<DATA SCIENCE TOOLBOX: PYTHON PROGRAMMING>

PROJECT REPORT

(Project Semester January-April 2025)

(Exploratory Data Analysis on Air Pollution in Uttar Pradesh)

Submitted by

(Vikas Gupta)

Registration :- 12316618

Programme and Section :- CSE K23GR

Course Code :- INT375

Under the Guidance of

(Ms. Gargi Sharma)

Discipline of CSE/IT

Lovely School of Computer Science and Engineering

Lovely Professional University, Phagwara

CERTIFICATE

This is to certify that Vikas Gupta bearing Registration no. 12316618 has completed INT375 project titled, "Exploratory data analysis on Air Pollution Analysis in Uttar Pradesh" under my guidance and supervision. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature and Name of the Supervisor

Designation of the Supervisor

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab.

Date: 12th April, 2025

DECLARATION

I, Vikas Gupta, student of Computer Science and Engineering under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 12th April, 2025

Registration No. 12316618 **Name of the student :- Vikas Gupta**

Air Pollution Analysis in Uttar Pradesh

- **Name – Vikas Gupta**
- **Reg – no – 12316618**
- **Roll-no- 05**
- **Section – K23GR**
- **In This project I have covered almost every point of python libraries including NumPy pandas mat plot and seaborn**
- **The Website from which I have taken this dataset is -- https://data.gov**

➢ **Objective:** The goal of this project is to analyze the air pollution levels in various parts of Uttar Pradesh using a cleaned dataset. The focus is on understanding the distribution, trends, and correlations among the primary air pollutants: PM10 (RSPM), SO2, and NO2.

Data Preprocessing:

- Loaded a CSV dataset containing pollution readings.
- Converted the 'Sampling Date' column to datetime format.
- Removed the 'SPM' column as it contained only missing values.
- Handled missing values in numeric columns by replacing them with their respective column means.

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

# Load dataset
file_path = r"C:\All subject\INT375DATA SCIENCE TOOLBOX PYTHON PROGRAMMING\ppp\Cleaned_AIR_Pollution_UP.csv"
df = pd.read_csv(file_path)

# Convert 'Sampling Date' to datetime format
if 'Sampling Date' in df.columns:
    df['Sampling Date'] = pd.to_datetime(df['Sampling Date'], errors='coerce')
else:
    print("Warning: 'Sampling Date' column not found.")
```

```
•
•   # Drop 'SPM' column if it exists and has all missing values
•   if 'SPM' in df.columns and df['SPM'].isna().all():
•       df.drop(columns=['SPM'], inplace=True)
•
•   # Display dataset structure and first few rows
•   print("\nDataset Info:")
•   print(df.info())
•   print("\nFirst 5 Rows:")
•   print(df.head())
•
•   # Check and fill missing values (numerical columns)
•   print("\nMissing Values:")
•   print(df.isnull().sum())
•
•   numeric_cols = df.select_dtypes(include='number').columns
•   df[numeric_cols] = df[numeric_cols].fillna(df[numeric_cols].mean())
•
•   # Summary statistics
•   print("\nSummary Statistics:")
•   print(df.describe())
```

```
Dataset Info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3430 entries, 0 to 3429
Data columns (total 10 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   Stn Code                        3430 non-null   int64
 1   Sampling Date                   1320 non-null   datetime64[ns]
 2   State                           3430 non-null   object
 3   City/Town/Village/Area          3430 non-null   object
 4   Location of Monitoring Station  3430 non-null   object
 5   Agency                          3430 non-null   object
 6   Type of Location                3430 non-null   object
 7   SO2                             3430 non-null   float64
 8   NO2                             3430 non-null   float64
 9   RSPM/PM10                       3430 non-null   float64
dtypes: datetime64[ns](1), float64(3), int64(1), object(5)
memory usage: 268.1+ KB
None
```

```
First 5 Rows:
   Stn Code Sampling Date         State City/Town/Village/Area  ...                 Type of Location  SO2  NO2  RSPM/PM10
0       555    2012-01-03  Uttar Pradesh            Allahabad  ...  Residential, Rural and other Areas  3.0 21.0      283.0
1       555    2012-01-05  Uttar Pradesh            Allahabad  ...  Residential, Rural and other Areas  3.0 21.0      358.0
2       555    2012-01-09  Uttar Pradesh            Allahabad  ...  Residential, Rural and other Areas  3.0 18.0      265.0
3       555    2012-01-11  Uttar Pradesh            Allahabad  ...  Residential, Rural and other Areas  3.0 16.0      272.0
4       555           NaT  Uttar Pradesh            Allahabad  ...  Residential, Rural and other Areas  3.0 22.0      384.0

[5 rows x 10 columns]
```

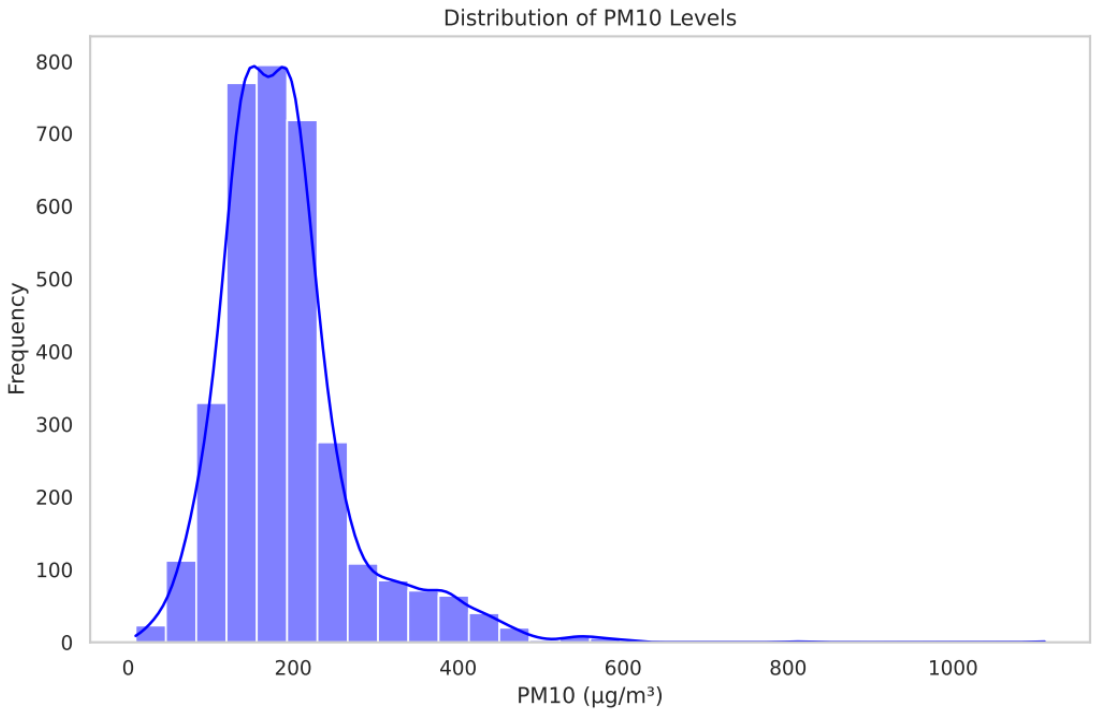
```
Missing Values:
Stn Code                          0
Sampling Date                  2110
State                             0
City/Town/Village/Area            0
Location of Monitoring Station    0
Agency                            0
Type of Location                  0
SO2                               0
NO2                               0
RSPM/PM10                         0
dtype: int64

Summary Statistics:
          Stn Code                  Sampling Date          SO2          NO2     RSPM/PM10
count  3430.000000                           1320  3430.000000  3430.000000  3430.000000
mean    462.617201  2012-06-23 11:27:16.363636224    13.089611    29.988568   190.192336
min       6.000000            2012-01-01 00:00:00     1.000000     2.000000     9.000000
25%     258.000000            2012-04-03 00:00:00     7.000000    23.000000   138.000000
50%     535.000000            2012-06-12 00:00:00    10.000000    30.000000   179.000000
75%     718.000000            2012-10-01 00:00:00    18.000000    35.000000   218.000000
max     730.000000            2012-12-12 00:00:00   183.000000   592.000000  1111.000000
std     245.646316                            NaN     9.749688    13.894226    81.869530
```

Exploratory Data Analysis & Visualizations:

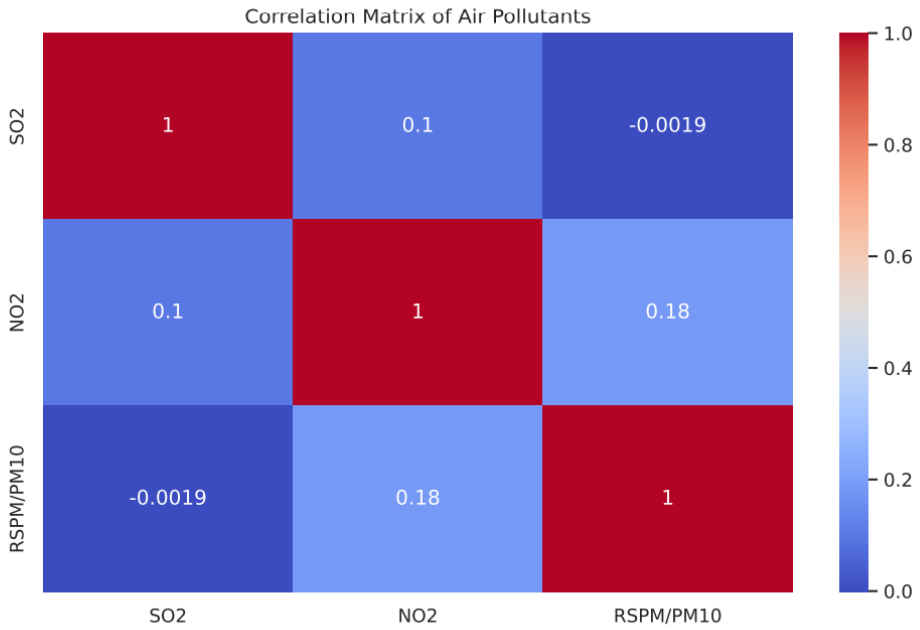
1. **Distribution of PM10:**
   - A histogram with KDE showed the spread of PM10 levels.
   - PM10 levels were highly variable, indicating varying air quality across locations.

```
1. Histogram of PM10 (RSPM) levels
 plt.figure(figsize=(10, 6))
 sns.histplot(df['RSPM/PM10'].dropna(), bins=30, kde=True,color='blue')
 plt.title('Distribution of PM10 Levels')
 plt.xlabel('PM10 (µg/m³)')
 plt.ylabel('Frequency')
 plt.grid()
 plt.show()
```



2. **Correlation Matrix:**
   - A heatmap displayed strong correlations among SO2, NO2, and PM10.
   - Positive correlations suggest that pollutant levels tend to rise together.

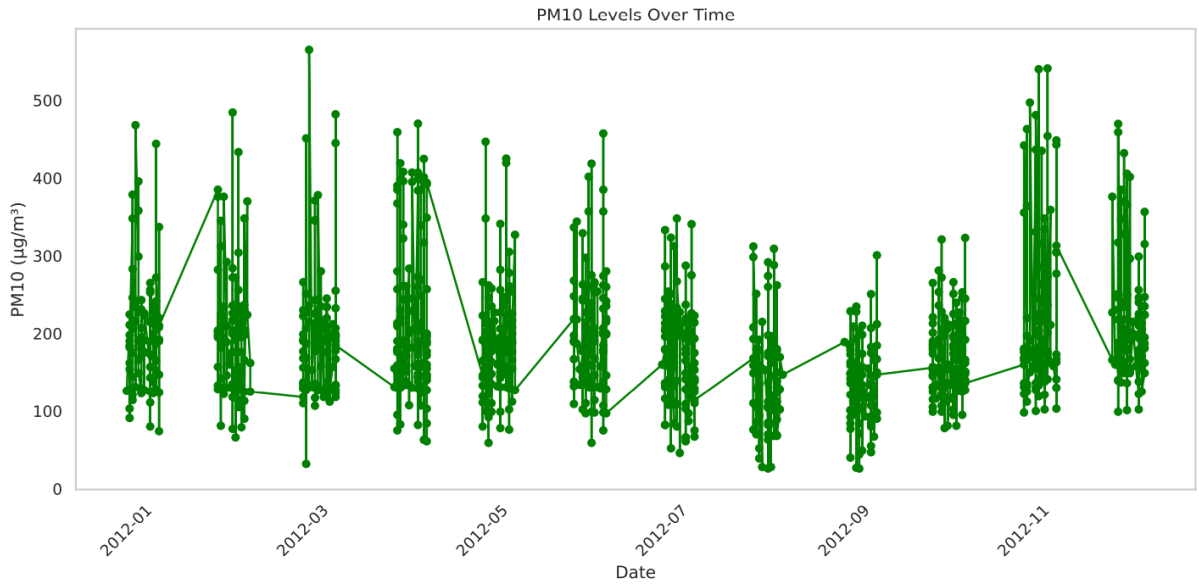
```
 2. Correlation heatmap of pollutants
plt.figure(figsize=(10, 6))
corr = df[['SO2', 'NO2', 'RSPM/PM10']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix of Air Pollutants')  # Shows relationship strength between SO2, NO2, PM10
plt.show()
```



3. PM10 Over Time:

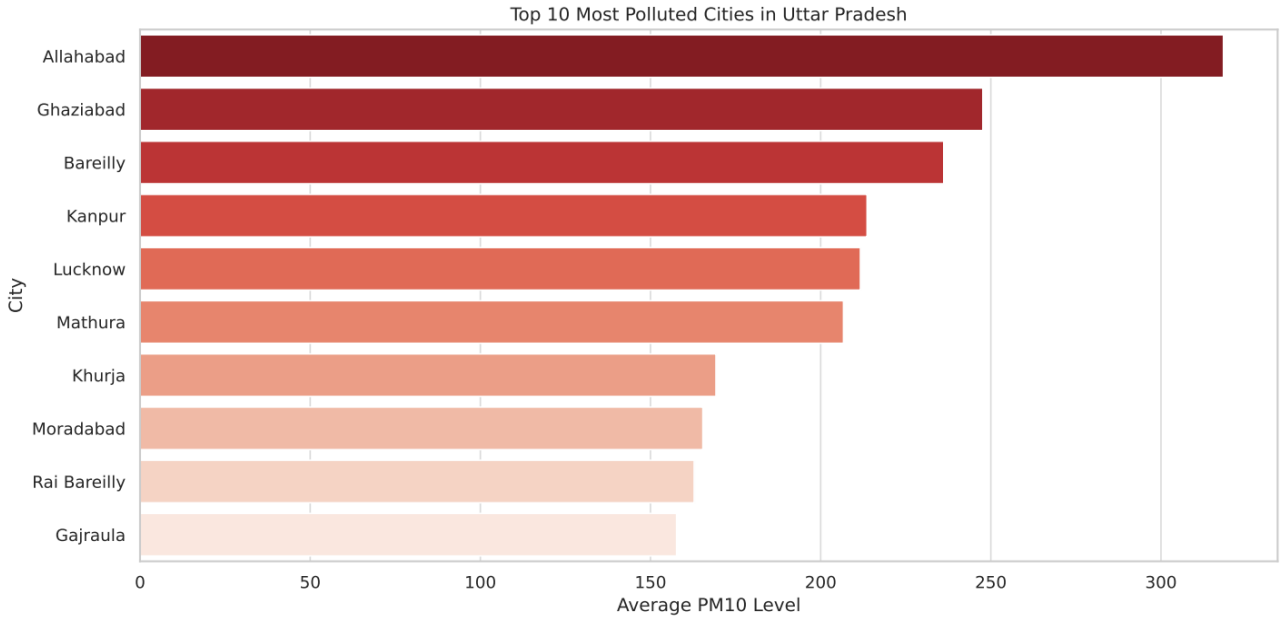
- A time series line plot highlighted seasonal fluctuations in PM10.
- Some months showed clear spikes in pollution levels, possibly due to winter and festivities.

```
 3. PM10 over time (line plot)
  plt.figure(figsize=(12, 6))
  df_sorted = df.sort_values('Sampling Date')
  plt.plot(df_sorted['Sampling Date'], df_sorted['RSPM/PM10'],
marker='o', linestyle='-', color='green')
  plt.title('PM10 Levels Over Time')
  plt.xlabel('Date')
  plt.ylabel('PM10 (µg/m³)')
  plt.xticks(rotation=45)
  plt.grid()
  plt.tight_layout()
  plt.show()
```



4. **Top 10 Most Polluted Cities:**
    - Bar plot based on average PM10 values revealed the most affected cities.
    - Major urban areas with dense populations and traffic topped the list.

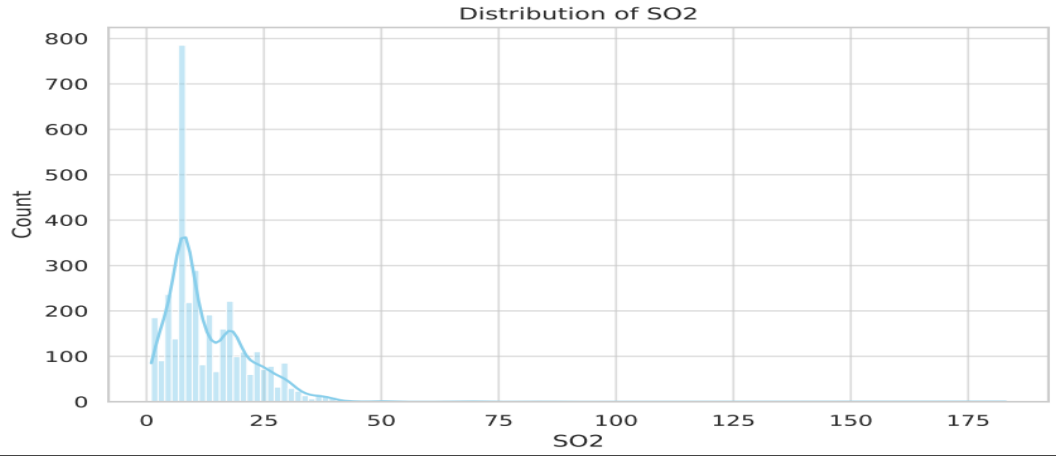
```
Top 10 most polluted cities by average PM10
  city_pollution =
df.groupby('City/Town/Village/Area')['RSPM/PM10'].mean().sort_values(as
cending=False).head(10)
  plt.figure(figsize=(12, 6))
  sns.barplot(x=city_pollution.values, y=city_pollution.index,
palette='Reds_r')
  plt.xlabel('Average PM10 Level')
  plt.ylabel('City')
  plt.title('Top 10 Most Polluted Cities in Uttar Pradesh (by PM10)')
  plt.tight_layout()
  plt.show()
```

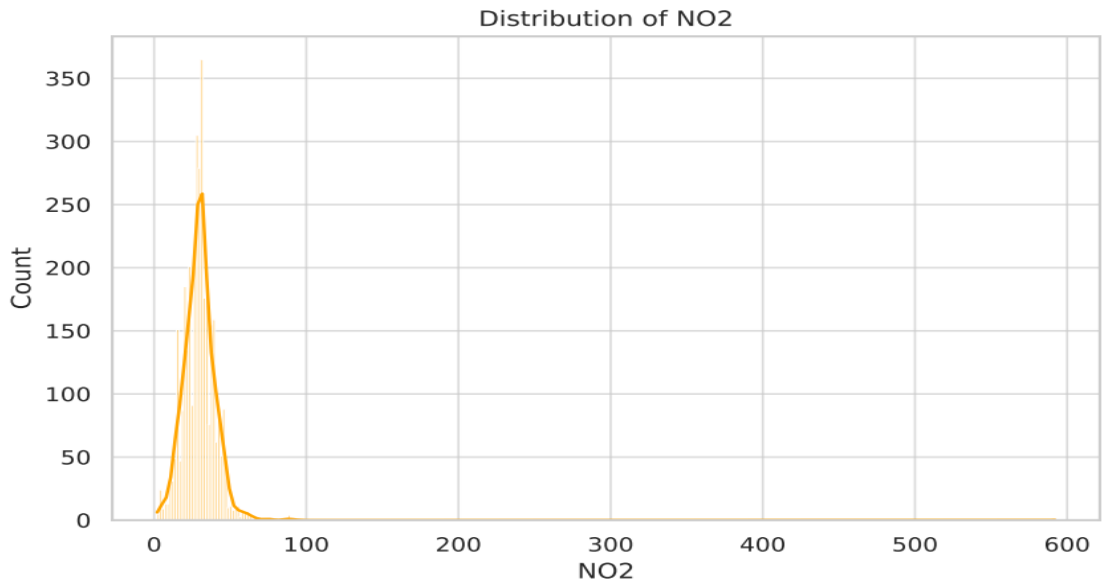


5. **Distribution of SO2 and NO2:**

- Separate histograms with KDE for each pollutant.
- Both pollutants showed normal distributions with some outliers

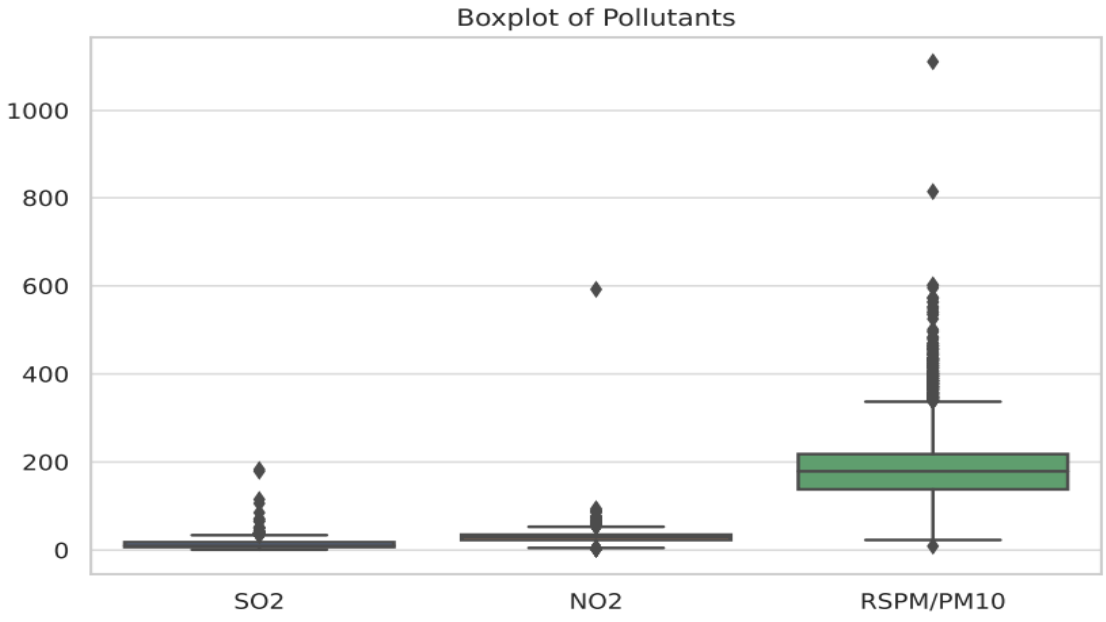
```
- Histogram of SO2
- plt.figure(figsize=(8, 5))
- sns.histplot(df['SO2'].dropna(), kde=True, color='skyblue')
- plt.title('Distribution of SO2')  # Spread of SO2 concentrations
- plt.show()
-  Histogram of NO2
- plt.figure(figsize=(8, 5))
- sns.histplot(df['NO2'].dropna(), kde=True, color='orange')
- plt.title('Distribution of NO2')  # Spread of NO2 concentrations
- plt.show()
```





6. **Boxplot of Pollutants:**
   - Boxplots provided insight into the spread and outliers for SO2, NO2, and PM10.
   - PM10 had the largest spread and number of outliers.

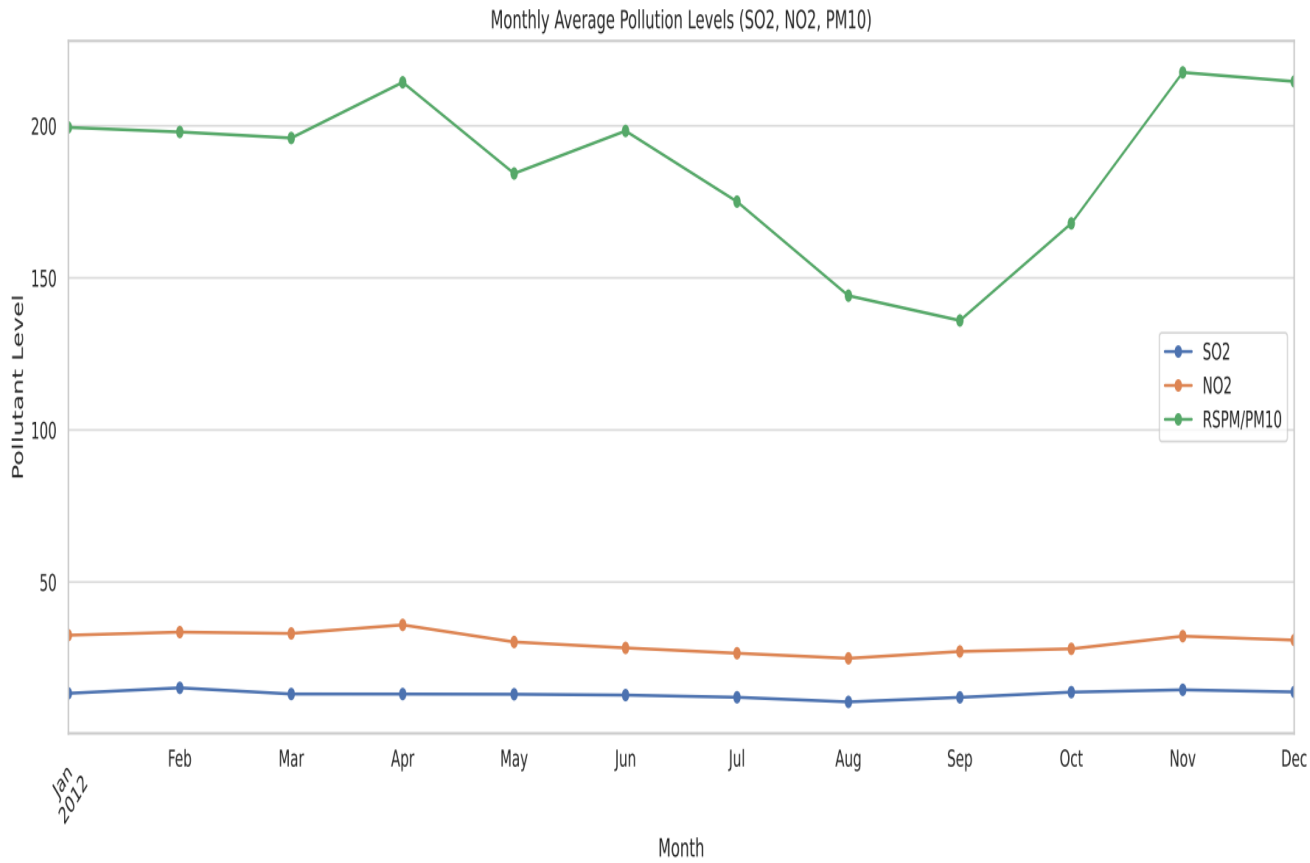
```
Boxplot for pollutants
plt.figure(figsize=(8, 5))
sns.boxplot(data=df[['SO2', 'NO2', 'RSPM/PM10']])
plt.title('Boxplot of Pollutants')
plt.show()
```



7. **Monthly Average Pollution Levels:**
   - Line plots of SO2, NO2, and PM10 averaged by month.
   - Showed cyclic patterns and possible improvement in recent months.

```
# 8. Monthly average pollution levels (SO2, NO2, PM10)
df_time = df.dropna(subset=['Sampling Date'])
df_time['Month'] = df_time['Sampling Date'].dt.to_period('M')
monthly_avg = df_time.groupby('Month')[['SO2', 'NO2',
'RSPM/PM10']].mean()

plt.figure(figsize=(15, 6))
monthly_avg.plot(marker='o')
plt.title('Monthly Average Pollution Levels (SO2, NO2, PM10)')  #
Trends by month
plt.xlabel('Month')
plt.ylabel('Pollutant Level')
plt.grid(True)
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



Conclusions:

- PM10 is the most concerning pollutant in terms of both distribution and concentration.
- Pollution levels tend to peak in specific months, indicating seasonality.
- Urban centers are significantly more polluted, requiring targeted interventions.
- The correlation among pollutants hints at common sources such as vehicular emissions and industrial activity.