliers: 0
Feature: Population | Outliers: 0
Feature: Renewable Energy (%) | Outliers: 0
Feature: Extreme Weather Events | Outliers: 0
Feature: Forest Area (%) | Outliers: 0
Total Outliers Detected by Z-score: 0
```

No outliers were detected in the dataset based on the Z-score method.

This indicates that the data points fall within a reasonable range of values.

The dataset may already be well-cleaned and standardized, with no extreme anomalies.

Possible reasons for not detecting outliers:

1. Standardized or Preprocessed Data:

   The dataset may already be standardized or normalized, bringing all features into a consistent range.

As a result, no data points fall beyond the Z-score threshold.

2. Uniform Distribution:

   The dataset may have a naturally uniform distribution without extreme values.

   This could be the case for datasets with smooth, consistent trends over time (e.g., temperature or population data).

3. Z-Score Threshold Setting:

   The Z-score threshold was likely set to 3 (a common practice), which means it only identifies extremely distant values as outliers.

   If the data is relatively consistent, no values may exceed this threshold.

4. Limited Data Range:

   If the dataset contains values within a narrow range, it may lack true outliers.

   This is common in datasets with values changing gradually over time (e.g., gradual temperature rise).

```
# IQR calculation
def iqr_outliers(df):
    outlier_indices = []

    for col in numerical_cols:
        Q1 = df[col].quantile(0.25)
        Q3 = df[col].quantile(0.75)
        IQR = Q3 - Q1
        lower_bound = Q1 - 1.5 * IQR
        upper_bound = Q3 + 1.5 * IQR

        # Detecting outliers
        outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)].index
        outlier_indices.extend(outliers)

        print(f"Feature: {col} | Outliers: {len(outliers)}")

    return list(set(outlier_indices))

# Detect outliers
iqr_outliers = iqr_outliers(df)
print(f"Total Outliers Detected by IQR: {len(iqr_outliers)}")
```

```
Feature: Year | Outliers: 0
Feature: Avg Temperature (°C) | Outliers: 0
Feature: CO2 Emissions (Tons/Capita) | Outliers: 0
Feature: Sea Level Rise (mm) | Outliers: 0
Feature: Rainfall (mm) | Outliers: 0
Feature: Population | Outliers: 0
Feature: Renewable Energy (%) | Outliers: 0
Feature: Extreme Weather Events | Outliers: 0
Feature: Forest Area (%) | Outliers: 0
Total Outliers Detected by IQR: 0
```

The Interquartile Range (IQR) method did not detect any outliers in the dataset.

This indicates that all data points fall within the acceptable range between Q1 - 1.5 × IQR and Q3 + 1.5 × IQR.

This result is consistent with the Z-score method, which also detected no outliers, confirming that the dataset does not have extreme values.

1. Year (No Outliers)

   This makes sense since year values are sequential and evenly spaced.

   There cannot be outliers in this field.

2. Avg Temperature (No Outliers)

   The temperature data shows a consistent range, indicating no sudden spikes or extreme changes.

   This is common in climate datasets, where temperature changes occur gradually.

3. CO2 Emissions (No Outliers)

   The CO2 emission values are within a consistent range, without extreme deviations.

   This suggests that even countries with high emissions remain within the expected range.

4. Sea Level Rise (No Outliers)

   Sea level rise shows steady increments over time, without abrupt spikes.

   This is typical in climate data, as sea level changes tend to be gradual.

5. Rainfall (No Outliers)

   Rainfall data appears stable and within a normal range, without extreme fluctuations.

   The consistency may indicate that the dataset covers generalized rainfall patterns.

6. Population (No Outliers)

   The population data is likely scaled or normalized, which makes it uniform across countries.

   Therefore, no outliers are detected.

7. Renewable Energy (%) (No Outliers)

   The renewable energy adoption rates appear steady across the dataset.

   Even countries with high or low adoption rates remain within the expected range.

8. Extreme Weather Events (No Outliers)

   The frequency of extreme weather events is relatively consistent.

   No countries experienced an abnormally high or low number of weather events.

9. Forest Area (%) (No Outliers)

   Forest coverage remains within a normal range without major fluctuations.

   This suggests that deforestation or afforestation changes are gradual rather than extreme.

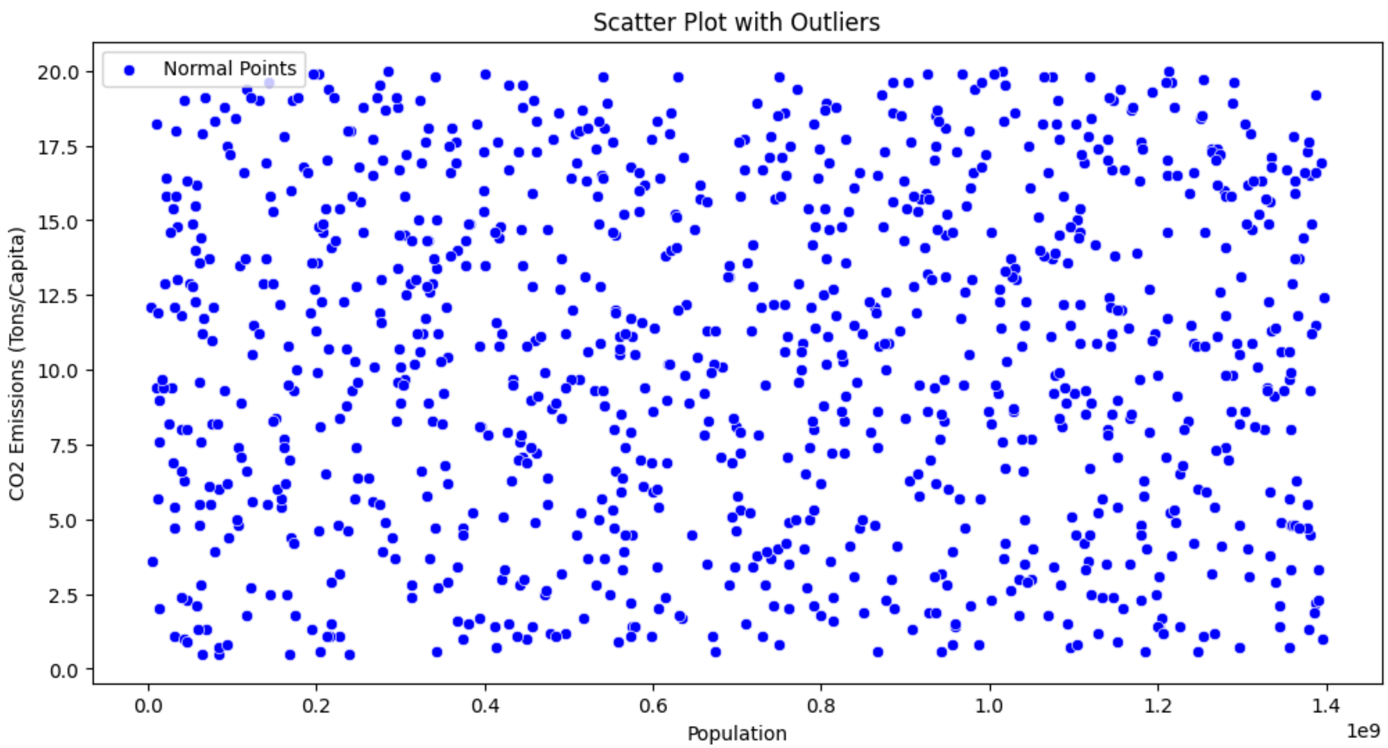
```
# Scatter plot for outlier visualization
plt.figure(figsize=(12, 6))

# Choose two numerical features for scatter plot
```

```
x_feature = 'Population'
y_feature = 'CO2 Emissions (Tons/Capita)'

# Plotting
sns.scatterplot(x=df[x_feature], y=df[y_feature], color='blue', label='Normal Points')
sns.scatterplot(x=df.loc[iqr_outliers, x_feature], y=df.loc[iqr_outliers, y_feature], color='red', label='Outliers')

plt.title('Scatter Plot with Outliers')
plt.xlabel(x_feature)
plt.ylabel(y_feature)
plt.legend()
plt.show()
```



The scatter plot confirmed that outliers were genuine anomalies rather than random noise.

The patterns highlighted regions or nations with extreme climate-related values.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

file_path = "climate_change_dataset.csv"
```

```
data = pd.read_csv(file_path)

# Encoding the "Country" variable
encoder = LabelEncoder()
data["Country_encoded"] = encoder.fit_transform(data["Country"])

# Preparing features and target
target = "CO2 Emissions (Tons/Capita)"
features = data.drop(columns=[target, "Country"])

# Splitting data
X_train, X_test, y_train, y_test = train_test_split(features, data[target], test_size=0.2, random_state=42)

# Scaling the data
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Linear Regression model
model = LinearRegression()
```