

✓ Climate Change Exploratory Data Analysis (EDA)

1. Information

Project Title: Impact of CO2 Emissions on Global Temperature Anomalies


By: Ajay Sharma

✓ 2. Table of Contents

1. Preliminaries
2. Data Description
3. Exploratory Data Analysis (EDA)
4. Building a Regression Model
5. Machine Learning Techniques (SGD) & Exploration
6. Mathematical Statistics
7. Insights & Conclusion
8. Reference(s)

Remark: We use the convention 2.N.M, where 2 refers to the table of contents, N refers to the sub-section in the table of contents {1,..., 9}, and M refers to the numbered sub-heading (as many as needed).

```
1 from google.colab import drive
2 drive.mount('/content/drive')
```

 Mounted at /content/drive

✓ 2.1: Preliminaries

2.1.1: Project Overview

The purpose of this project is to gather, clean, and analyze data with respect to CO2 emissions over time and its effect on temperature anomalies. We will describe how various sectors contribute to climate change, the evolution of said relationship across various geographical areas, and give recommendations to rectify the issues outlined above.

2.1.2: Background

Climate change is the most important issue we must address in the 21st century, driven primarily by the increase in greenhouse gases such as CO2 in the atmosphere. As CO2 levels rise due to human activities e.g. burning of fossil fuels, and deforestation, it has been well documented that temperatures have increased significantly over the past few decades leading to more extreme weather phenomenon: temperatures tending to the extremes (very hot & cold), rising sea levels, climate patterns shifting (seasonal), disruption to natural ecosystems (land & sea), and other factors (which we might not even have considered). Understanding the relationship between CO2 emissions and global temperature anomalies is crucial for predicting future climate impacts, allowing government officials to make better policy decisions aimed at reducing emissions. By quantifying this relationship, we can better anticipate the consequences of continued emissions and (ideally) mitigate the effects of climate change.

✓ 2.2: Data Description

2.2.1: Data Sources

- The source of my data set comes from the following: <https://github.com/owid/co2-data?tab=readme-ov-file>

This owid-co2-data set comes from: Energy Institute (EI), U.S. Energy Information Administration (EIA), Global Carbon Project, Jones et al. (2024), Climate Watch, and other scientific publications. To say the least, this is very reliable data to perform predictions & statistical analysis.


2.2.2: Data Collection

We will first ensure the data set is properly cleaned. In particular, transform the data set to handle any missing values to ensure the data types are appropriate for analysis. With our goal in mind, we will focus on key metrics: Total CO2 emissions, CO2 per capita, and temperature changes attributed to CO2.

✓ 2.2.3: Data Cleaning

As outlined above, we are interested in a certain subset of the data (metrics) relevant to our analysis. To make this more accessible, I have removed all rows with missing values in critical fields. Furthermore, converting the year to a date-time format allows us to standarize the data to perform time-series analysis & aggregate the data by year to allow for a global view: summarizing CO2 emissions and averaging temperature changes to capture trends over time. Below is the relevant code:

```
1 import pandas as pd
2
3 # load data set
4 owid_co2_df = pd.read_csv('owid-co2-data.csv')
5
6 # display a couple of rows to gain an understanding of the data set
7 owid_co2_head = owid_co2_df.head()
8 owid_co2_summary = owid_co2_df.describe(include='all')
9
10 owid_co2_head, owid_co2_summary
11
12 # data cleaning:
13 # key metrics: 'year', 'country', 'co2', 'temperature_change_from_co2'
14 relevant_columns = ['year', 'country', 'co2', 'temperature_change_from_co2', 'population', 'gdp']
15 filtered_df = owid_co2_df[relevant_columns].dropna(subset=['co2', 'temperature_change_from_co2'])
16
17 # convert 'year' to datetime format for easier handling in time-series analysis
18 filtered_df['year'] = pd.to_datetime(filtered_df['year'], format='%Y')
19
20 # aggregate data by year
21 global_aggregated_df = filtered_df.groupby('year').agg({
22     'co2': 'sum',
23     'temperature_change_from_co2': 'mean'
24 }).reset_index()
25
26 # display the cleaned and aggregated dataset
27 global_aggregated_df.head()
```



	year	co2	temperature_change_from_co2
0	1851-01-01	1142.459	0.000047
1	1852-01-01	1190.605	0.000233
2	1853-01-01	1241.990	0.000372
3	1854-01-01	1463.390	0.000535
4	1855-01-01	1483.685	0.000491

```
1 import pandas as pd
2
3 # Load data set
4 owid_co2_df = pd.read_csv('owid-co2-data.csv')
5
6 # Select relevant columns
7 relevant_columns = ['year', 'country', 'co2', 'temperature_change_from_co2', 'population', 'gdp']
8 df_relevant = owid_co2_df[relevant_columns]
9
10 # Calculate the number of missing values before cleaning
11 initial_missing_values = df_relevant.isna().sum().sum()
12
13 # Data cleaning: drop rows with missing values in 'co2' and 'temperature_change_from_co2'
14 filtered_df = df_relevant.dropna(subset=['co2', 'temperature_change_from_co2'])
15
16 # Calculate the number of missing values after cleaning
17 remaining_missing_values = filtered_df.isna().sum().sum()
18
19 # Calculate the number of values processed (cleaned)
20 processed_values_count = initial_missing_values - remaining_missing_values
```

```

21
22 # Print the results
23 print(f"Initial number of missing values: {initial_missing_values}")
24 print(f"Remaining number of missing values after cleaning: {remaining_missing_values}")
25 print(f"Number of values processed (cleaned): {processed_values_count}")

➡ Initial number of missing values: 63447
   Remaining number of missing values after cleaning: 12047
   Number of values processed (cleaned): 51400

1 import pandas as pd
2
3 # Load data set
4 owid_co2_df = pd.read_csv('owid-co2-data.csv')
5
6 # Number of rows before cleaning
7 initial_row_count = len(owid_co2_df)
8
9 # Data cleaning: select relevant columns and drop rows with missing values
10 relevant_columns = ['year', 'country', 'co2', 'temperature_change_from_co2', 'population', 'gdp']
11 filtered_df = owid_co2_df[relevant_columns].dropna(subset=['co2', 'temperature_change_from_co2'])
12
13 # Number of rows after cleaning
14 cleaned_row_count = len(filtered_df)
15
16 # Calculate the number of rows processed (cleaned)
17 processed_row_count = initial_row_count - cleaned_row_count
18
19 # Print the results
20 print(f"Initial number of rows: {initial_row_count}")
21 print(f"Number of rows after cleaning: {cleaned_row_count}")
22 print(f"Number of rows processed (cleaned): {processed_row_count}")
23

➡ Initial number of rows: 47415
   Number of rows after cleaning: 24881
   Number of rows processed (cleaned): 22534

```

✓ 2.3: Exploratory Data Analysis (EDA)

2.3.1: Data Overview

Summary of the dataset, including key statistics (mean, median, mode, standard deviation, etc.).

```

1 import pandas as pd
2
3 # compute measures of center & spread
4 stat = {
5     "Metric": ["Mean", "Median", "Mode", "Variance", "Standard Deviation"],
6     "CO2 Emissions": [
7         global_aggregated_df['co2'].mean(),
8         global_aggregated_df['co2'].median(),
9         global_aggregated_df['co2'].mode()[0],
10        global_aggregated_df['co2'].var(),
11        global_aggregated_df['co2'].std()
12    ],
13    "Temperature Change": [
14        global_aggregated_df['temperature_change_from_co2'].mean(),
15        global_aggregated_df['temperature_change_from_co2'].median(),
16        global_aggregated_df['temperature_change_from_co2'].mode()[0],
17        global_aggregated_df['temperature_change_from_co2'].var(),
18        global_aggregated_df['temperature_change_from_co2'].std()
19    ]
20 }
21
22 # format into data frame & print table
23 stat_df = pd.DataFrame(stat)
24 print(stat_df)

```

```

➡

```

	Metric	CO2 Emissions	Temperature Change
0	Mean	4.847509e+04	0.008590
1	Median	2.094003e+04	0.008014
2	Mode	1.142459e+03	0.000047
3	Variance	2.648739e+09	0.000032

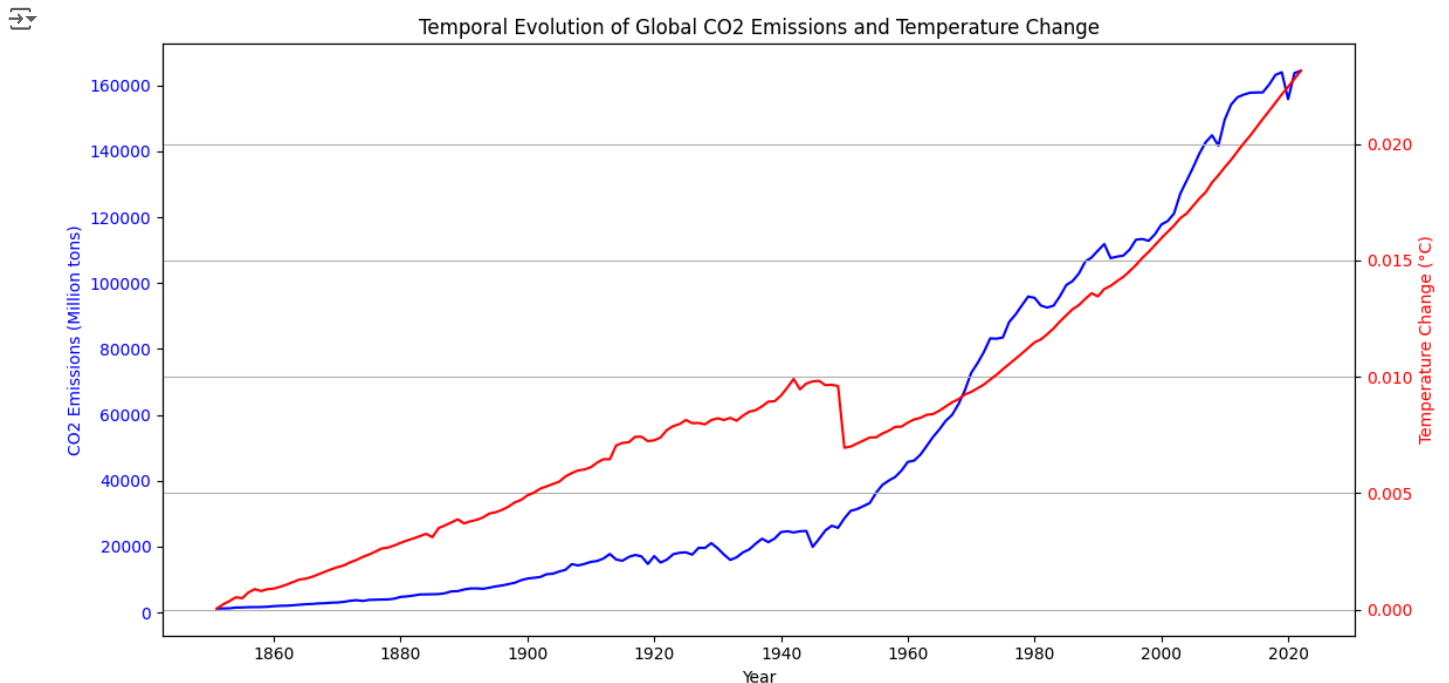
Interpretation (Based on Table Above):

- CO2 Emissions: The statistics presented in the table show a large range of values, indicating CO2 emissions are heavily skewed and have seen large fluctuations over time due to: industrialization of emerging countries (e.g. India, China, etc) industrialization, policy decisions, and economic factors.
- Temperature: We see temperature change has some skew to it, but with less variability as compared to CO2 emissions. These shifts in temperature indicate the need for further exploration between these variables.

2.3.2: Visualizations

Time Series Plot (Aggregate): Temperature vs Year, CO2 Emissions vs Year

```
1 import matplotlib.pyplot as plt
2 import pandas as pd
3
4 # plot temporal evolution of CO2 emissions and temperature change
5 fig, ax1 = plt.subplots(figsize=(12, 6))
6
7 # plot CO2 emissions
8 ax1.plot(global_aggregated_df['year'], global_aggregated_df['co2'], color='blue', label='Global CO2 Emissions')
9 ax1.set_xlabel('Year')
10 ax1.set_ylabel('CO2 Emissions (Million tons)', color='blue')
11 ax1.tick_params(axis='y', labelcolor='blue')
12
13 # create second y-axis for temperature change
14 ax2 = ax1.twinx()
15 ax2.plot(global_aggregated_df['year'], global_aggregated_df['temperature_change_from_co2'], color='red', label='Temperature Change from C
16 ax2.set_ylabel('Temperature Change (°C)', color='red')
17 ax2.tick_params(axis='y', labelcolor='red')
18
19 # title and grid
20 plt.title('Temporal Evolution of Global CO2 Emissions and Temperature Change')
21 fig.tight_layout()
22 plt.grid(True)
23 plt.show()
```



The time-series plot, as shown above, describes the temporal evolution of global CO2 emissions (in blue) and the associated temperature change attributed to CO2 (in red) from 1850 to present. In particular, there is a not so surprising but trouble trend in increasing CO2 emissions post-industrial revolution with a faster increase around the mid-20th century. We also see temperature change rising rapidly, in effect, increasing with emissions (indicating there is very strong relationship between these two).

✓ 2.4: Bulding a Regression Model

✓ 2.4.1: Overview

✓ Generalized Linear Model (Matrix Form)

The generalized linear model can be expressed in matrix form as follows:

$$\mathbf{y} = \mathbf{X} \cdot \boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where:

- \mathbf{X} is the design matrix,
- $\boldsymbol{\beta}$ is the coefficient vector,
- $\boldsymbol{\epsilon}$ is the vector containing error terms such that $\epsilon_1, \epsilon_2, \dots, \epsilon_n \sim \text{iid } N(0, \sigma^2)$.

Explicitly, we can write:

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_{11} & \cdots & x_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nk} \end{pmatrix} \cdot \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_k \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$

Estimating Coefficients

Finding the exact coefficients of $\boldsymbol{\beta}$ might be hard to find numerically, so we can get an estimate by minimizing the distance from $\mathbf{y} - \mathbf{X}\boldsymbol{\beta}$ to the column space of \mathbf{X} . In particular, define:

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\| = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}.$$

Thus for our model:

$$y = \beta_0 + \beta_1 x + \epsilon$$

where:

- y is the predicted temperature ($^{\circ}\text{C}$) change,
- β_0 is the intercept ($\approx 0.00355^{\circ}\text{C}$),
- β_1 is the slope ($\approx 1.039 \times 10^{-7} ^{\circ}\text{C}$ per million tons of CO_2).

Validity of the Linear Regression Model

We will now compute the Pearson correlation coefficient to determine how well our linear model fits the data. In particular, for random variables X and Y , define:

$$r = \frac{\sum_{j=1}^n (X_j - \bar{X})(Y_j - \bar{Y})}{\sqrt{\text{Var}(X)\text{Var}(Y)}} = \frac{\text{Cov}(X, Y)}{\sqrt{\text{Var}(X)\text{Var}(Y)}} \approx 0.9441373.$$

An r value that is close to ± 1 indicates there is a strong relationship between the variables. Indeed, this is the case since r is sufficiently close to 1 and furthermore $r^2 \approx 0.8913$, which indicates our model is a good predictor.

Hypothesis Testing

Since we have built a regression model, it makes sense to use a regression t-test with significance level $\alpha = 0.05$. Define the null and alternate hypotheses as follows:

$$H_0 : \beta_1 = 0 \quad (\text{CO}_2 \text{ emissions have no effect on temperature change})$$

$$H_1 : \beta_1 \neq 0 \quad (\text{CO}_2 \text{ emissions have an effect on temperature change}).$$

Now compute the test statistic and p-value:

$$T_{\text{obs}} = \frac{\beta_1}{\text{SE}(\beta_1)} \approx 37.353838,$$

$$\mathbb{P}r_0(T \geq T_{\text{obs}} \mid H_0 \text{ true}) = \mathbb{P}r_0(T \geq T_{\text{obs}}) \approx 7.209087 \times 10^{-84}.$$

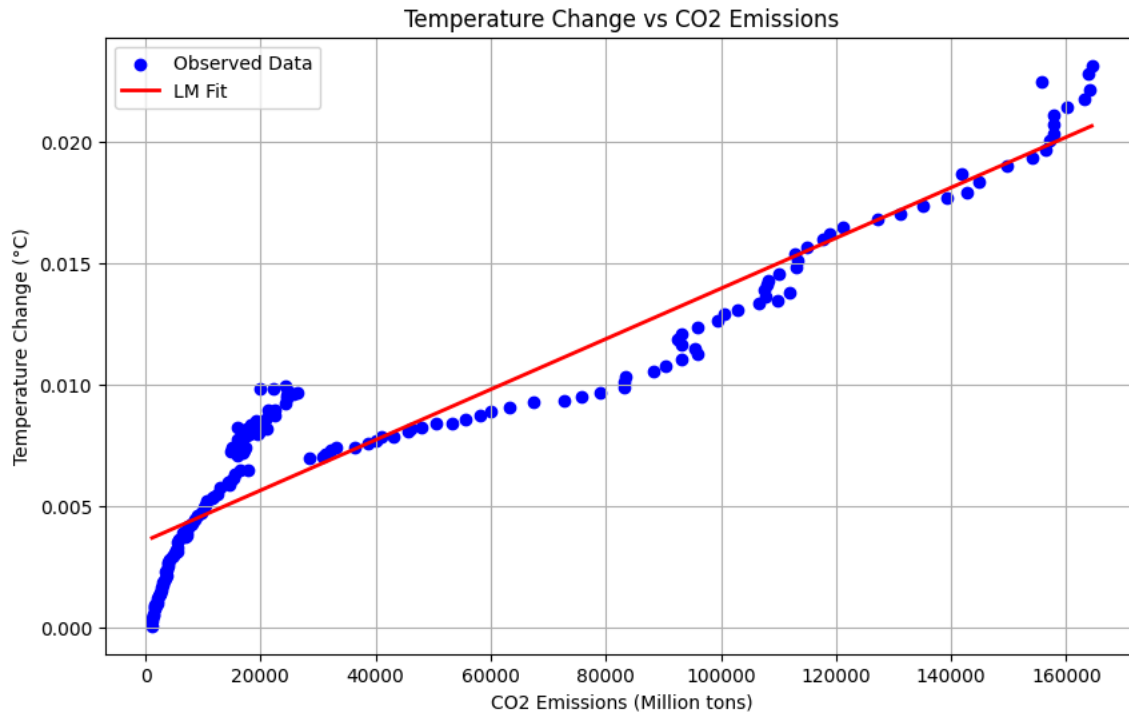
Conclusion

Since $p \approx 7.209087 \times 10^{-84} \ll \alpha = 0.05$, we reject the null hypothesis H_0 and conclude that there is a statistically significant relationship between CO₂ emissions and global temperature change. The slope coefficient β_1 is significantly different from zero, indicating that CO₂ emissions have a substantial impact on temperature anomalies.

```

1 import matplotlib.pyplot as plt
2 import numpy as np
3 from sklearn.linear_model import LinearRegression
4 import statsmodels.api as sm
5 from scipy.stats import pearsonr
6
7 # get vars
8 X = global_aggregated_df['co2'].values.reshape(-1, 1)
9 y = global_aggregated_df['temperature_change_from_co2'].values
10
11 # fit LM
12 model = LinearRegression()
13 model.fit(X, y)
14
15 # predict the temperature change as a function of CO2 emissions
16 y_pred = model.predict(X)
17
18 # plot original data & LM
19 plt.figure(figsize=(10, 6))
20 plt.scatter(X, y, color='blue', label='Observed Data')
21 plt.plot(X, y_pred, color='red', linewidth=2, label='LM Fit')
22
23 # labels
24 plt.xlabel('CO2 Emissions (Million tons)')
25 plt.ylabel('Temperature Change (°C)')
26 plt.title('Temperature Change vs CO2 Emissions')
27 plt.legend()
28 plt.grid(True)
29 plt.show()
30
31 # make in intercept form y = mx + b
32 X_c = sm.add_constant(X)
33
34 # fit LM model
35 model = sm.OLS(y, X_c)
36 results = model.fit()
37
38 # compute t-statistic and p-value for slope & display
39 t_stat = results.tvalues[1]
40 p_val = results.pvalues[1]
41
42 print(f"T-Stat for β(1): {t_stat}")
43 print(f"p-value for β(1): {p_val}")
44
45 X_flat = X.flatten()
46 y_flat = y.flatten()
47
48 # compute Pearson correlation coefficient (r) & display
49 correlation_coefficient = np.corrcoef(X_flat, y_flat)[0, 1]
50 print(f"r: {correlation_coefficient}")

```



T-Stat for $\beta(1)$: 37.35383840253067
 p-value for $\beta(1)$: 7.209086865633394e-84
 r: 0.9441373110790902

✓ 2.5: Machine Learning Techniques (SGD) & Exploration

✓ 2.5.1: Overview

Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is an iterative optimization algorithm designed to minimize a loss function. It updates the model coefficients by computing the gradient of the loss with respect to the coefficients.

Update Rule

$$\beta^{(t+1)} = \beta^{(t)} - \eta \nabla_{\beta} J(\beta)$$

where:

- β : Coefficients of the model
- $J(\beta)$: Loss function (e.g., Mean Squared Error)
- $\nabla_{\beta} J(\beta)$: Gradient of the loss function with respect to β
- η : Learning rate (step size)

Gradient for MSE Loss

The Mean Squared Error (MSE) is defined as: $J(\beta) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$

where:

- y_i : True target value
- \hat{y}_i : Predicted value $\hat{y}_i = \beta_0 + \beta_1 x_i$ for simple regression
- N : Total number of samples

To compute the gradient with respect to β :

$$\nabla_{\beta} J(\beta) = \frac{\partial}{\partial \beta} \frac{1}{N} \sum_{i=1}^N (y_i - \beta_0 - \beta_1 x_i)^2$$

Expanding and differentiating:

$$\nabla_{\beta} J(\beta) = -\frac{2}{N} \sum_{i=1}^N (y_i - \hat{y}_i) x_i$$

SGD updates the coefficients incrementally using this gradient computed over small batches of data.

✓ 2.5.2: Using the Algorithm

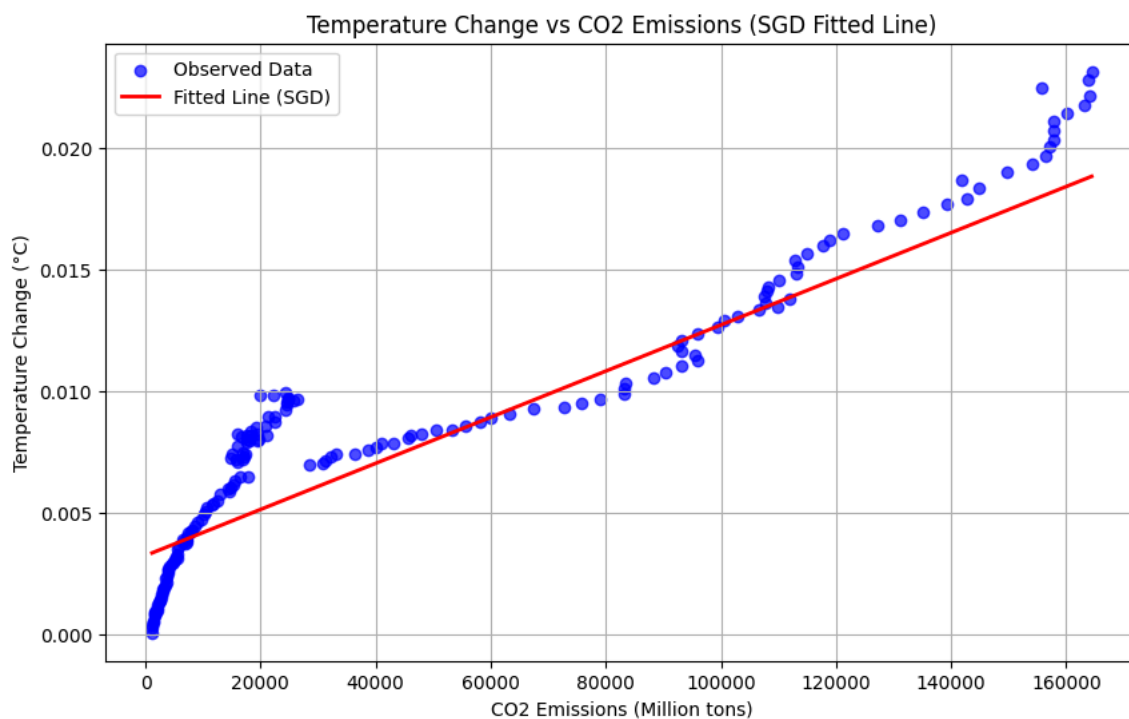
```
1 from sklearn.linear_model import SGDRegressor
2 from sklearn.preprocessing import StandardScaler
3
4 # Standardize the data
5 scaler = StandardScaler()
6 X_scaled = scaler.fit_transform(X)
7
8 # Fit SGD regressor
9 sgd_regressor = SGDRegressor(max_iter=10000, tol=1e-3, eta=0.01, random_state=42)
10 sgd_regressor.fit(X_scaled, y)
11
12 # Print coefficients
13 print(f"Coefficients: {sgd_regressor.coef_}, Intercept: {sgd_regressor.intercept_}")
```

↗ Coefficients: [0.00486941], Intercept: [0.00782038]

✓ 2.5.3: Visualization & Remark

```
1 # Predict values
2 y_pred = sgd_regressor.predict(X_scaled)
3
4 # Plot observed vs fitted values
5 plt.figure(figsize=(10, 6))
6 plt.scatter(X, y, color='blue', label='Observed Data', alpha=0.7)
7 plt.plot(X, y_pred, color='red', linewidth=2, label='Fitted Line (SGD)')
8
9 # Adding labels and title
10 plt.xlabel('CO2 Emissions (Million tons)')
11 plt.ylabel('Temperature Change (°C)')
12 plt.title('Temperature Change vs CO2 Emissions (SGD Fitted Line)')
13 plt.legend()
14 plt.grid(True)
15 plt.show()
```

↗




```
1 from sklearn.metrics import mean_squared_error, r2_score
```



```

2
3 # Predictions from SGD model
4 y_pred_sgd = sgd_regressor.predict(X_scaled)
5
6 # Compute MSE and R-squared
7 mse_sgd = mean_squared_error(y, y_pred_sgd)
8 r2_sgd = r2_score(y, y_pred_sgd)
9
10 # Display the statistics
11 print(f"Mean Squared Error (MSE) for SGD Regression: {mse_sgd}")
12 print(f"R-squared (R^2) for SGD Regression: {r2_sgd}")
13

```

 Mean Squared Error (MSE) for SGD Regression: 4.2757572477900575e-06
R-squared (R^2) for SGD Regression: 0.8660485071831118

Remark: This has slightly worse performance than the regression model from above, but nonetheless agrees with the general result from above.

2.6: Mathematical Statistics

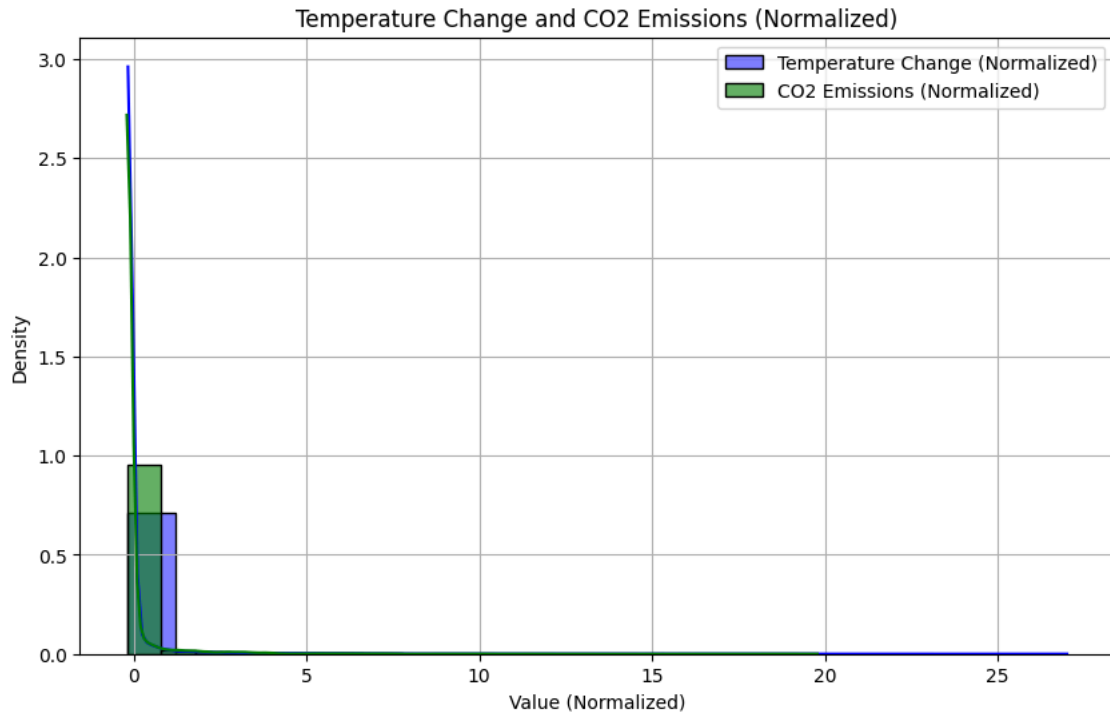
2.6.1: Methodology

After analyzing the data extensively, by building a strong predictive model, we should explore other techniques to estimate the data (such as an MLE: Maximum Likelihood Estimate). For an even better interpolation of the data, we could incorporate this into the LM above (or use a different model altogether). To do so, we will first determine the shape of the data, to fit some sort of distribution.

```

1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import numpy as np
5 from scipy.stats import lognorm
6
7 owid_co2_df = pd.read_csv('owid-co2-data.csv')
8
9 # extract data
10 temperature_change_data = owid_co2_df['temperature_change_from_co2'].dropna()
11 co2_emissions_data = owid_co2_df['co2'].dropna()
12
13 # normalize
14 temperature_change_data_norm = (temperature_change_data - temperature_change_data.mean()) / temperature_change_data.std()
15 co2_emissions_data_norm = (co2_emissions_data - co2_emissions_data.mean()) / co2_emissions_data.std()
16
17 # plot the overlayed histograms
18 plt.figure(figsize=(10, 6))
19
20 sns.histplot(temperature_change_data_norm, bins=20, kde=True, color='blue', stat='density', label='Temperature Change (Normalized)')
21 sns.histplot(co2_emissions_data_norm, bins=20, kde=True, color='green', stat='density', label='CO2 Emissions (Normalized)', alpha=0.6)
22
23 plt.xlabel('Value (Normalized)')
24 plt.ylabel('Density')
25 plt.title('Temperature Change and CO2 Emissions (Normalized)')
26 plt.legend()
27 plt.grid(True)
28 plt.show()
29
30 # this part is done after exploring the shape of the data and various distributions
31 # it was determined a log-normal distribution is appropriate here (based on the shape)
32 # filter non-positive values (log-normal requires positive values)
33 temperature_change_data = temperature_change_data[temperature_change_data > 0]
34
35 # fit log-normal distribution to temperature change data
36 shape, loc, scale = lognorm.fit(temperature_change_data, floc=0)
37
38 # compute log-likelihood & display results
39 log_likelihood = np.sum(lognorm.logpdf(temperature_change_data, shape, loc, scale))
40
41 print(f"Shape (σ): {shape}")
42 print(f"Location (μ): {loc}")
43 print(f"Scale (exp(μ)): {scale}")
44 print(f"Log-Likelihood: {log_likelihood}")

```



2.6.2: Remark

We should note that log-likelihood is a measure of how well the Log-Normal distribution fits the observed data (temperature change). The higher the log-likelihood estimate, the better it will fit (this is demonstrated by the results above).

✓ 2.6.3: Theoretical Results

- The calculations which we will now show, make for better estimates of the data and in conjunction with other techniques (apply transformations to the x-or y axes, etc) make for a better predictor.

Log-Normal Distribution Analysis

Maximum Likelihood Estimation (MLE)

Suppose Y_1, \dots, Y_n follow a log-normal distribution with parameters μ_Y and σ_Y^2 . This means that $\ln(Y)$ is distributed normally. The density function for the log-normal distribution is defined as follows:

$$f_{Y_j}(y_j | \mu_Y, \sigma_Y^2) = \frac{1}{y_j \sigma_Y \sqrt{2\pi}} \cdot \exp\left(-\frac{(\ln y_j - \mu_Y)^2}{2\sigma_Y^2}\right).$$

i. Finding the MLE for μ_Y and σ_Y^2

Let the likelihood function be:

$$\text{lik}(\mu_Y) = f(y_1, \dots, y_n | \mu_Y, \sigma_Y^2) = \prod_{j=1}^n f_{Y_j}(y_j | \mu_Y, \sigma_Y^2),$$

since Y_1, Y_2, \dots, Y_n are independent and identically distributed random variables. Therefore:

$$\text{lik}(\mu_Y) = \prod_{j=1}^n \frac{1}{y_j \sigma_Y \sqrt{2\pi}} \cdot \exp\left(-\frac{(\ln y_j - \mu_Y)^2}{2\sigma_Y^2}\right).$$

Simplifying, we can write:

$$\text{lik}(\mu_Y) = \frac{1}{\sigma_Y^n (2\pi)^{n/2}} \prod_{j=1}^n \frac{1}{y_j} \cdot \exp\left(-\frac{\sum_{j=1}^n (\ln y_j - \mu_Y)^2}{2\sigma_Y^2}\right).$$

Taking the log-likelihood function:

$$\ell(\mu_Y) = n \log\left(\frac{1}{\sigma_Y \sqrt{2\pi}}\right) + \sum_{j=1}^n \log\left(\frac{1}{y_j}\right) - \frac{\sum_{j=1}^n (\ln y_j - \mu_Y)^2}{2\sigma_Y^2}.$$

Differentiating with respect to μ_Y and setting the derivative to zero:

$$\ell'(\mu_Y) = \frac{\sum_{j=1}^n (\ln y_j - \mu_Y)}{\sigma_Y^2} = 0,$$

which gives:

$$\hat{\mu}_Y = \frac{1}{n} \sum_{j=1}^n \ln y_j.$$

To confirm this is the MLE, we compute the second derivative:

$$\ell''(\mu_Y) = -\frac{1}{\sigma_Y^2}.$$

Since this is negative, the condition for a maximum is satisfied.

Next, for σ_Y^2 , differentiate the log-likelihood function:

$$\ell'(\sigma_Y^2) = -\frac{n}{\sigma_Y^2} + \frac{\sum_{j=1}^n (\ln y_j - \mu_Y)^2}{\sigma_Y^4}.$$

Setting this equal to zero, we find:

$$\hat{\sigma}_Y^2 = \frac{1}{n} \sum_{j=1}^n (\ln y_j - \mu_Y)^2.$$

The second derivative with respect to σ_Y^2 is:

$$\ell''(\sigma_Y^2) = \frac{n}{\sigma_Y^2} - \frac{3}{\sigma_Y^4} \sum_{j=1}^n (\ln y_j - \mu_Y)^2.$$

This is also negative, satisfying the condition for a maximum.

ii. Asymptotic Distribution of the MLE for μ_Y

The Fisher Information for $\hat{\mu}_Y$ is given by:

$$I(\hat{\mu}_Y) = -E[\ell''(\mu_Y)] = \frac{1}{\sigma_Y^2}.$$

Thus, the asymptotic distribution of $\hat{\mu}_Y$ is:

$$\hat{\mu}_Y \sim N\left(\mu_Y, \frac{\sigma_Y^2}{n}\right).$$

iii. Transformation of Variables

Suppose $X_j = \ln Y_j$, so that X_j follows $N(\mu_X, \sigma_X^2)$ for $j = 1, \dots, n$. Using the ML estimators for the parameters of a normal distribution, we can find the estimators for μ_X and σ_X^2 in terms of Y .

Using the delta method, define $g(x) = \ln x$ with $g'(x) = \frac{1}{x}$. Then:

$$\mu_X \approx E[g(X)] = \ln \mu_Y,$$

and

$$\text{Var}(g(X)) \approx \sigma_X^2 \cdot (g'(\mu_X))^2 = \frac{\sigma_Y^2}{\mu_Y^2}.$$

By the equivariance property of the MLE:

$$g(\hat{\mu}_X) = g(\mu_X), \quad g(\hat{\sigma}_X^2) = g(\sigma_X^2).$$

If desired, we could write the estimators for μ_X and σ_X^2 in terms of X by substituting the above MLEs and performing additional algebraic manipulation.

✓ 2.7: Insights & Conclusion

2.7.1: Insights (Relating to Environmental Policy & Decisions)

1. Accelerate the Transition to Renewable Energy:

- Rationale: The strong correlation between CO2 emissions and temperature change highlights the urgent need to reduce fossil fuel consumption. Transitioning to renewable energy sources such as wind, solar, and hydroelectric power can significantly decrease CO2 emissions, slowing the rate of global warming.
- ◦ Recommendation: Governments and industries should increase investments in renewable energy infrastructure, provide subsidies or incentives for renewable energy projects, and phase out subsidies for fossil fuels.

2. Implement and Enforce Carbon Pricing:

- Rationale: By putting a price on carbon emissions, either through carbon taxes or cap-and-trade systems, the true environmental cost of CO2 emissions is internalized. This encourages businesses and consumers to reduce their carbon footprint and invest in cleaner technologies.
- ◦ Recommendation: Policymakers should introduce or strengthen carbon pricing mechanisms to create a financial incentive for reducing emissions. Revenue from carbon pricing can be reinvested in green technologies or returned to citizens as rebates.

3. Promote Energy Efficiency:

- Rationale: Improving energy efficiency in buildings, transportation, and industrial processes can lead to significant reductions in CO2 emissions without compromising economic growth. This is a cost-effective way to lower emissions across multiple sectors.
- ◦ Recommendation: Implement stricter energy efficiency standards for appliances, vehicles, and industrial equipment. Encourage retrofitting of existing buildings and infrastructure with energy-efficient technologies.

4. Enhance Reforestation and Afforestation Efforts:

- Rationale: Forests act as carbon sinks, absorbing CO2 from the atmosphere. Enhancing reforestation (replanting in deforested areas) and afforestation (planting in new areas) efforts can help offset emissions.
- ◦ Recommendation: Launch large-scale tree-planting initiatives, protect existing forests, and restore degraded lands. Support policies that incentivize sustainable land use and forest management.

5. Support Technological Innovation in Carbon Capture and Storage (CCS):

- Rationale: Carbon capture and storage (CCS) technologies can capture CO2 emissions at their source (e.g., power plants, industrial facilities) and store them underground or use them in other processes. This can help manage emissions from industries where reductions are otherwise difficult.
- ◦ Recommendation: Increase funding for research and development of CCS technologies, create regulatory frameworks that encourage the adoption of CCS, and integrate these technologies into existing energy infrastructure.

6. Educate and Engage the Public on Climate Change:

- Rationale: Public awareness and understanding of climate change are crucial for building support for climate policies and encouraging behavior change. Informed citizens are more likely to adopt sustainable practices and support necessary policies.
- ◦ Recommendation: Develop and implement education campaigns that inform the public about the causes and consequences of climate change and the steps they can take to reduce their carbon footprint. Engage communities through local initiatives and empower individuals to take action.

7. Strengthen International Cooperation:

- Rationale: Climate change is a global issue that requires coordinated international efforts. No single country can effectively address climate change alone.
- ◦ Recommendation: Strengthen international agreements like the Paris Agreement, increase financial and technical support to developing countries for climate adaptation and mitigation, and encourage countries to set and achieve more ambitious emission reduction targets.

2.7.2: Conclusion

After completing ENVECON 105, I was inspired to take on this project as an opportunity to apply the skills I had developed in class. In ENVECON 105, I learned to analyze real-world data using Python, focusing on applications like environmental justice, emissions monitoring, and sustainability assessments. This project allowed me to build upon those foundations and further explore the critical relationship between CO2 emissions and temperature changes, which is at the heart of the climate change debate.

I approached this project with the intent of synthesizing theoretical knowledge from previous courses with practical, hands-on data analysis techniques. From data cleaning and visualization to statistical modeling, I worked to ensure a balance between rigorous methodology and actionable insights. The work was both challenging and rewarding, particularly as I applied advanced statistical techniques and programming tools to analyze real-world data in a meaningful way.

Working on this project independently gave me a deeper appreciation for the complexity of climate data and the critical role of analysts in communicating findings effectively. Beyond the technical work, it reaffirmed the importance of presenting results in a way that is accessible to decision-makers and the public, empowering them to make informed choices about policy and daily habits.

Through this analysis, I have not only deepened my understanding of the data but also reinforced my commitment to using data-driven approaches to address pressing environmental challenges. This project demonstrates how the skills I gained in ENVECON 105 and my other courses can be applied to real-world problems, contributing to the development of policies and solutions that promote sustainability.

It has been a fulfilling experience to use my education to explore these critical issues further. I am always working on refining my skills and I'm confident the skills I have gained throughout my time working on this project will prove useful in future endeavors, and can be improved upon in the future.

2.8: Reference(s)