

IBM NAAN MUDHALVAN - PHASE 5
DOMAIN – DATA ANALYTICS WITH COGNOS
Air Quality Analysis in Tamil Nadu

Project Title	Air Quality Analysis in Tamil Nadu
Name	DEEPIKA V
Team Members	DEEPIKA V AMRASH BHANU A FOWZIA K HARINI K
Reg No	420421104014

Introduction:

In Tamil Nadu, the "Air Quality Analysis" project seeks to analyze and visualize data from air quality monitoring stations. The primary goal is to uncover air pollution trends, pinpoint pollution hotspots, and construct a predictive model for RSPM/PM10 estimation based on SO2 and NO2 levels. By leveraging Python and pertinent libraries, this project will contribute to informed decision-making and environmental well-being in the region.

Project Objectives:

- **Data Analysis:** Analyze air quality data from Tamil Nadu monitoring stations to identify historical and current air pollution trends.
- **Hotspot Identification:** Pinpoint regions with consistently high pollution levels, helping authorities target pollution control measures effectively.
- **Predictive Model:** Develop a predictive model using Python and relevant libraries to estimate RSPM/PM10 levels based on SO2 and NO2 concentrations, aiding in forecasting air quality.

Analysis Approach:

1. Data Collection and Preprocessing:

- Collect and preprocess air quality data using Python (Pandas) to ensure data quality and consistency.

2. Exploratory Data Analysis (EDA):

- Utilize Python libraries (Matplotlib, Seaborn) to conduct EDA, generating visual insights like time series plots and correlation matrices.

3. High Pollution Area Identification:

- Apply geospatial visualization tools (e.g., Folium) in Python to identify regions with persistent high pollution levels.

4. Predictive Modeling:

- Develop a Python-based predictive model (Scikit-Learn) to estimate RSPM/PM10 levels based on SO2 and NO2 concentrations.

Visualization Techniques:

- **Time Series Plots:** Visualize pollutant trends over time to identify seasonality and trends (Matplotlib).
- **Heatmaps:** Display correlation between pollutants for insights into relationships (Seaborn).
- **Geospatial Maps:** Use geographical maps with color-coding to visualize pollution levels across regions (Folium).

Air pollution trends and pollution levels in Tamil Nadu:

The analysis outlined in the project provides valuable insights into air pollution trends and pollution levels in Tamil Nadu by utilizing various data-driven techniques and methodologies. Here's how the analysis offers these insights:

1. **Trend Analysis:** The time series analysis allows for the identification of long-term air pollution trends. By decomposing the time series data, you can distinguish whether pollution levels are rising, declining, or remaining relatively stable over time. This information provides a fundamental understanding of how air quality has evolved in Tamil Nadu, helping stakeholders make informed decisions and policies for air quality management.
2. **Seasonal Variations:** Seasonal decomposition reveals recurring patterns in air pollution data. Understanding these seasonal variations is critical for recognizing specific periods of the year when air quality is more likely to deteriorate. This information enables preparedness and proactive measures to address season-specific pollution challenges.
3. **Anomaly Detection:** By analyzing the residual component, you can identify unusual events or outliers in the pollution data. These anomalies may represent unexpected pollution incidents or unique circumstances. Investigating these outliers can help in understanding their causes and taking corrective actions to prevent similar events in the future.
4. **Geospatial Insights:** Utilizing geospatial visualization techniques, the project helps in mapping and plotting air quality data across different monitoring stations or geographical areas in Tamil Nadu. This spatial analysis provides information on the distribution of pollution levels, helping to pinpoint areas with consistently high pollution and assess regional disparities in air quality.
5. **Correlation Analysis:** The correlation analysis between different pollutants (SO2, NO2, RSPM/PM10) elucidates the relationships between these variables. This is essential for identifying potential pollution sources and understanding how different pollutants interact to contribute to overall air quality. It helps in focusing pollution control efforts on the most influential factors.

6. **Predictive Modeling:** The development of a predictive model to estimate RSPM/PM10 levels based on SO2 and NO2 levels enables forecasting future air quality conditions. This model offers a proactive approach to air quality management by allowing authorities to anticipate pollution levels and take timely preventive measures.

In summary, the analysis provides a holistic view of air pollution in Tamil Nadu. It facilitates the recognition of historical trends, seasonally influenced variations, spatial disparities, pollutant interdependencies, and predictive capabilities. These insights empower decision-makers, environmental agencies, and the public to:

- Address long-term air quality challenges with data-backed strategies.
- Prepare and respond effectively to seasonal variations in air quality.
- Investigate and rectify abnormal pollution events.
- Identify and mitigate pollution sources.
- Make informed predictions for future air quality conditions.

Instruction of load the dataset, perform calculations, and create visualizations using Python:

To replicate the air quality analysis, load the dataset, perform calculations, and create visualizations using Python, you can follow these step-by-step instructions:

Step 1: Setting Up Your Python Environment

Before starting, ensure you have Python and necessary libraries installed. You can use libraries like pandas, matplotlib, seaborn, folium, and statsmodels for the analysis. If you haven't already installed these libraries, you can use pip to do so. Open a terminal or command prompt and run the following commands:

```
pip install pandas matplotlib seaborn folium statsmodels
```

Step 2: Load the Dataset

Download the air quality dataset (cpcb_dly_aq_tamil_nadu-2014.csv) to your working directory or specify the full path to the dataset in code.

Dataset Link: <https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014>

```
import pandas as pd
# Load the dataset
data = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')
```

Step 3: Data Preprocessing

Perform data preprocessing to handle missing values and format the date column.

```
import pandas as pd

# Load the dataset
data = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')

# Filter the columns you need for analysis
data = data[['Sampling Date', 'RSPM/PM10', 'SO2', 'NO2']]

# Drop rows with missing values in any of the selected columns
data.dropna(subset=['Sampling Date', 'RSPM/PM10', 'SO2', 'NO2'], inplace=True)

# Convert 'Sampling Date' to datetime with the correct format (two-digit year 'yy')
data['Sampling Date'] = pd.to_datetime(data['Sampling Date'], format='%d-%m-%y')

# Set 'Sampling Date' as the index
data.set_index('Sampling Date', inplace=True)
```

Step 4: Exploratory Data Analysis (EDA) and Visualization

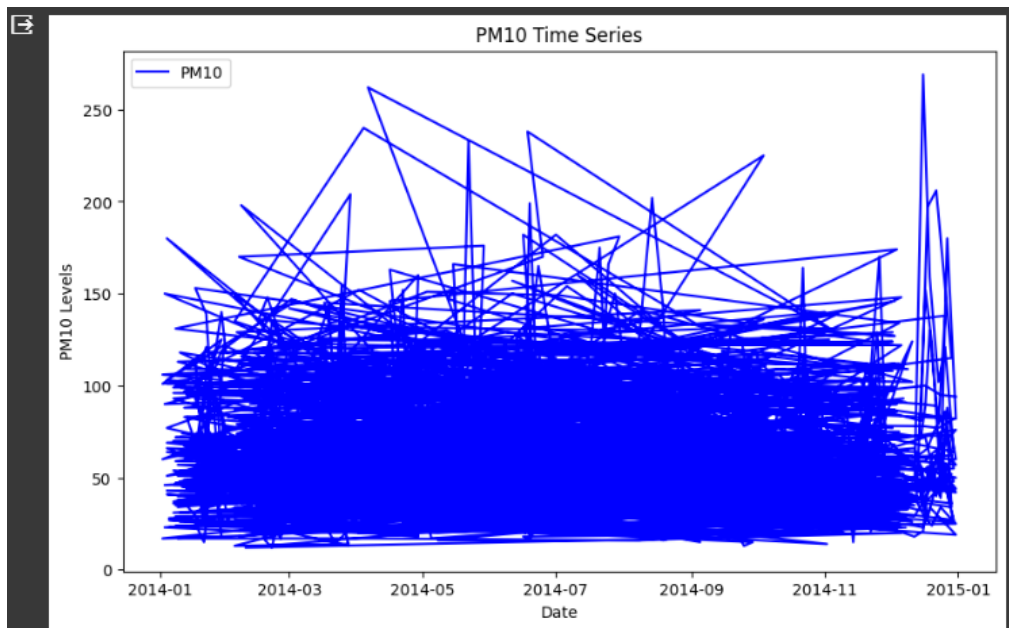
Explore the data through visualizations to gain insights into air quality trends and pollution levels.

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

# Load the dataset and perform data preprocessing
data = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')

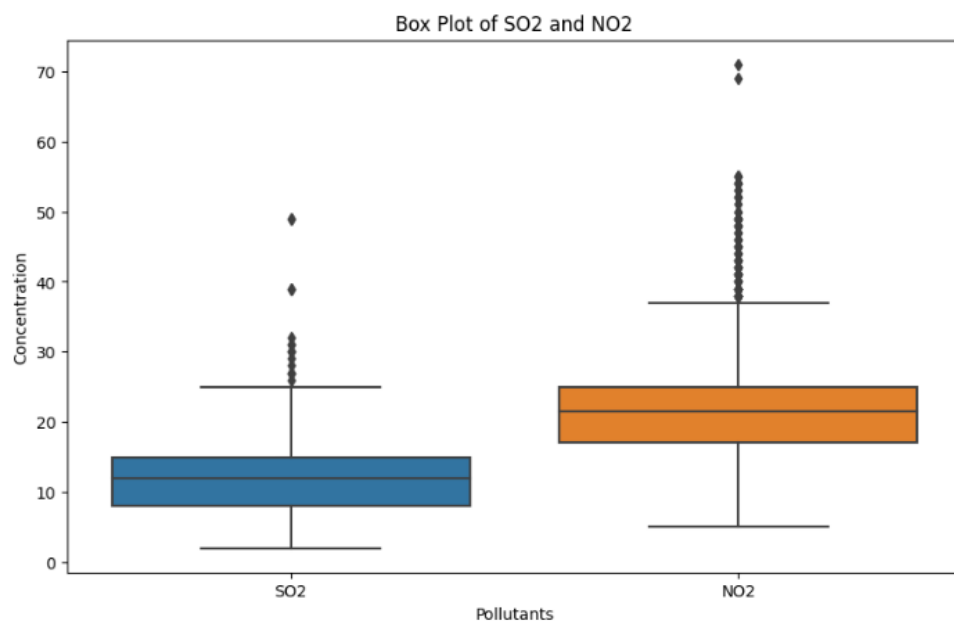
# Time Series Plot
plt.figure(figsize=(10, 6))
plt.plot(data.index, data['RSPM/PM10'], label='PM10', color='blue')
plt.xlabel('Date')
plt.ylabel('PM10 Levels')
plt.title('PM10 Time Series')
plt.legend()
plt.show()
```

Output:



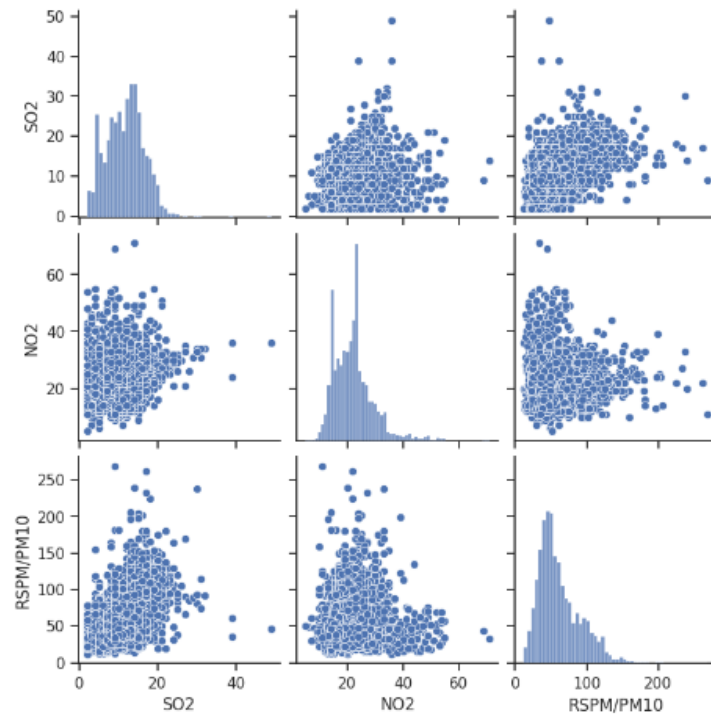
```
# Box Plot for SO2 and NO2
plt.figure(figsize=(10, 6))
sns.boxplot(data=data[['SO2', 'NO2']])
plt.title('Box Plot of SO2 and NO2')
plt.xlabel('Pollutants')
plt.ylabel('Concentration')
plt.show()
```

Output:



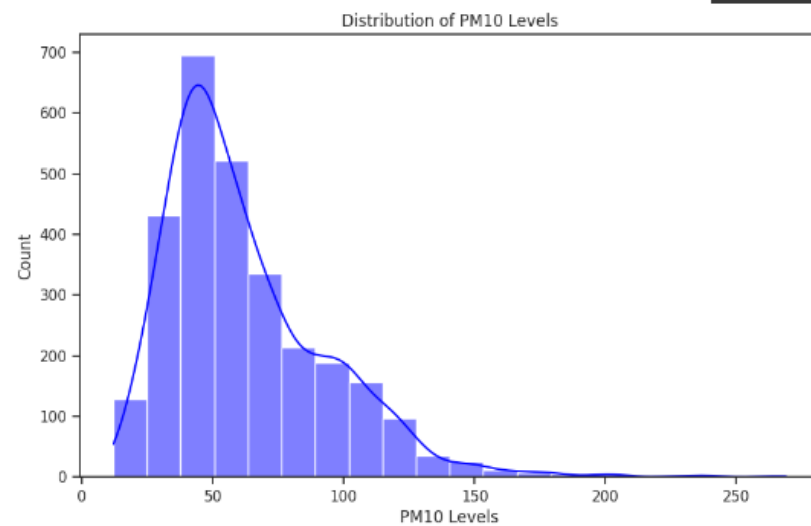
```
# Pair Plot for Pollution Correlations
sns.set(style="ticks")
sns.pairplot(data[['SO2', 'NO2', 'RSPM/PM10']])
plt.show()
```

Output:



```
# Histograms
plt.figure(figsize=(10, 6))
sns.histplot(data['RSPM/PM10'], kde=True, color='blue', bins=20)
plt.title('Distribution of PM10 Levels')
plt.xlabel('PM10 Levels')
plt.show()
```

Output:



Key findings from the air quality analysis and visualizations can include:

1. **Air Quality Trends:** The time series analysis shows the trend in RSPM/PM10 levels over time, allowing us to identify periods of high and low pollution. This information helps in understanding the long-term air quality trends in Tamil Nadu.
2. **Seasonal Patterns:** Seasonal decomposition reveals the seasonal component of air pollution levels. It can help identify recurring patterns, such as pollution peaks during certain months or seasons, which may be influenced by weather or other factors.
3. **Correlations:** The correlation heatmap provides insights into the relationships between different pollutants. For example, it can reveal whether there's a strong correlation between SO2 and NO2 levels or whether they both have an impact on RSPM/PM10 levels.
4. **Geospatial Distribution:** If you've created a geospatial map, it can highlight areas with high and low pollution levels. This can be valuable for understanding regional variations in air quality and targeting pollution control efforts.
5. **Outliers:** Through exploratory data analysis, you may identify outliers in the data, which could represent extreme pollution events or errors in data collection. Investigating these outliers can provide insights into specific pollution incidents.
6. **Distribution of Pollution Levels:** Histograms or other distribution visualizations show how pollution levels are distributed, which can be useful for understanding the typical range of air quality in the region.
7. **Box Plots:** Box plots can provide information about the central tendency and spread of SO2 and NO2 levels. They may reveal outliers and variations in the data.

Overall, these findings can help stakeholders, policymakers, and environmental scientists better understand air quality trends, the impact of different pollutants, and areas that require attention to mitigate air pollution in Tamil Nadu.

Importing the necessary python libraries:

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

Importing the csv file:

```
data = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')
data.head()
```

Output:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station		Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
0	38	01-02-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board		Industrial Area	11.0	17.0	55.0	NaN
1	38	01-07-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board		Industrial Area	13.0	17.0	45.0	NaN
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board		Industrial Area	12.0	18.0	50.0	NaN
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board		Industrial Area	15.0	16.0	46.0	NaN
4	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board		Industrial Area	13.0	14.0	42.0	NaN

```
data.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2879 entries, 0 to 2878
Data columns (total 11 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   Stn Code                             2879 non-null   int64
 1   Sampling Date                         2879 non-null   object
 2   State                                2879 non-null   object
 3   City/Town/Village/Area               2879 non-null   object
 4   Location of Monitoring Station        2879 non-null   object
 5   Agency                               2879 non-null   object
 6   Type of Location                     2879 non-null   object
 7   SO2                                  2868 non-null   float64
 8   NO2                                  2866 non-null   float64
 9   RSPM/PM10                           2875 non-null   float64
10   PM 2.5                               0 non-null      float64
dtypes: float64(4), int64(1), object(6)
memory usage: 247.5+ KB
```

Air quality analysis and creating visualizations:

Analyze air quality data to evaluate pollution levels and use data visualization tools to depict trends and insights graphically, enhancing understanding and communication of air quality information.

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
# Read the DataFrame
data = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')
# Summary statistics of the numerical columns
summary_stats = data.describe()
# Correlation between variables
correlation = data.corr()
```

Output:

```
<ipython-input-41-70c100cebe5f>:11: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. See
correlation = data.corr()
```

Visualizations:

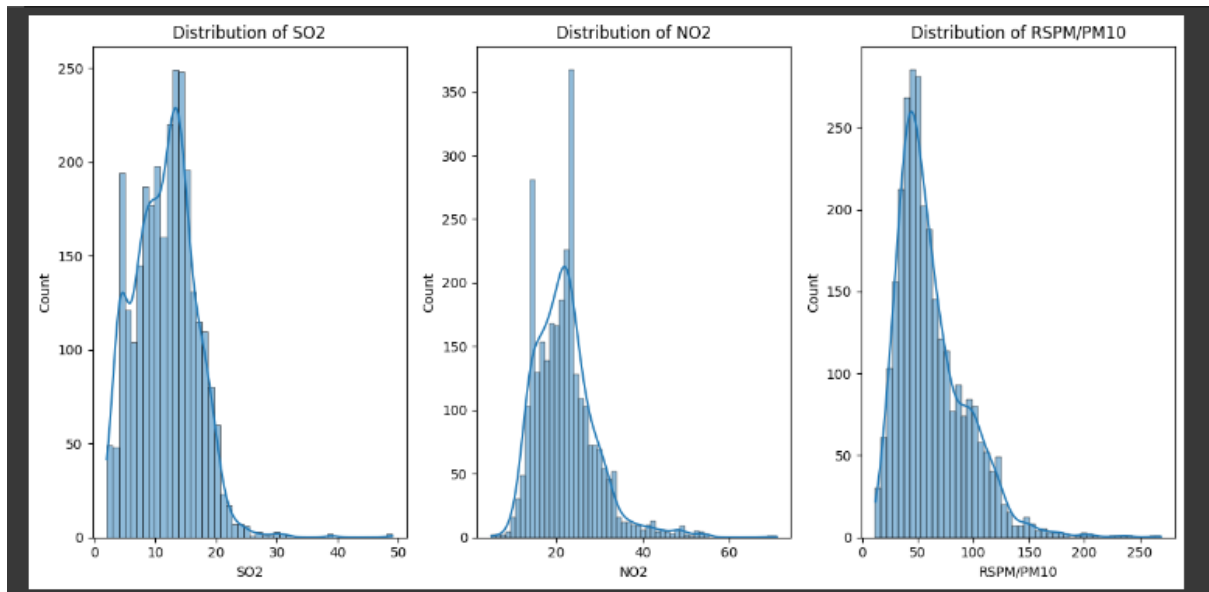
Distribution of SO2, NO2, RSPM/PM10

```
# Visualizations-Distribution of SO2, NO2, RSPM/PM10
plt.figure(figsize=(12, 6))
plt.subplot(1, 3, 1)
sns.histplot(data['SO2'].dropna(), kde=True)
plt.title('Distribution of SO2')
plt.subplot(1, 3, 2)
sns.histplot(data['NO2'].dropna(), kde=True)
```



```
plt.title('Distribution of NO2')
plt.subplot(1, 3, 3)
sns.histplot(data['RSPM/PM10'].dropna(), kde=True)
plt.title('Distribution of RSPM/PM10')
plt.tight_layout()
plt.show()
```

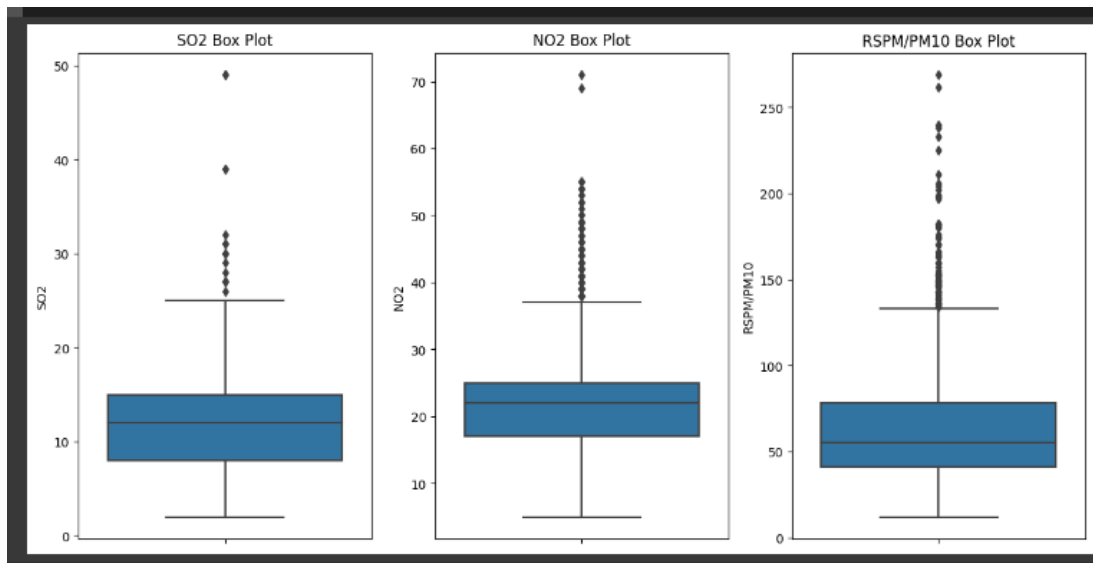
Output:



Box plots for SO2, NO2, and RSPM/PM10:

```
# Box plots for SO2, NO2, and RSPM/PM10
plt.figure(figsize=(12, 6))
plt.subplot(1, 3, 1)
sns.boxplot(data=data, y='SO2')
plt.title('SO2 Box Plot')
plt.subplot(1, 3, 2)
sns.boxplot(data=data, y='NO2')
plt.title('NO2 Box Plot')
plt.subplot(1, 3, 3)
sns.boxplot(data=data, y='RSPM/PM10')
plt.title('RSPM/PM10 Box Plot')
plt.tight_layout()
plt.show()
```

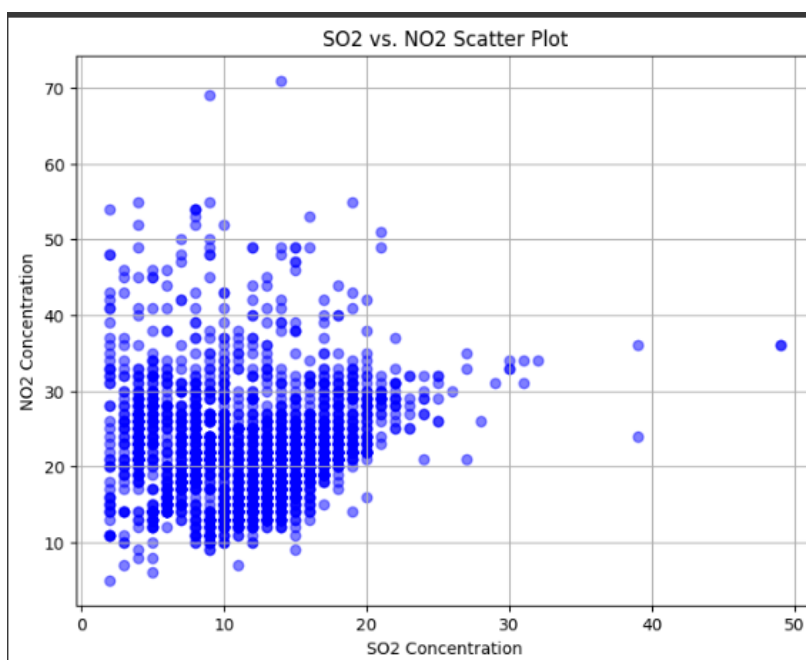
Output:



```
# Data Preprocessing (Handling Missing Values)
data.dropna(subset=['SO2', 'NO2'], inplace=True)

# Scatter Plot for SO2 vs. NO2
plt.figure(figsize=(8, 6))
plt.scatter(data['SO2'], data['NO2'], c='b', alpha=0.5)
plt.xlabel('SO2 Concentration')
plt.ylabel('NO2 Concentration')
plt.title('SO2 vs. NO2 Scatter Plot')
plt.grid(True)
plt.show()
```

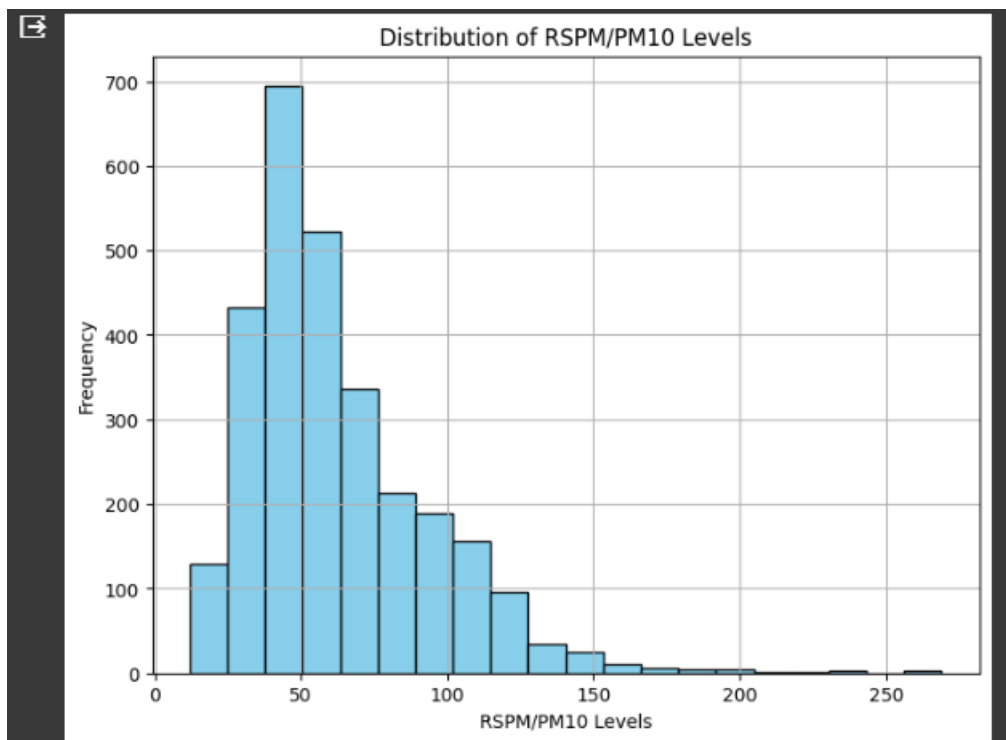
Output:



```
# Data Preprocessing (Handling Missing Values)
data.dropna(subset=['RSPM/PM10'], inplace=True)

# Histogram for RSPM/PM10 Levels
plt.figure(figsize=(8, 6))
plt.hist(data['RSPM/PM10'], bins=20, color='skyblue',
         edgecolor='black')
plt.xlabel('RSPM/PM10 Levels')
plt.ylabel('Frequency')
plt.title('Distribution of RSPM/PM10 Levels')
plt.grid(True)
plt.show()
```

Output:



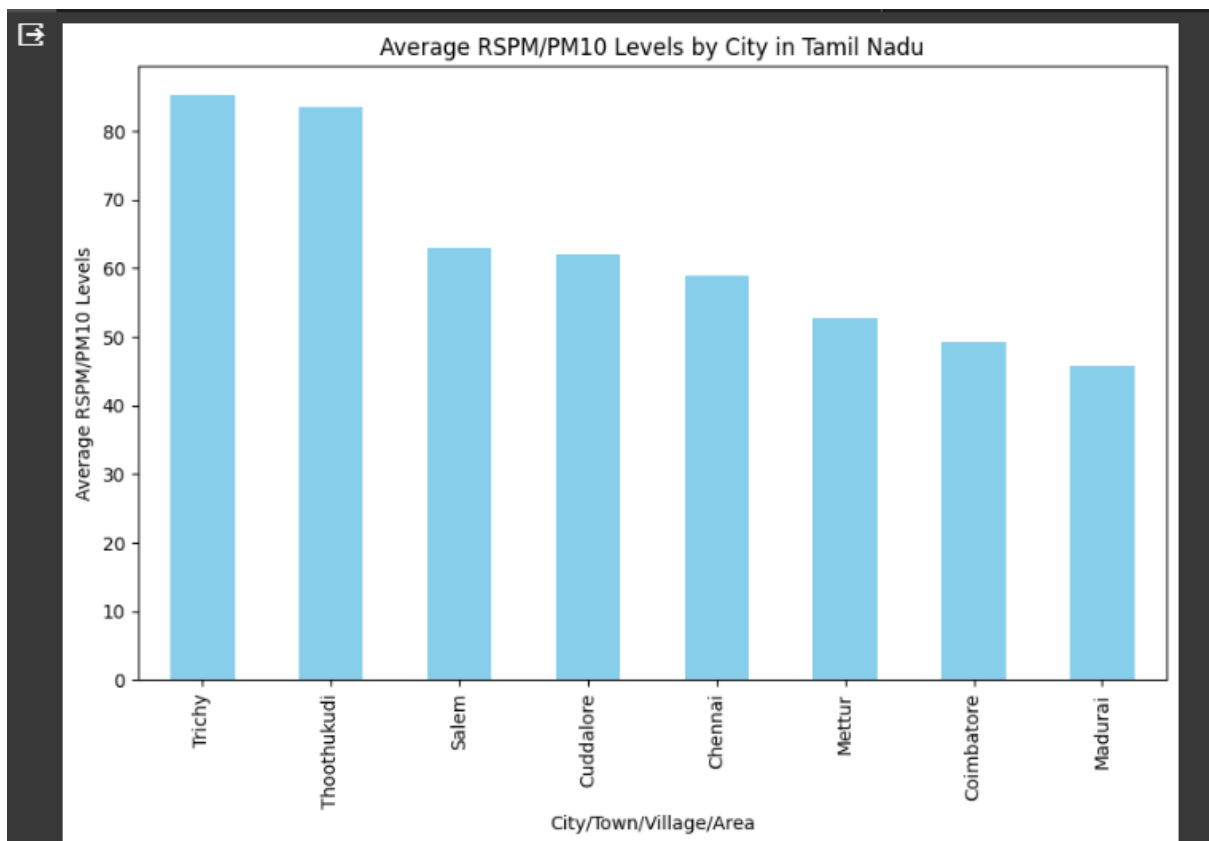
```
# Data Preprocessing (Handling Missing Values)
data.dropna(subset=['RSPM/PM10', 'City/Town/Village/Area'],
            inplace=True)

# Group data by city and calculate the average RSPM/PM10 levels
city_avg_rspm =
data.groupby('City/Town/Village/Area')['RSPM/PM10'].mean().sort_values(
ascending=False)

# Bar Chart for Average RSPM/PM10 Levels by City
```

```
plt.figure(figsize=(10, 6))
city_avg_rspm.plot(kind='bar', color='skyblue')
plt.xlabel('City/Town/Village/Area')
plt.ylabel('Average RSPM/PM10 Levels')
plt.title('Average RSPM/PM10 Levels by City in Tamil Nadu')
plt.xticks(rotation=90)
plt.show()
```

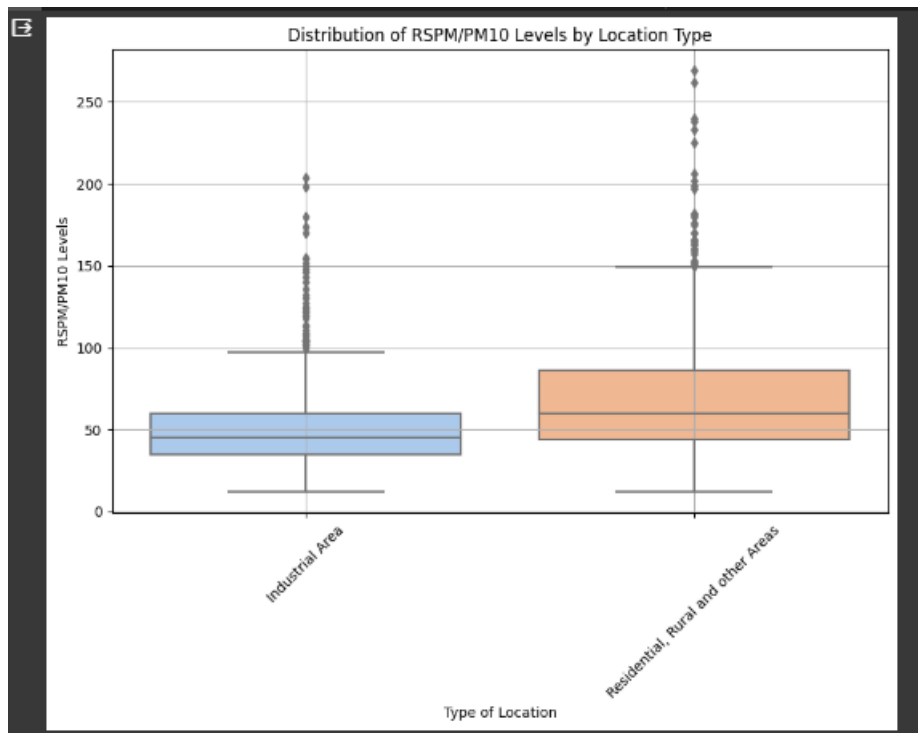
Output:



```
# Data Preprocessing (Handling Missing Values)
data.dropna(subset=['RSPM/PM10', 'Type of Location'], inplace=True)

# Box Plot for RSPM/PM10 Levels by Type of Location
plt.figure(figsize=(10, 6))
sns.boxplot(x='Type of Location', y='RSPM/PM10', data=data,
palette='pastel')
plt.xlabel('Type of Location')
plt.ylabel('RSPM/PM10 Levels')
plt.title('Distribution of RSPM/PM10 Levels by Location Type')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```

Output:



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

# Data Collection
data = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')

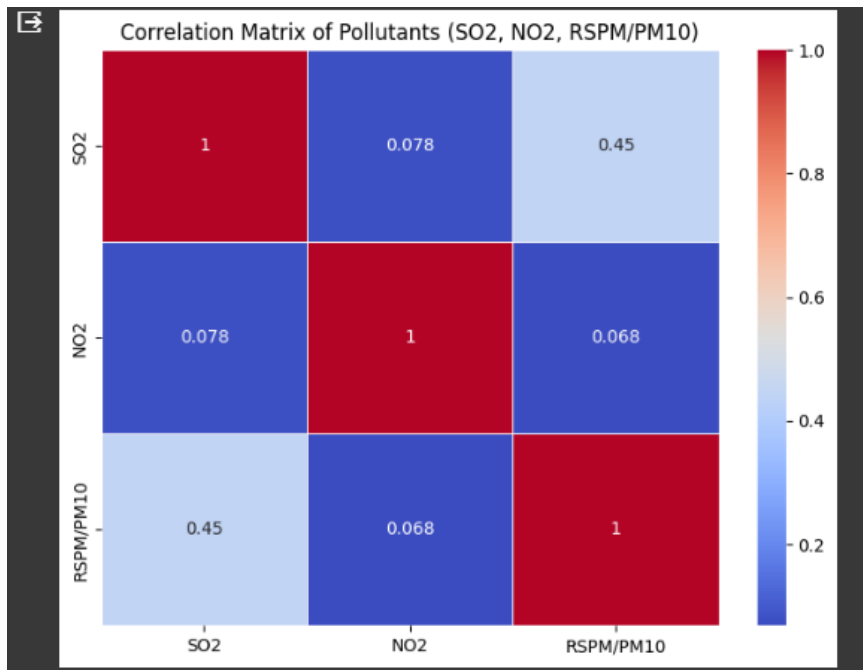
# Data Preprocessing (Handling Missing Values)
data.dropna(subset=['SO2', 'NO2', 'RSPM/PM10'], inplace=True)

# Select columns for analysis
selected_columns = ['SO2', 'NO2', 'RSPM/PM10']

# Calculate and visualize the correlation matrix
correlation_matrix = data[selected_columns].corr()

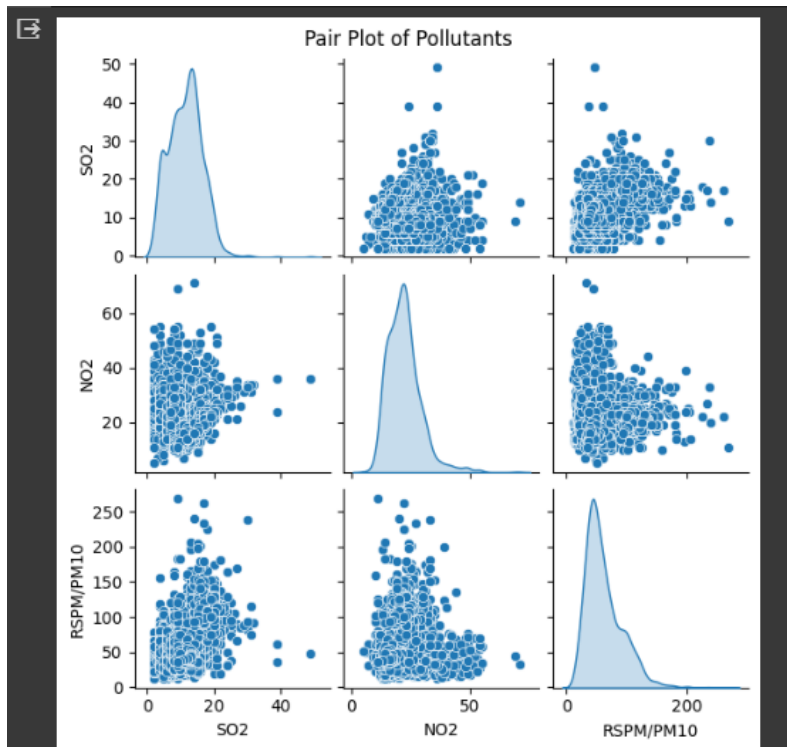
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',
            linewidths=.5)
plt.title('Correlation Matrix of Pollutants (SO2, NO2, RSPM/PM10)')
plt.show()
```

Output:



```
# Pair plot for visualization
sns.pairplot(data[selected_columns], diag_kind='kde', height=2)
plt.suptitle('Pair Plot of Pollutants', y=1.02)
plt.show()
```

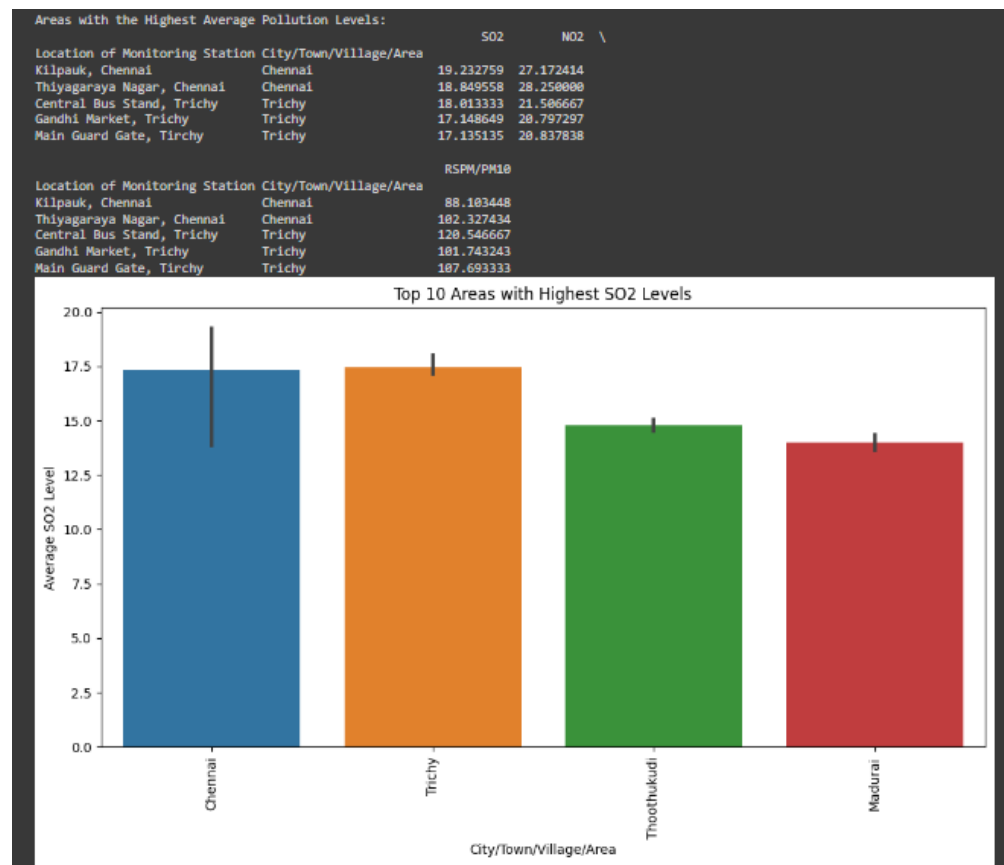
Output:



Calculate average SO2, NO2, and RSPM/PM10 levels across different monitoring stations, cities, or areas. Identify pollution trends and areas with high pollution levels.

```
# Group the data by the relevant columns (e.g., 'Location of Monitoring Station', 'City/Town/Village/Area')
grouped_data = data.groupby(['Location of Monitoring Station', 'City/Town/Village/Area'])
# Calculate the average levels for SO2, NO2, and RSPM/PM10
average_levels = grouped_data[['SO2', 'NO2', 'RSPM/PM10']].mean()
# Sort the data to identify areas with high pollution levels
sorted_data = average_levels.sort_values(by=['SO2', 'NO2', 'RSPM/PM10'], ascending=False)
# Display the areas with the highest pollution levels
print("Areas with the Highest Average Pollution Levels:")
print(sorted_data.head())
# You can also reset the index for further analysis or visualization
sorted_data.reset_index(inplace=True)
# Plotting the data for visualization
import matplotlib.pyplot as plt
# Bar plot for the top 10 areas with the highest SO2 levels
plt.figure(figsize=(12, 6))
sns.barplot(data=sorted_data.head(10), x='City/Town/Village/Area', y='SO2')
plt.title("Top 10 Areas with Highest SO2 Levels")
plt.xlabel('City/Town/Village/Area')
plt.ylabel('Average SO2 Level')
plt.xticks(rotation=90)
plt.show()
```

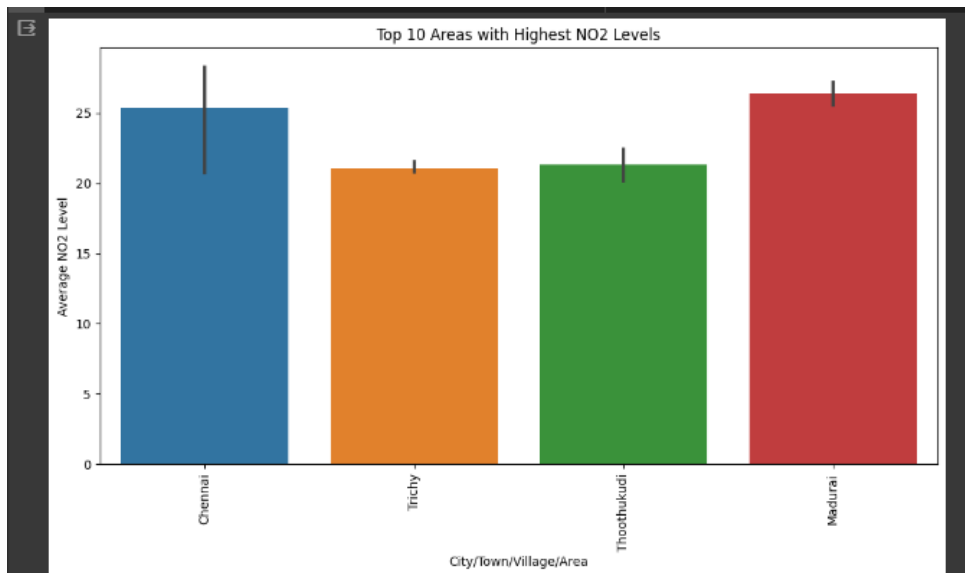
Output:



Bar plot for the top 10 areas with the highest NO2 levels:

```
# Bar plot for the top 10 areas with the highest NO2 levels
plt.figure(figsize=(12, 6))
sns.barplot(data=sorted_data.head(10), x='City/Town/Village/Area', y='NO2')
plt.title('Top 10 Areas with Highest NO2 Levels')
plt.xlabel('City/Town/Village/Area')
plt.ylabel('Average NO2 Level')
plt.xticks(rotation=90)
plt.show()
```

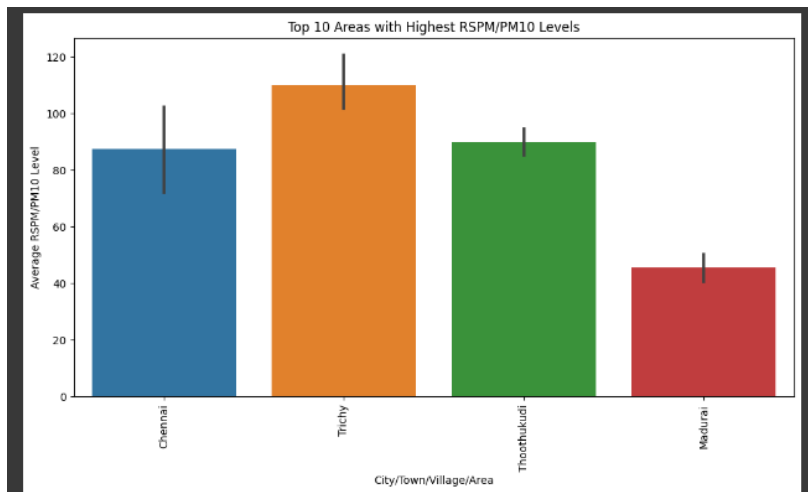
Output:



Bar plot for the top 10 areas with the highest RSPM/PM10 levels:

```
# Bar plot for the top 10 areas with the highest RSPM/PM10 levels
plt.figure(figsize=(12, 6))
sns.barplot(data=sorted_data.head(10), x='City/Town/Village/Area', y='RSPM/PM10')
plt.title('Top 10 Areas with Highest RSPM/PM10 Levels')
plt.xlabel('City/Town/Village/Area')
plt.ylabel('Average RSPM/PM10 Level')
plt.xticks(rotation=90)
plt.show()
```


Output:

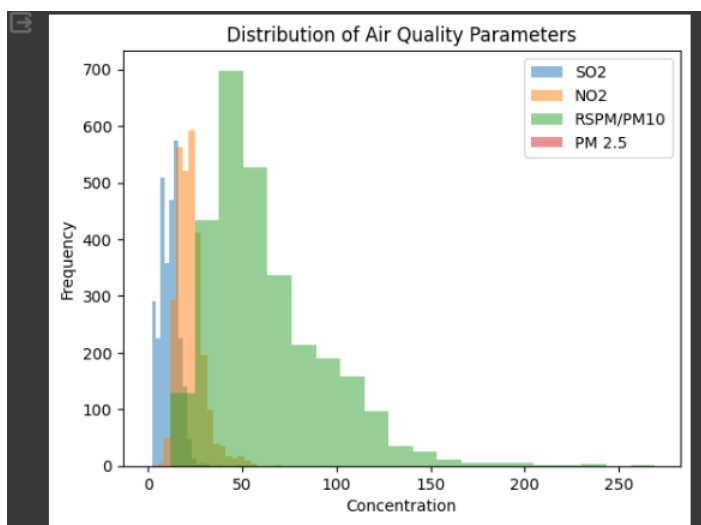


Generate informative visualizations using Matplotlib and Seaborn to represent data graphically, aiding in data exploration, analysis, and communication of insights.

#Histograms to visualize the distribution of air quality parameters like SO2, NO2, RSPM/PM10, and PM2.5:

```
import matplotlib.pyplot as plt
plt.hist(df['SO2'].dropna(), bins=20, alpha=0.5, label='SO2')
plt.hist(df['NO2'].dropna(), bins=20, alpha=0.5, label='NO2')
plt.hist(df['RSPM/PM10'].dropna(), bins=20, alpha=0.5, label='RSPM/PM10')
plt.hist(df['PM 2.5'].dropna(), bins=20, alpha=0.5, label='PM 2.5')
plt.xlabel('Concentration')
plt.ylabel('Frequency')
plt.legend()
plt.title('Distribution of Air Quality Parameters')
plt.show()
```

Output:

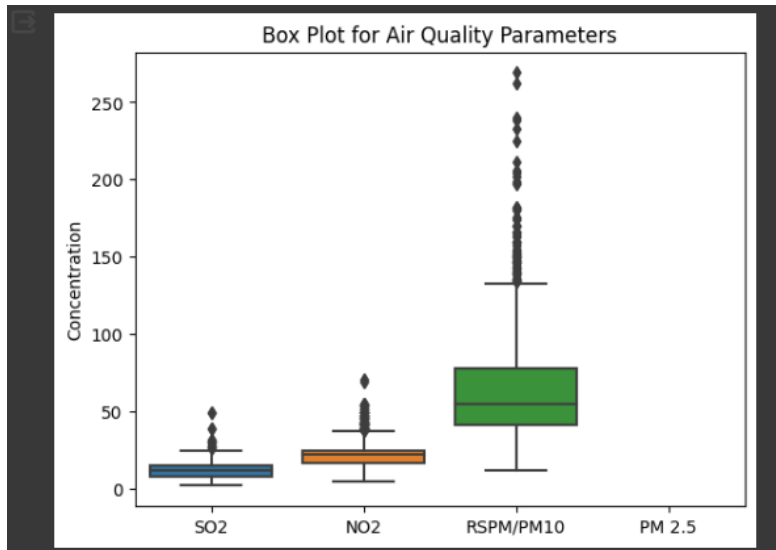


Box Plots to visualize the distribution and identify outliers:

#Box Plots to visualize the distribution and identify outliers:

```
import seaborn as sns
sns.boxplot(data=data[['SO2', 'NO2', 'RSPM/PM10', 'PM 2.5']], orient='v')
plt.ylabel('Concentration')
plt.title('Box Plot for Air Quality Parameters')
plt.show()
```

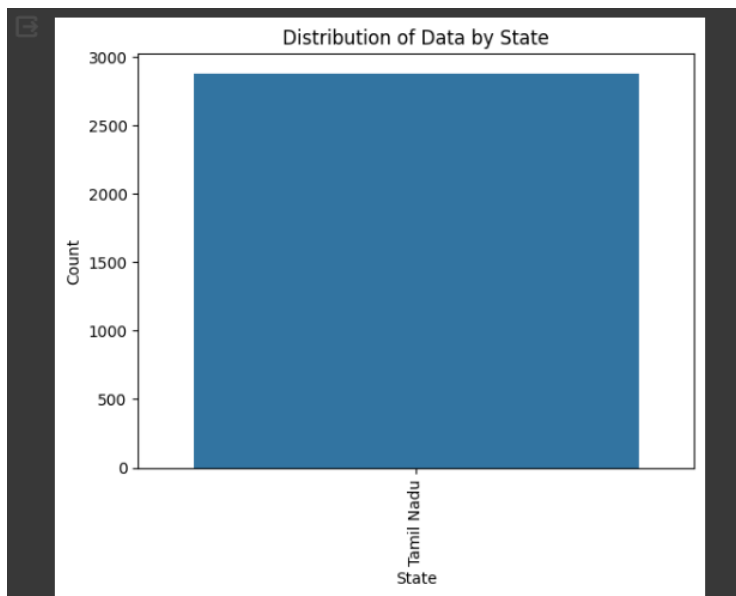
Output:



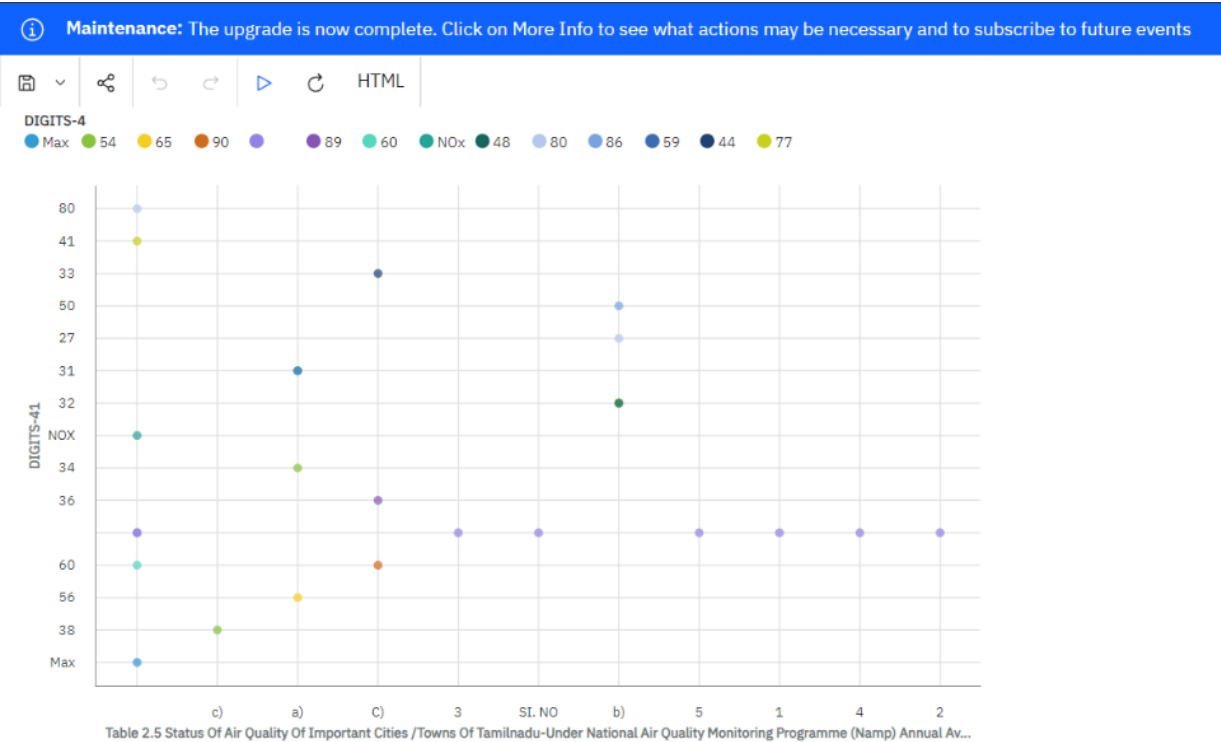
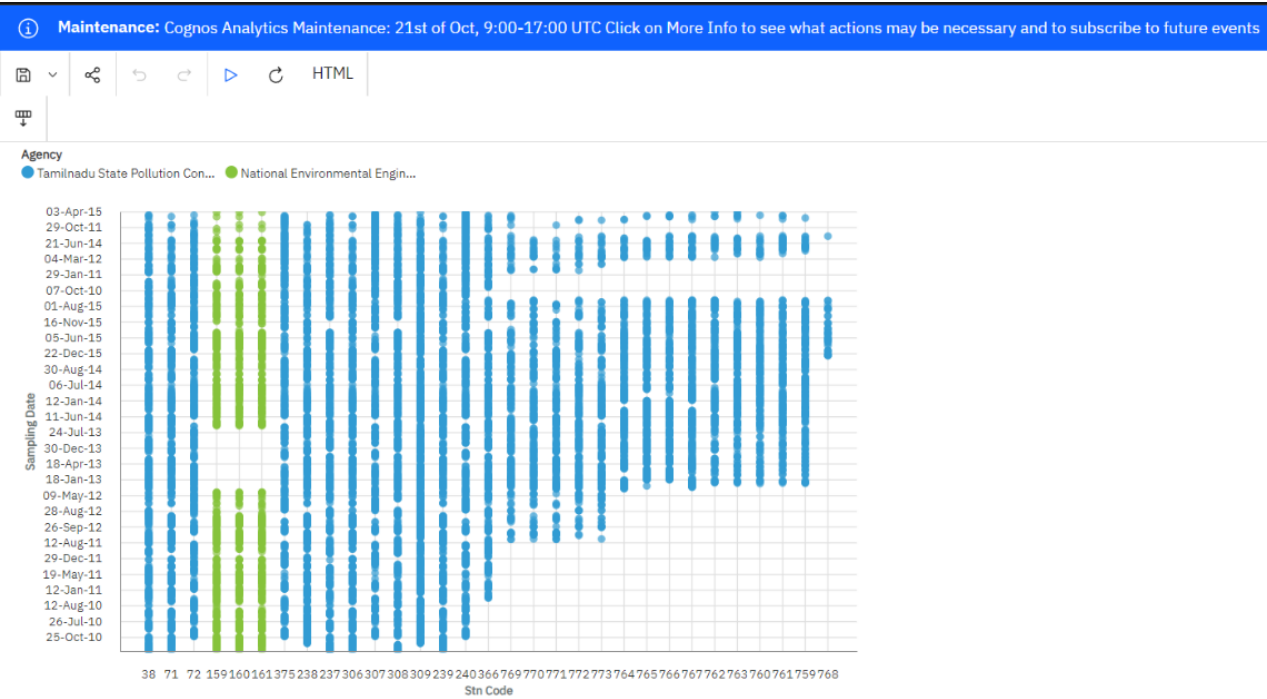
#Bar Plots for visualizing categorical data, e.g., "State," "City/Town/Village/Area," or "Type of Location"


```
sns.countplot(data=data, x='State')
plt.xlabel('State')
plt.ylabel('Count')
plt.title('Distribution of Data by State')
plt.xticks(rotation=90)
plt.show()
```

Output:

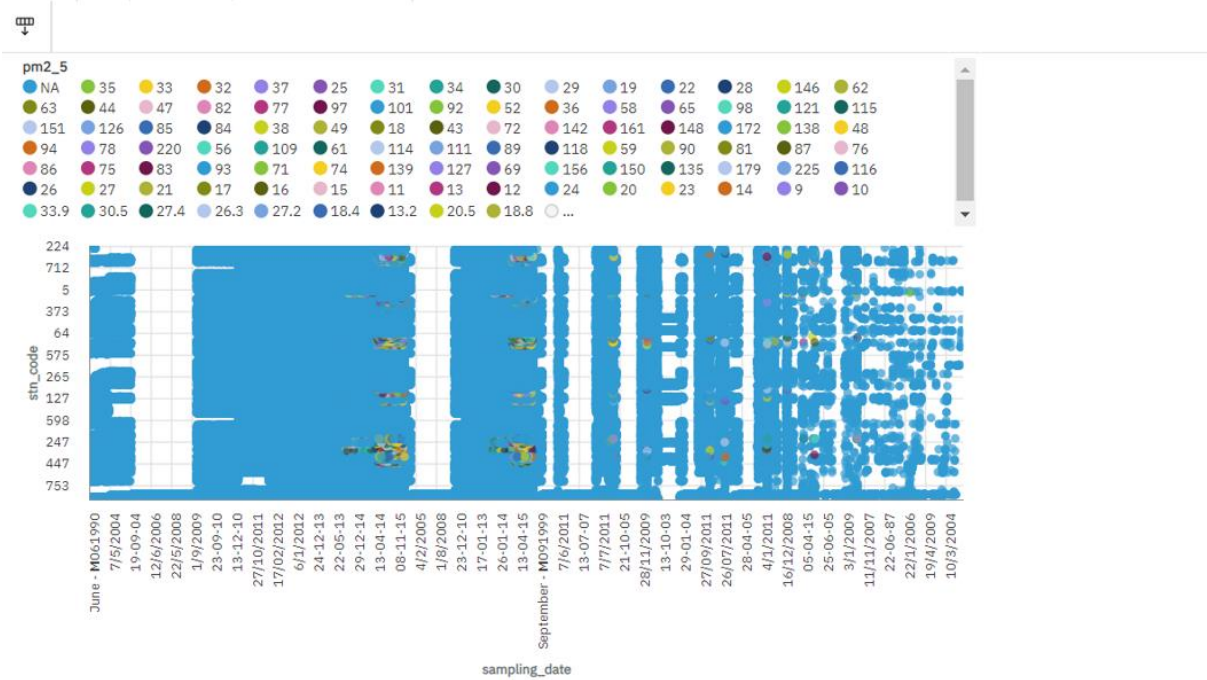



Visualization using IBM Cognos:



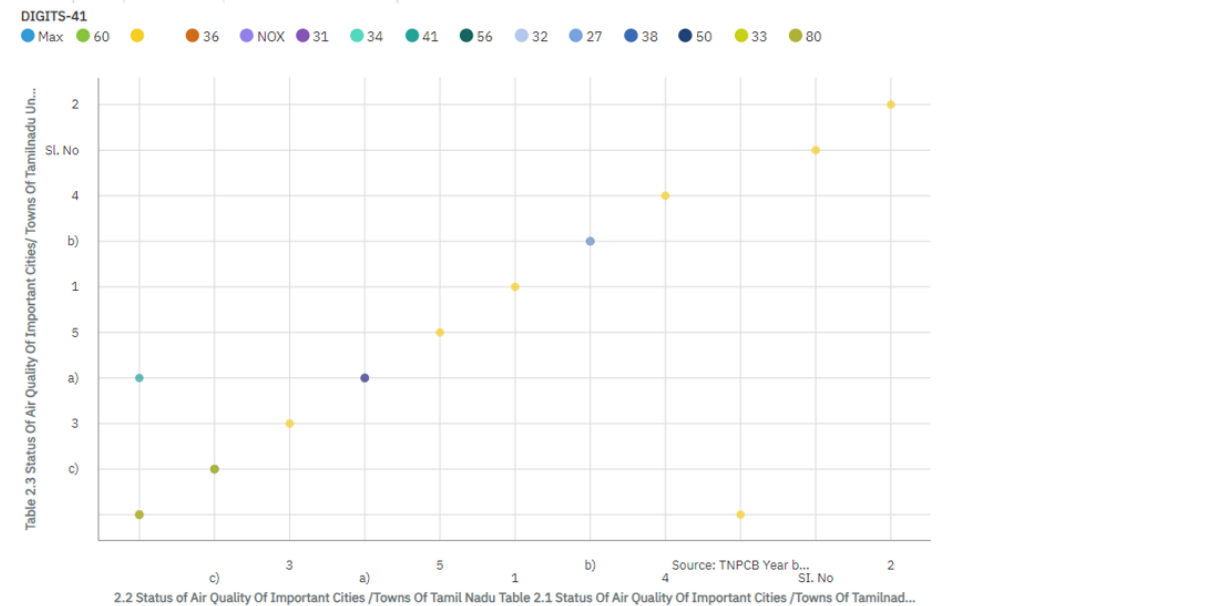
 **Maintenance:** The upgrade is now complete. Click on More Info to see what actions may be necessary and to subscribe to future events

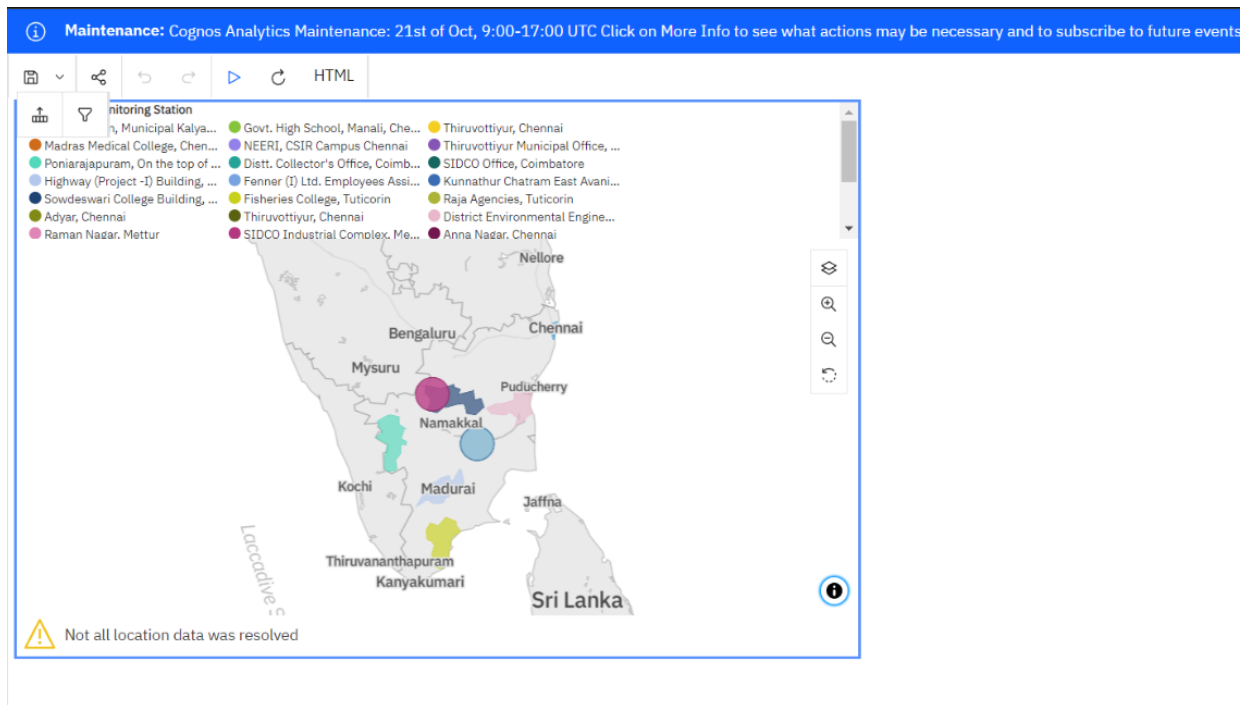
      HTML



 **Maintenance:** The upgrade is now complete. Click on More Info to see what actions may be necessary and to subscribe to future events

      HTML





Conclusion:

The preprocessing phase of the "Air Quality Analysis in Tamil Nadu" project is a crucial preparatory stage. By collecting, cleaning, and transforming the air quality dataset, we ensure that the data is of high quality and well-suited for analysis and modeling. This process is fundamental to our objective of gaining insights into air pollution trends, identifying pollution hotspots, and constructing a predictive model for RSPM/PM10 estimation based on SO₂ and NO₂ levels. With clean and consistent data, we are better equipped to make informed decisions and take steps toward improving air quality in Tamil Nadu.