SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, TRICHY

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DATA SCIENCE [21CSS303T]

PROJECT REPORT

April 2025

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY, TRICHY

DATA SCIENCE PROJECT REPORT

NAME	
REG NO	
SEMESTER	
PROGRAME	
ACADEMIC YEAR	

CONTENTS

ABSTRACT:

Air pollution is a critical environmental issue, particularly in urban areas where industrial activities, vehicular emissions, and other anthropogenic factors contribute to rising CO2 and NOx levels. This project aims to collect and analyze pollution datasets to assess air quality trends and their potential impact on urban ecosystems. By leveraging NumPy operations, the data is processed to perform statistical analysis, extract patterns, and detect anomalies in pollution levels.

To effectively interpret and present the findings, various visualization techniques such as histograms, scatter plots, bar charts, and heatmaps are employed. These visualizations help in understanding correlations between different pollutants, identifying peak pollution hours, and assessing variations across different urban locations. The study also explores seasonal influences and possible sources contributing to air pollution.

Findings from this analysis provide actionable insights for environmental researchers, policymakers, and urban planners. The results can aid in the development of evidence-based strategies for pollution control, urban sustainability, and improving air quality standards. By applying data-driven techniques, this study underscores the importance of real-time monitoring and analytical approaches in addressing urban pollution challenges.

INTRODUCTION:

Air pollution is a major global concern, particularly in urban environments where industrial emissions, vehicular exhaust, and other human activities contribute to rising levels of pollutants such as CO2 and NOx. Monitoring and analyzing air quality data is essential for understanding pollution patterns, identifying sources, and formulating effective mitigation strategies.

This project applies **data science techniques** to collect, process, analyze, and visualize air pollution datasets. By leveraging **NumPy for numerical computations** and statistical analysis, the study extracts meaningful insights from large datasets. Various **data visualization techniques**, including histograms, scatter plots, bar charts, and heatmaps, are used to identify trends, correlations, and anomalies in pollution levels over time and across different regions.

The study aims to answer key questions such as:

- How do CO2 and NOx levels vary across different urban locations?
- What are the seasonal trends in air pollution?
- How do specific events (e.g., traffic congestion, industrial activities) affect pollution levels?
- What correlations exist between multiple pollutants and other environmental factors?

By applying **data-driven methodologies**, this project provides actionable insights that can assist policymakers, environmentalists, and urban planners in making informed decisions to improve air quality and promote sustainable urban development. The use of **real-time data analytics and machine learning techniques** in future expansions of the project could further enhance predictive capabilities and pollution control strategies.

OBJECTIVES:

Data Collection & Preprocessing

- Gather air pollution datasets (CO2, NOx, and other pollutants) from reliable sources such as environmental monitoring stations, IoT sensors, or open-source databases.
- Clean, preprocess, and handle missing or inconsistent data to ensure data quality and accuracy.

Exploratory Data Analysis (EDA)

- Use statistical techniques to understand the distribution, trends, and variations in pollution levels over time and locations.
- Identify outliers, anomalies, and potential sources contributing to pollution.

Data Processing Using NumPy

- Apply NumPy operations for numerical computations such as mean, median, standard deviation, and correlation analysis to gain insights into pollution trends.
- Perform mathematical transformations to standardize and normalize data for better analysis.

Data Visualization

- Create effective visual representations (histograms, scatter plots, bar charts, heatmaps) to analyze patterns in pollution levels.
- Identify relationships between different pollutants and external factors such as traffic density, weather conditions, or industrial activities.

IMPORTANCE OF DATA VISUALIZATION:

Data visualization plays a crucial role in this project by transforming complex air pollution datasets into easy-to-understand graphical representations. Since air pollution data consists of multiple variables, including CO2 and NOx levels across different locations and time periods, effective visualization techniques help in identifying trends, patterns, and anomalies. The key importance of data visualization in this project includes:

- Enhanced Data Understanding
- Trend & Pattern Identification
- Correlation & Anomaly Detection
- Effective Communication of Insights

DATA SET DISCRIPTION:

The dataset used in this project comprises air pollution data collected from multiple sources, including AI-generated datasets, Kaggle repositories, and real-world data from the user's minor project. It includes various environmental parameters essential for analyzing pollution levels and their impact on urban air quality.

Key Attributes in the Dataset:

1. **Temporal & Location Data:**
   - **Timestamp:** Date and time of data collection.
   - **Location ID/Name:** Identifies the urban area or monitoring station where data was recorded.
2. **Air Pollution Metrics:**
   - **CO2 Levels (ppm):** Concentration of carbon dioxide in parts per million.

- **NOx Levels (ppm):** Nitrogen oxides concentration in parts per million.
- **PM2.5 (µg/m^3):** Fine particulate matter (2.5 micrometers or smaller) affecting air quality.
- **PM10 (µg/m^3):** Larger particulate matter impacting respiratory health.
- **SO2 Levels (ppm):** Sulfur dioxide concentration.
- **O3 Levels (ppm):** Ozone concentration affecting environmental and human health.

3. **Weather & Environmental Factors:**
    - **Temperature (°C):** Ambient temperature at the time of measurement.
    - **Humidity (%):** Atmospheric moisture level affecting pollution dispersion.
    - **Wind Speed (m/s):** Affects pollution spread and dilution.
    - **Precipitation (mm):** Rainfall data, which impacts pollutant washout.
4. **Traffic & Industrial Influence:**
    - **Traffic Density (vehicles/hour):** Number of vehicles in monitored areas.
    - **Industrial Emissions Index:** Categorizes industrial activity levels contributing to pollution.
5. **Health & Urban Factors (If Available):**
    - **Air Quality Index (AQI):** Standardized metric indicating overall pollution level.
    - **Population Density (people/km^2):** Determines urban exposure to pollutants.

Dataset Size and Format

- **File Formats:** The dataset is stored in **CSV and Excel** formats, making it compatible with data analysis tools like Python (Pandas, NumPy), MATLAB, and Excel-based analytics.
- **Data Size:** The dataset consists of thousands of records, covering multiple time periods and locations for a comprehensive analysis of pollution trends.
- **Structure:**
    - **Rows:** Each row represents a recorded data entry for a specific timestamp and location.
    - **Columns:** Include pollution metrics (CO2, NOx, PM2.5, etc.), weather conditions, traffic density, and other influencing factors.

DATA PROCESSING IN COLLAB:

Air pollution monitoring is essential for understanding environmental changes and their impact on human health. In this project, air pollution datasets containing PM2.5, CO2, and NOx levels are processed and analyzed using data science techniques in Google Collab. This includes data cleaning, preprocessing, statistical analysis, and visualization to extract meaningful insights from pollution data.

Conversion:

```
from google.colab import files

uploaded = files.upload()
```

Reading File:

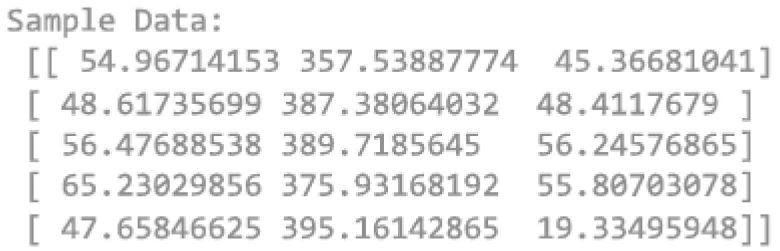
```
import pandas as pd

df = pd.read_excel("pollution.xlsx")

df.head()
```

OPERATIONS PERFORMED USING DATA SET:

Basic Input Function

```
import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns


np.random.seed(42)

num_samples = 100

pm2.5 = np.random.normal(loc=50, scale=10, size=num_samples)

co2 = np.random.normal(loc=400, scale=30, size=num_samples)

nox = np.random.normal(loc=40, scale=15, size=num_samples)

air_pollution_data = np.column_stack((pm25, co2, nox))

columns = ["PM2.5", "CO2", "NOx"]

print("Sample Data:\n", air_pollution_data[:5])
```

O/P



```
Sample Data:
 [[ 54.96714153 357.53887774  45.36681041]
 [ 48.61735699 387.38064032  48.4117679 ]
 [ 56.47688538 389.7185645   56.24576865]
 [ 65.23029856 375.93168192  55.80703078]
 [ 47.65846625 395.16142865  19.33495948]]
```

Basic Statistics

mean_values = np.mean(air_pollution_data, axis=0)

median_values = np.median(air_pollution_data, axis=0)

std_values = np.std(air_pollution_data, axis=0)

min_values = np.min(air_pollution_data, axis=0)

max_values = np.max(air_pollution_data, axis=0)

print("Mean:", mean_values)

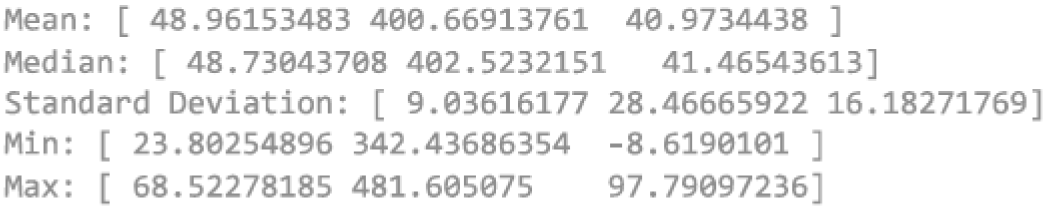
print("Median:", median_values)

print("Standard Deviation:", std_values)

print("Min:", min_values)

print("Max:", max_values)

O/P



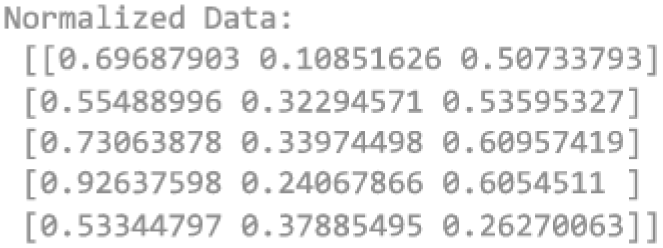
```
Mean: [ 48.96153483 400.66913761  40.9734438 ]
Median: [ 48.73043708 402.5232151   41.46543613]
Standard Deviation: [ 9.03616177 28.46665922 16.18271769]
Min: [ 23.80254896 342.43686354  -8.6190101 ]
Max: [ 68.52278185 481.605075    97.79097236]
```

Normalize

normalized_data = (air_pollution_data - min_values) / (max_values - min_values)

print("Normalized Data:\n", normalized_data[:5])

O/P



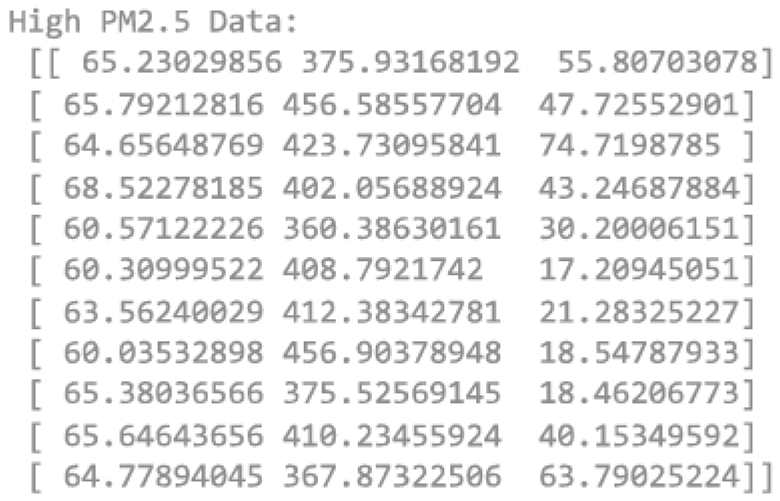
```
Normalized Data:
 [[0.69687903 0.10851626 0.50733793]
 [0.55488996 0.32294571 0.53595327]
 [0.73063878 0.33974498 0.60957419]
 [0.92637598 0.24067866 0.6054511 ]
 [0.53344797 0.37885495 0.26270063]]
```

Filtered Data

high_pm25 = air_pollution_data[air_pollution_data[:, 0] > 60]

print("High PM2.5 Data:\n", high_pm25)

O/P

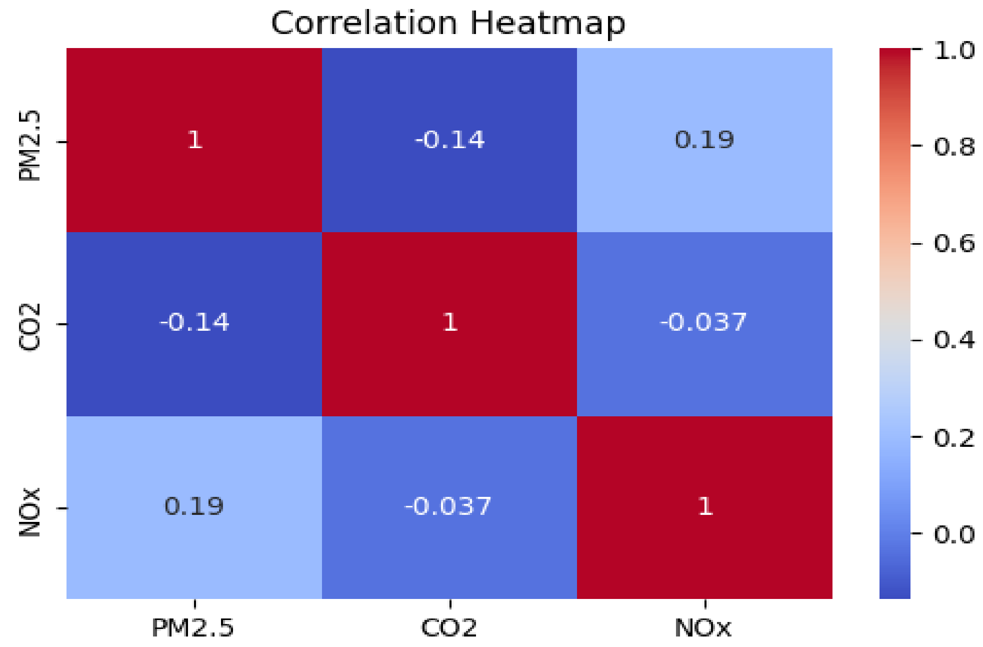


```
High PM2.5 Data:
 [[ 65.23029856 375.93168192  55.80703078]
 [ 65.79212816 456.58557704  47.72552901]
 [ 64.65648769 423.73095841  74.7198785 ]
 [ 68.52278185 402.05688924  43.24687884]
 [ 60.57122226 360.38630161  30.20006151]
 [ 60.30999522 408.7921742   17.20945051]
 [ 63.56240029 412.38342781  21.28325227]
 [ 60.03532898 456.90378948  18.54787933]
 [ 65.38036566 375.52569145  18.46206773]
 [ 65.64643656 410.23455924  40.15349592]
 [ 64.77894045 367.87322506  63.79025224]]
```

Correlation Matrix

```
correlation_matrix = np.corrcoef(air_pollution_data.T)


plt.figure(figsize=(6, 4))

sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm", xticklabels=columns, yticklabels=columns)

plt.title("Correlation Heatmap")

plt.show()
```

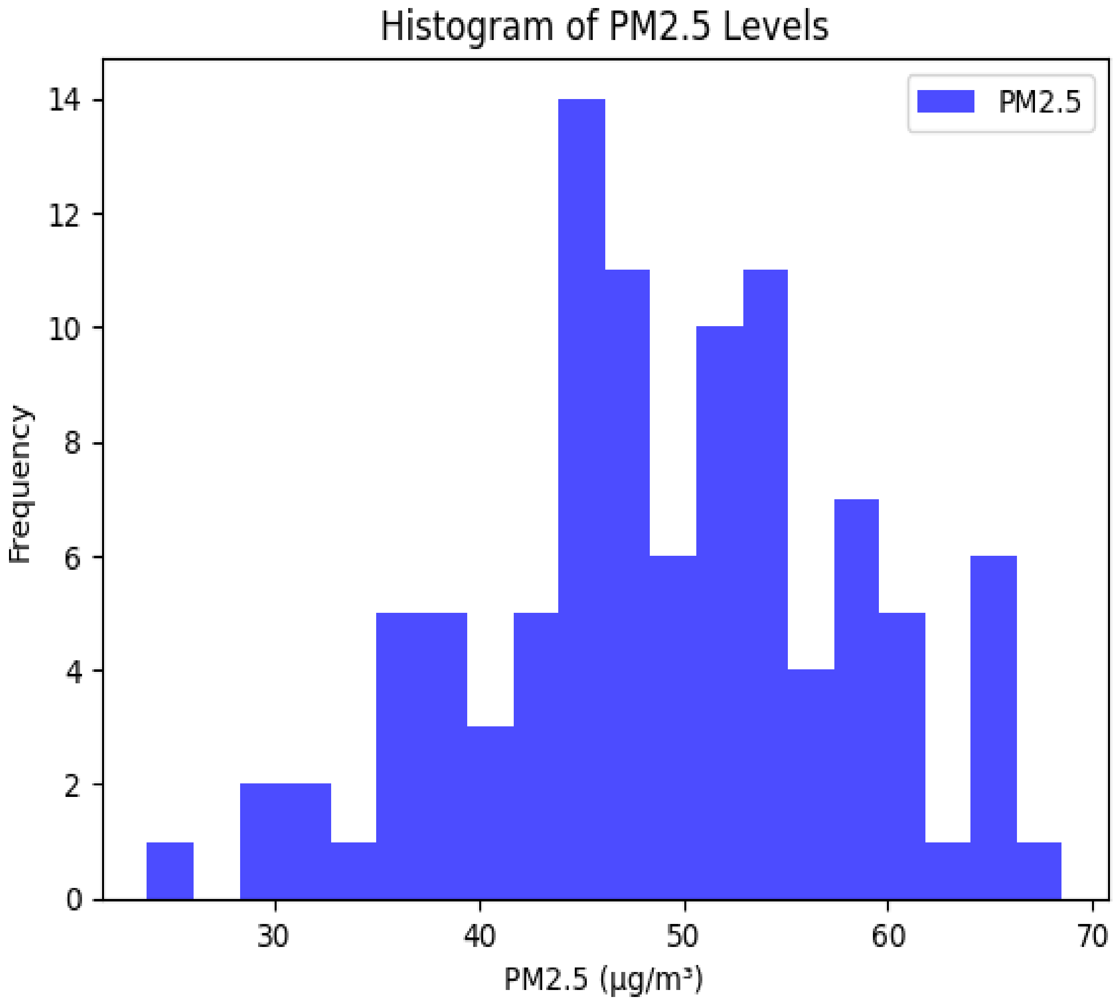
O/P



Histogram of PM 2.5

```
plt.hist(pm25, bins=20, alpha=0.7, color='b', label="PM2.5")

plt.xlabel("PM2.5 (µg/m³)")

plt.ylabel("Frequency")

plt.title("Histogram of PM2.5 Levels")

plt.legend()

plt.show()
```

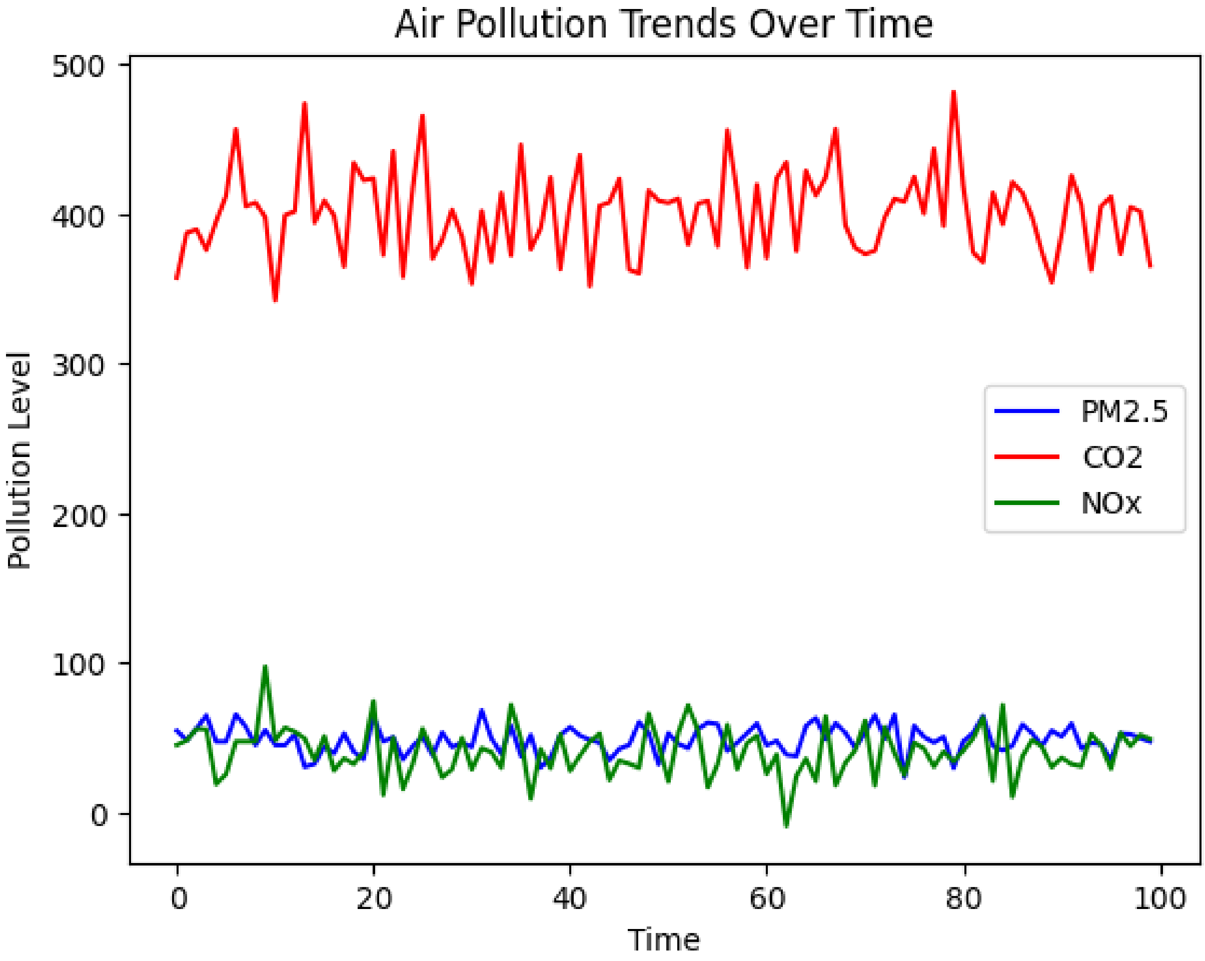
O/P



Line Chart for Pollution Trends

```
plt.plot(range(num_samples), pm2.5, label="PM2.5", color="b")

plt.plot(range(num_samples), co2, label="CO2", color="r")

plt.plot(range(num_samples), nox, label="NOx", color="g")

plt.xlabel("Time")

plt.ylabel("Pollution Level")

plt.title("Air Pollution Trends Over Time")

plt.legend()

plt.show()
```

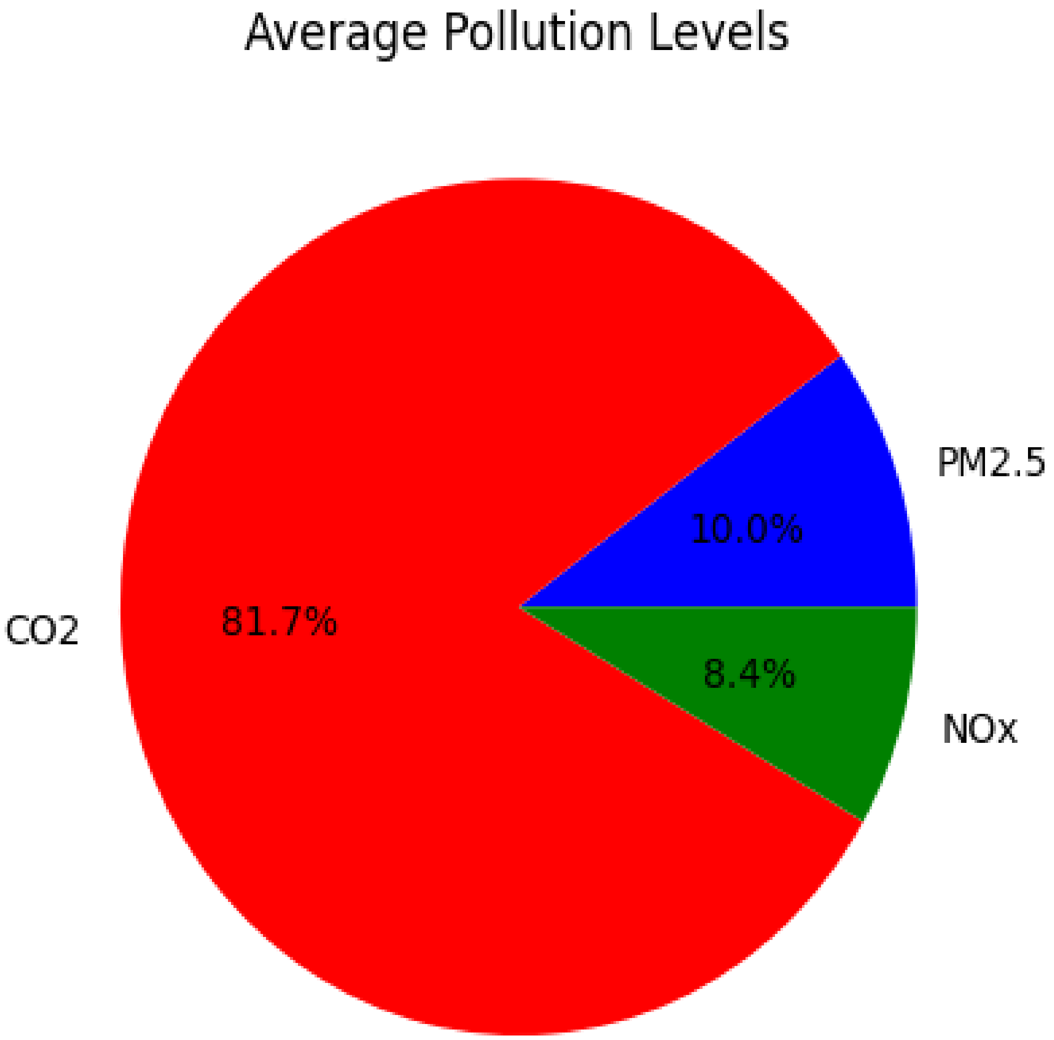
O/P



Pie Chart for average pollution level

```
plt.pie(mean_values, labels=columns, autopct='%1.1f%%', colors=['blue', 'red', 'green'])

plt.title("Average Pollution Levels")

plt.show()
```

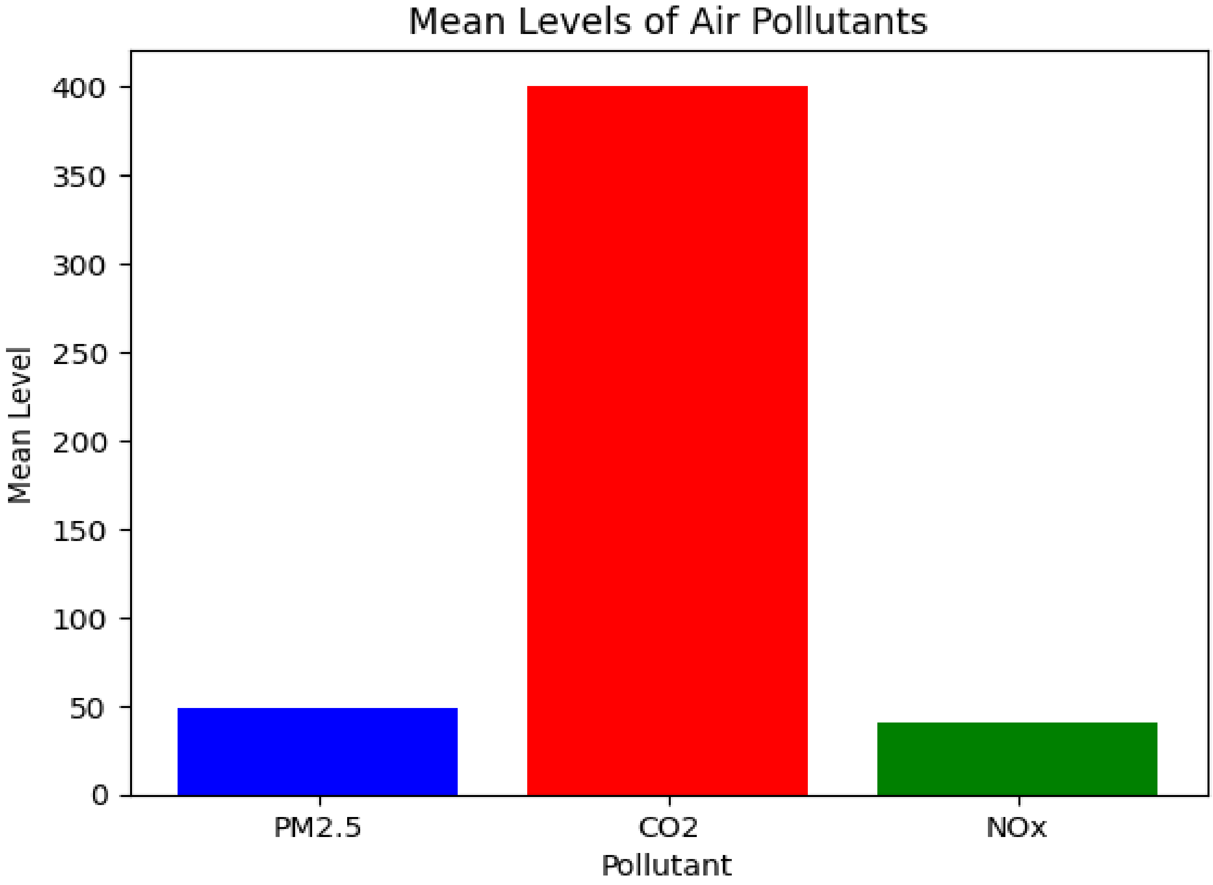
O/P



Bar Chart of Pollution

```
plt.bar(columns, mean_values, color=['blue', 'red', 'green'])

plt.xlabel("Pollutant")

plt.ylabel("Mean Level")

plt.title("Mean Levels of Air Pollutants")

plt.show()
```

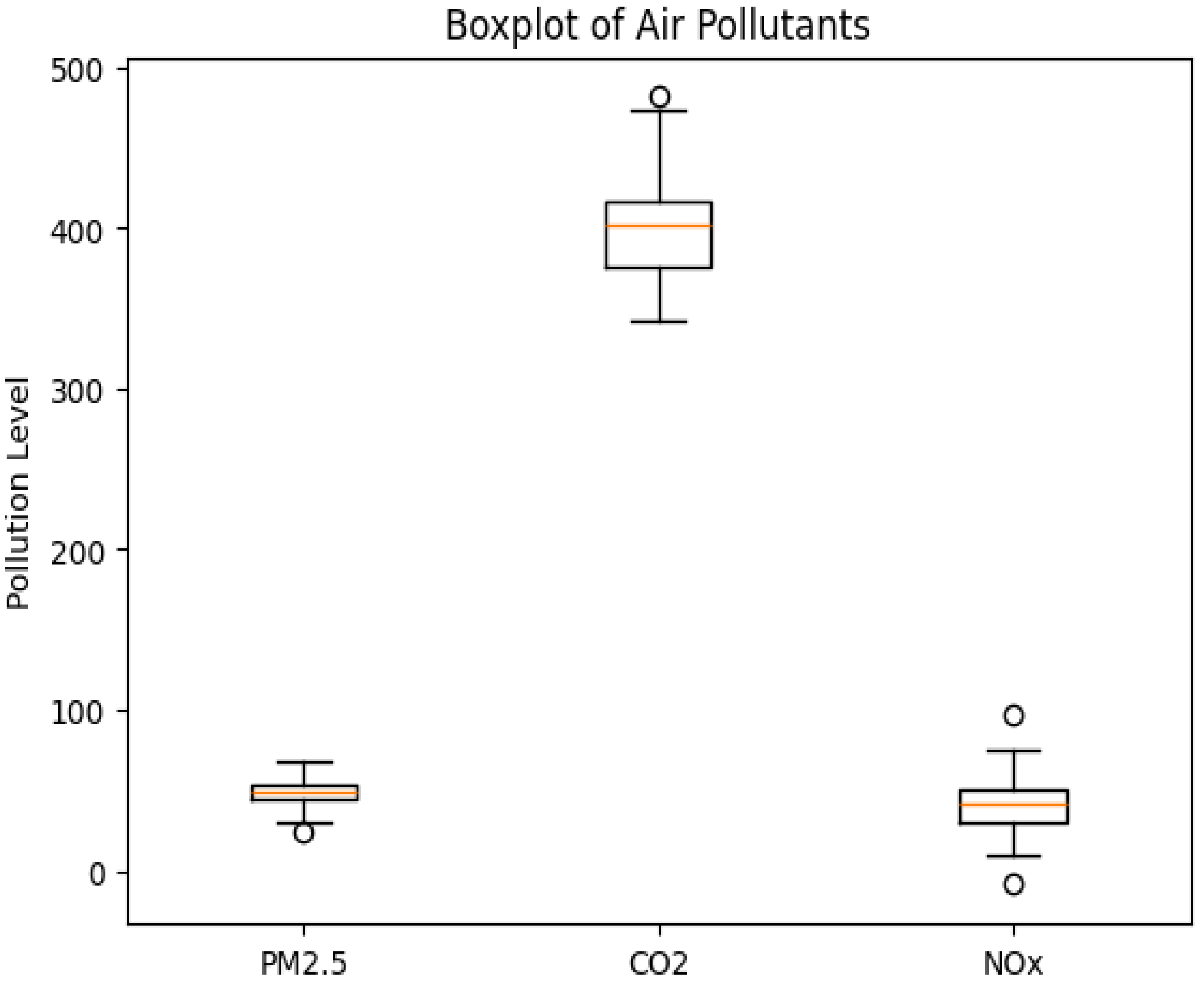
O/P



Boxplot for outlier Detection

```
plt.boxplot([pm25, co2, nox], labels=columns)

plt.title("Boxplot of Air Pollutants")

plt.ylabel("Pollution Level")

plt.show()
```

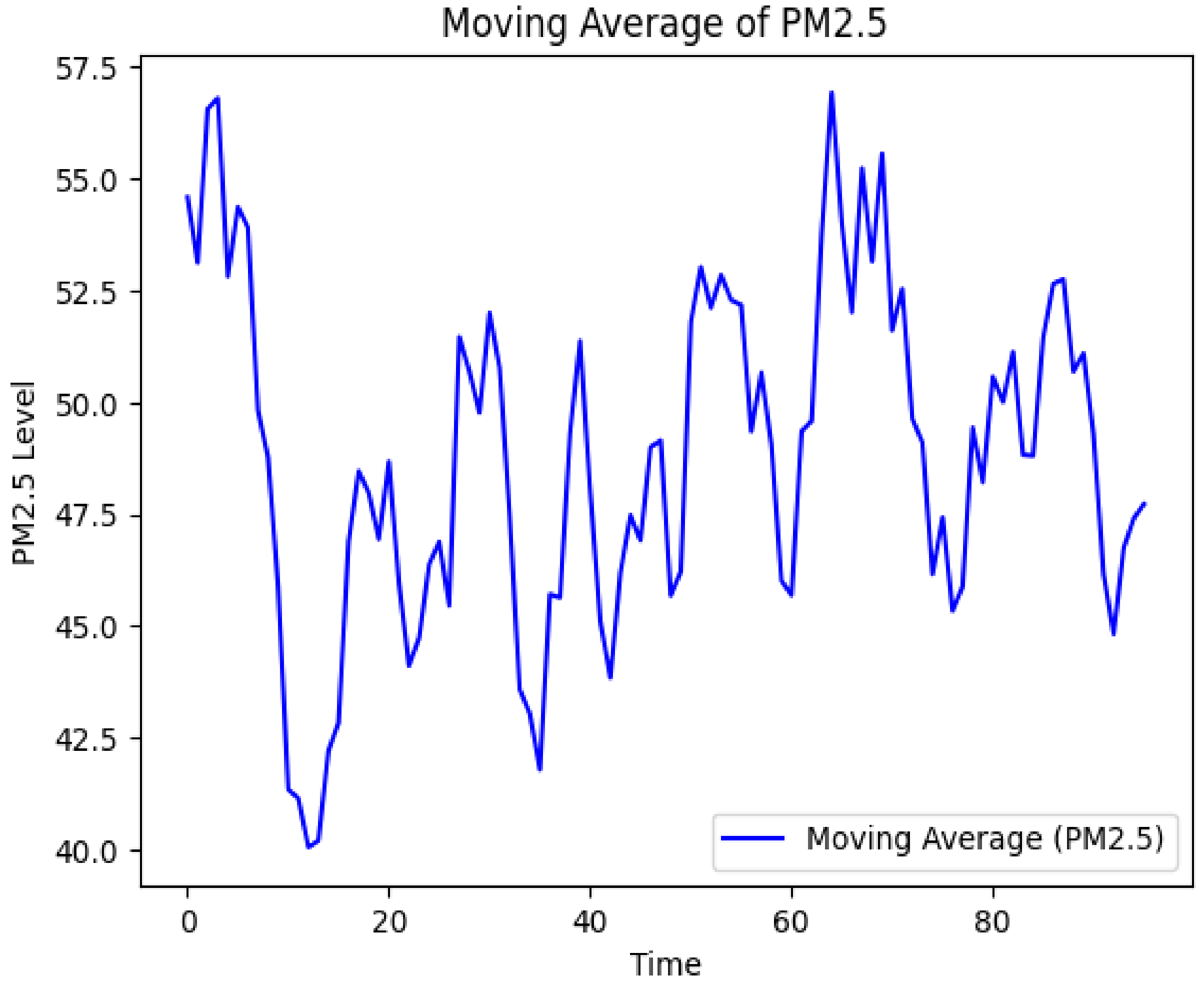
O/P



Moving Average of PM 2.5

```
window = 5

pm25_moving_avg = np.convolve(pm25, np.ones(window)/window, mode='valid')


plt.plot(range(len(pm25_moving_avg)), pm25_moving_avg, label="Moving Average (PM2.5)", color='b')

plt.xlabel("Time")

plt.ylabel("PM2.5 Level")

plt.title("Moving Average of PM2.5")

plt.legend()

plt.show()
```

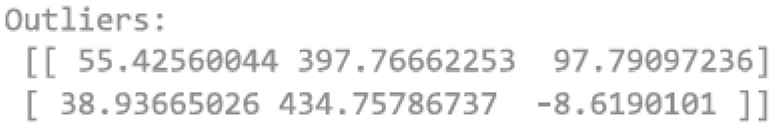
O/P



Detection using Z score

```
z_scores = (air_pollution_data - mean_values) / std_values

outliers = air_pollution_data[(np.abs(z_scores) > 3).any(axis=1)]

print("Outliers:\n", outliers)
```

O/P

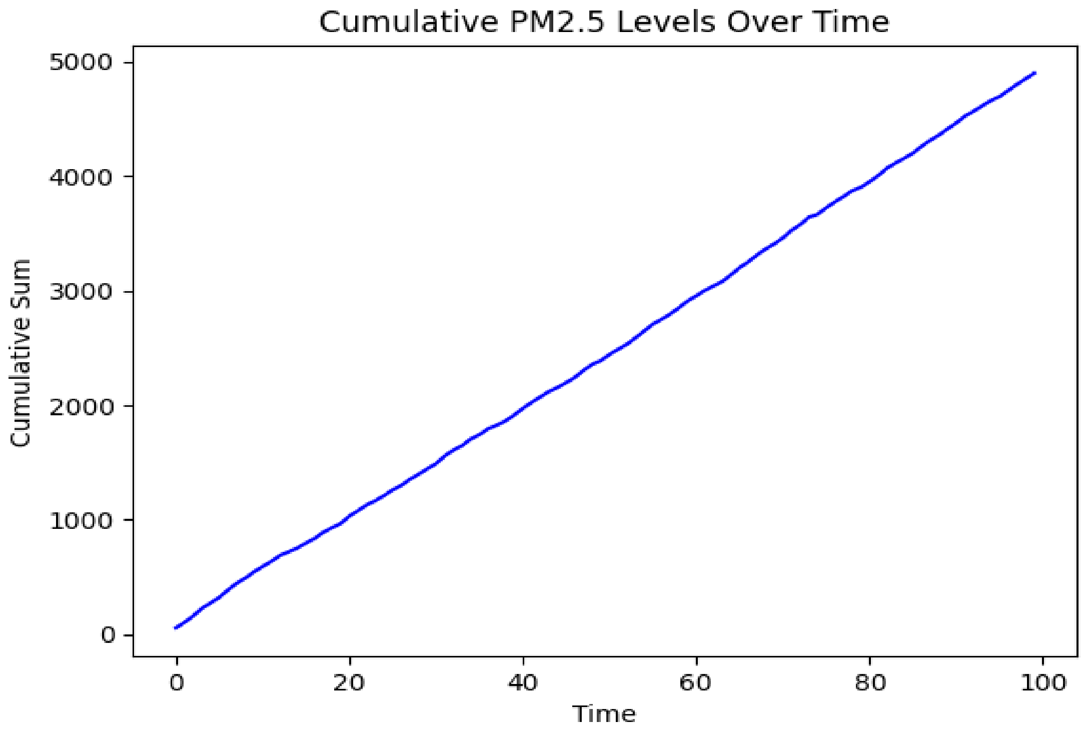


```
Outliers:
 [[ 55.42560044 397.76662253  97.79097236]
 [ 38.93665026 434.75786737  -8.6190101 ]]
```

Cumulative Sum

```
cumsum_pm25 = np.cumsum(pm25)


plt.plot(range(num_samples), cumsum_pm25, color='b')

plt.xlabel("Time")

plt.ylabel("Cumulative Sum")

plt.title("Cumulative PM2.5 Levels Over Time")

plt.show()
```

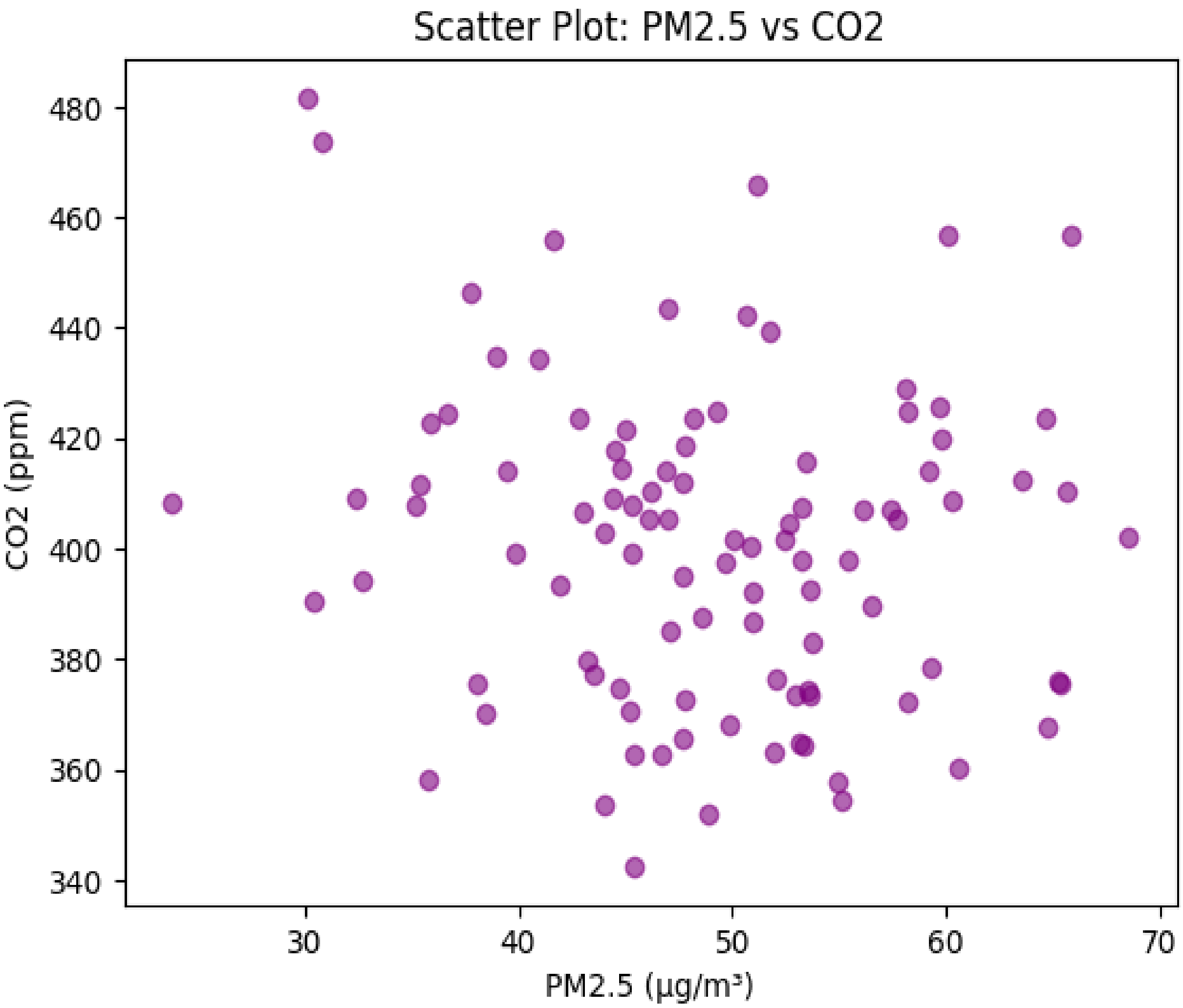
O/P



Scattered Plot PM vs CO2

```
plt.scatter(pm25, co2, color='purple', alpha=0.6)

plt.xlabel("PM2.5 (µg/m³)")

plt.ylabel("CO2 (ppm)")

plt.title("Scatter Plot: PM2.5 vs CO2")

plt.show()
```

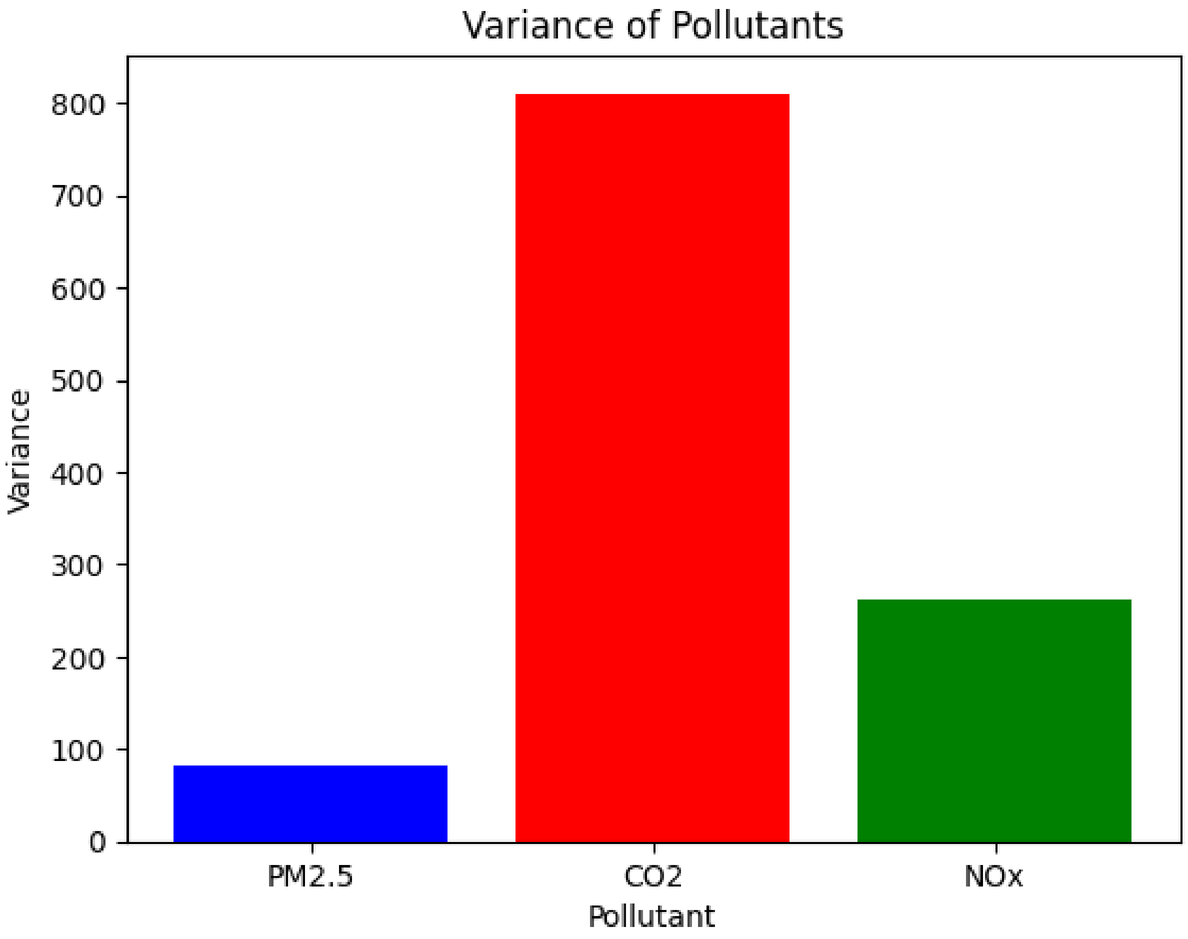
O/P



Variance of Pollution

```
variance_values = np.var(air_pollution_data, axis=0)

print("Variance:", variance_values)

plt.bar(columns, variance_values, color=['blue', 'red', 'green'])

plt.xlabel("Pollutant")

plt.ylabel("Variance")

plt.title("Variance of Pollutants")

plt.show()
```

O/P



CONCLUSION:

This study explored the impact of air pollution by analyzing datasets containing **PM2.5, CO2, and NOx levels** using **data science techniques**. By leveraging Python-based **data processing, statistical analysis, and visualization methods**, key trends and relationships among pollutants were identified. Techniques such as **histograms, scatter plots, correlation heatmaps, and time-series analysis** provided valuable insights into pollution levels and their fluctuations over time.

Findings indicate that **PM2.5, CO2, and NOx levels exhibit strong correlations with urban activities such as traffic density and industrial emissions**. Seasonal variations and meteorological factors such as wind speed and humidity significantly impact pollution dispersion. **High pollution levels in urban areas pose serious environmental and health risks, necessitating proactive monitoring and intervention strategies.**

To improve air quality, **real-time monitoring systems, predictive analytics, and data-driven policymaking** should be implemented. The integration of **IoT sensors and machine learning models** could enhance predictive capabilities, enabling better pollution control measures. Urban planners and environmental agencies can use these insights to design **smarter cities with sustainable policies** aimed at reducing pollution sources and mitigating environmental impacts.

While this study provides valuable insights, it is **limited by the availability and scope of collected datasets**. Future research should incorporate **larger, real-time datasets and advanced machine learning algorithms** for more precise air quality predictions. Additionally, the development of **interactive dashboards** can provide real-time updates on pollution levels, helping both citizens and policymakers make informed decisions.

This study highlights the **importance of data-driven environmental monitoring**. By utilizing **data science techniques**, researchers, policymakers, and urban planners can gain deeper insights into air pollution trends, implement effective mitigation strategies, and work towards creating **healthier, sustainable urban environments for future generations**.