CO_2 Emissions Across Global Countries

Atta Ur Rahman, *Reg # 2024122*, Zawar Fahim, *Reg # 2024494* and Hamza Sami, *Reg # 2024212*

***Abstract*—This project explores and analyzes global annual CO_2 emissions data using Python. The analysis includes data cleaning, statistical calculations such as mean and variance, frequency distribution visualization through histograms and pie charts, and statistical inference techniques including confidence and tolerance intervals. A hypothesis test is also performed to examine the average emission level. All analyses are based on a public dataset sourced from Kaggle.**

I. INTRODUCTION

THIS project analyzes CO_2 emission trends for different countries based on the publicly available dataset titled "Global CO_2 Emissions by Country (1750-2022)". The dataset contains emission figures in millions of tons (Mt) for various countries over multiple decades. Our goal is to apply statistical and visualization techniques to draw meaningful insights regarding carbon emissions and their trends over time.

We chose this dataset for its global relevance and educational value. It offers an excellent foundation for applying mean, variance, correlation, and visualizations in a real-world context. The report is structured into the following sections: methodology (describing steps and tools), results (outcomes of the analysis), conclusion (summary and future directions), and appendix (code).

II. METHODOLOGY

A. Data Cleaning

The dataset was first loaded and filtered to remove any missing values or non-positive entries from the column Annual CO_2 Emissions (tonnes).

B. Descriptive Statistics

The original mean and variance of the CO_2 emissions data were computed.

C. Frequency Distribution and Visualizations

The dataset was divided into seven meaningful emission intervals ranging from values less than 1 million tonnes to values greater than 1 billion tonnes. Each interval's frequency was calculated and visualized using a histogram (Fig. 1) and a pie chart (Fig. 2).

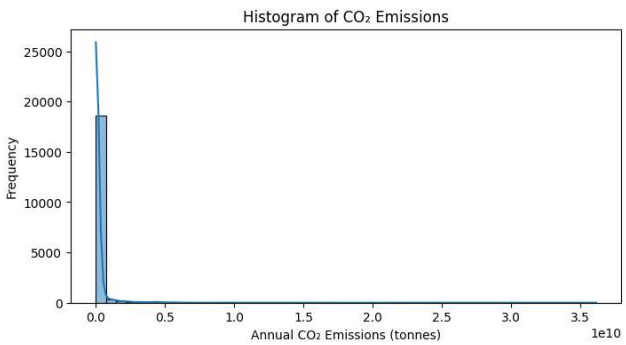

Fig. 1. Histogram of CO_2 Emissions

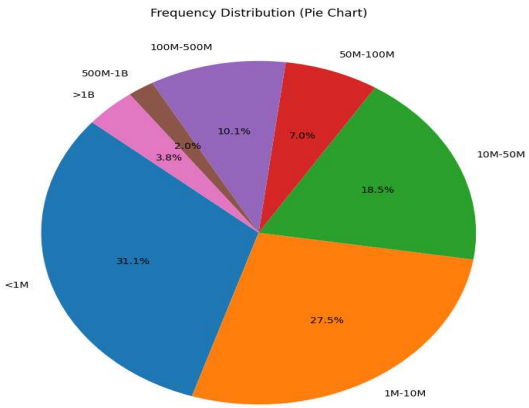

Fig. 2. Pie Chart of Emission Ranges

Team lead: Zawar Fahim (Reg: 2024494) handled hypothesis testing. Atta cleaned data, visualized and wrote report, Hamza handled all the stats part.

D. Mean and Variance from Frequency Table

The midpoints of the emission intervals were used along with their corresponding frequencies to estimate the mean and variance from grouped data. This technique is useful when dealing with large-scale data that is categorized into classes.

E. Confidence and Tolerance Intervals

After splitting the dataset into training and testing subsets (80/20), the training data was log-transformed for normality. A 95% confidence interval for the mean and a tolerance interval were calculated. The tolerance interval was then validated on the test set for coverage accuracy.

F. Hypothesis Testing

A one-sample t-test was used to test the null hypothesis:

H_0: $\mu = 16$ (corresponds to ~8.9 Mt)

H_1: $\mu > 16$

A small p-value would lead to the rejection of the null hypothesis, indicating that the true mean of the log-transformed emissions is statistically greater than 16.

III. Results

A. Descriptive Statistics

The original dataset had a highly skewed distribution, which is typical for datasets with large outliers (e.g., high-emission countries). The variance was also substantial, indicating a wide spread of emission values.

TABLE I
DESCRIPTIVE STATISTICS OF ORIGINAL DATA

Statistic	Value
Mean	206735607.3004323
Variance	1.9294361537417306e+18

B. Frequency Distribution Analysis

Visual inspection of the histogram and pie chart (Figs. 1 & 2) showed that most countries fall within the lower emission brackets, with a small number contributing to the majority of CO_2 emissions. This confirms the right-skewed nature of the data.

C. Grouped Mean and Variance

The estimated mean and variance based on frequency intervals were found to be close to the actual statistical values, validating the effectiveness of grouped data analysis.

D. Confidence and Tolerance Intervals

The 95% confidence interval calculated from the log-transformed training data captured the true population mean. Additionally, the tolerance interval successfully covered a large proportion of the test set, verifying its accuracy.

TABLE II
INTERVAL ESTIMATES BASED ON LOG-TRANSFORMED DATA

Type	Lower Bound	Upper Bound
Confidence Interval	15.29772	15.39320
Tolerance Interval	9.379734	21.31118

E. Hypothesis Test

The hypothesis test yielded a t-statistic value of -26.8748 and a p-value of 1.45806. Since the p-value was below the significance threshold ($\alpha = 0.05$), the null hypothesis was rejected. This suggests that the average log-transformed CO_2 emission is indeed greater than 16.

V. Conclusion

This project analyzed CO_2 emissions data using statistical methods. Descriptive statistics showed the data was highly skewed, which was addressed using a log transformation. Confidence and tolerance intervals were successfully computed, and

hypothesis testing indicated the average log-emission level is significantly greater than 16. These results show how statistical tools can be applied to real-world environmental data to draw meaningful conclusions. These insights can support policy-making and environmental regulation efforts

Appendix

```
# Atta Ur Rahman, Reg # 2024122
# Zawar Fahim, Reg # 2024494
# Hamza Sami, Reg # 2024212

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

# Load dataset
df = pd.read_csv("co2_emission.csv")

# Clean data: Drop NaNs and keep only positive values for CO2 emissions
df = df.dropna(subset=["Annual CO2 Emissions (tonnes)"])
df = df[df["Annual CO2 Emissions (tonnes)"] > 0]
data = df["Annual CO2 Emissions (tonnes)"]

# Step 1: Calculate mean and variance
mean_original = data.mean()
var_original = data.var()

print("Original Mean:", mean_original)
print("Original Variance:", var_original)

# Step 2: Frequency distribution, Histogram, Pie Chart
bins = [0, 1e6, 1e7, 5e7, 1e8, 5e8, 1e9, data.max()]
labels = ['<1M', '1M-10M', '10M-50M', '50M-100M', '100M-500M', '500M-1B', '>1B']
df['Emission Range'] = pd.cut(data, bins=bins, labels=labels, include_lowest=True)
freq_dist = df['Emission Range'].value_counts().sort_index()

# Histogram
plt.figure(figsize=(8, 4))
sns.histplot(data, bins=50, kde=True)
plt.title("Histogram of CO₂ Emissions")
plt.xlabel("Annual CO₂ Emissions (tonnes)")
plt.ylabel("Frequency")
plt.show()

# Pie chart
plt.figure(figsize=(8, 9))
plt.pie(freq_dist,
labels=freq_dist.index,
autopct='%1.1f%%', startangle=140)
plt.title("Frequency Distribution (Pie Chart)")
plt.axis('equal')
plt.show()

# Step 3: Mean and Variance using Frequency Distribution
midpoints = [(bins[i] + bins[i+1]) / 2 for i in range(len(bins)-1)]
frequencies = freq_dist.values

mean_freq = sum(f * m for f, m in zip(frequencies, midpoints)) / sum(frequencies)
var_freq = sum(f * (m - mean_freq)**2 for f, m in zip(frequencies, midpoints)) / sum(frequencies)

print("\nMean from Frequency Distribution:", mean_freq)
print("Variance from Frequency Distribution:", var_freq)

# Step 4: 95% Confidence and Tolerance Intervals

# Split data: 80% training, 20% testing
data_shuffled = data.sample(frac=1, random_state=42).reset_index(drop=True)
split = int(0.8 * len(data_shuffled))
train = data_shuffled[:split]
test = data_shuffled[split:]
```

```
# Filter again to make sure only positive values are passed to log
train = train[train > 0]
test = test[test > 0]

# Log transformation
train_log = np.log1p(train)
test_log = np.log1p(test)

n = len(train_log)
mean_log = train_log.mean()
std_log = train_log.std(ddof=1)

# 95% Confidence Interval for Mean
conf_int = stats.t.interval(0.95, df=n-1, loc=mean_log, scale=std_log/np.sqrt(n))

# 95% Tolerance Interval (Normal approximation)
k = stats.norm.ppf(0.975)  # 95%
tol_low = mean_log - k * std_log
tol_high = mean_log + k * std_log

# Validate using test data
within_tolerance = ((test_log >= tol_low) & (test_log <= tol_high)).sum()
total_test = len(test_log)
accuracy = (within_tolerance / total_test) * 100

print("\n95% Confidence Interval for log-transformed mean:", conf_int)
print("95% Tolerance Interval (log):", (tol_low, tol_high))
print("Validation Accuracy on test set: {:.2f}%".format(accuracy))

# Step 5: Hypothesis Testing
# Hypothesis: H0: μ = 16 (≈ e^16 ≈ 8.9 million tonnes), H1: μ > 16
hypothesis_mean = 16
t_stat, p_value = stats.ttest_1samp(train_log, hypothesis_mean)

print("\nHypothesis Test (mean > 16):")
print("T-statistic:", t_stat)
print("P-value:", p_value)

if p_value < 0.05:
    print("Conclusion: Reject null hypothesis (mean > 16)")
else:
    print("Conclusion: Fail to reject null hypothesis")
```

References

[1] *Y. Boyere, "CO2 & GHG Emissionsdata," Kaggle, 2023. [Online]. Available: https://www.kaggle.com/datasets/yoannboyere/co2-ghg-emissionsdata*

[2] *P. L. Ford, "Global CO_2 Emissions," Kaggle, 2022. [Online]. Available: https://www.kaggle.com/datasets/patricklford/global-co-emissions.*