Project Title: Air Quality Assessment in Tamil Nadu

1. **Problem Definition:** The primary objective of this project is to conduct a comprehensive analysis of air quality data collected from monitoring stations in Tamil Nadu. Our specific goals are:

Objective 1: Air Quality Trends Analysis: We will examine how air quality parameters such as RSPM/PM10, SO2 and NO2 have evolved over time across different regions.

Objective 2: Pollution Hotspot Identification – We aim to identify and pinpoint areas or monitoring stations with consistently high pollution levels, known as pollution hotspots.

Objective 3: Predictive modelling – We will develop a predictive model. The model's purpose is to estimate RSPM/PM10 levels based on SO2 and NO2 levels, which will enable us to forecast air quality.

2. Design Thinking:

2.1 Analysis Objectives:

To ensure clarity in our project, we have set the following analysis objectives:

Air Quality Trend Analysis: Examine historical air quality data to identify trends and variations in pollution levels over time.

Pollution Hotspot Identification: Determine regions within Tamil Nadu that consistently exhibit high pollution levels.

Predictive modeling: Develop a predictive model that can estimate RSPM/PM10 levels based on SO2 and NO2 concentrations.

2.2 The Analysis Approach:

Data collection:

We will begin by acquiring air quality data from the "Location-wise Daily Ambient Air Quality of Tamil Nadu for the year 2014" dataset, thoughtfully provided by the Tamil Nadu government. It is imperative that this dataset includes vital parameters such as RSPM/PM10, SO2, and NO2. We will meticulously validate the reliability and completeness of this dataset.

Data preprocessing:

The next step involves the comprehensive cleaning and preprocessing of the dataset using IBM Cognos. Our focus will be on addressing issues related to missing values, format inconsistencies, and potential outliers. This meticulous data preparation is indispensable to ensuring data integrity and precision for all subsequent analyses.

Exploratory Data Analysis (EDA):

A diverse range of exploratory data analysis (EDA) techniques will be applied during this phase using IBM Cognos. Our objective is to delve deeply into the dataset, identify crucial insights, and uncover trends, patterns, and potential outliers within the air quality data specific to Tamil Nadu for the year 2014. This phase plays a pivotal role in providing us with a comprehensive understanding of the dataset's intricacies and unique characteristics.

Statistical analysis:

Our approach to assessing air quality trends throughout Tamil Nadu for the year 2014 will involve the implementation of advanced statistical tests and analyses using IBM Cognos. We will meticulously explore variations in air quality parameters such as RSPM/PM10, SO2, and NO2 across different regions and monitoring stations within the state. Should pollution hotspots exist, we will rigorously identify and statistically validate them.

Machine learning modeling:

The final step in our project entails the development of a machine learning predictive model, specifically a regression model, using IBM Cognos. The primary objective of this model is to estimate RSPM/PM10 levels based on the levels of SO2 and NO2. A dedicated effort will be put into training and thoroughly evaluating the model's performance using IBM Cognos to ensure its accuracy and reliability in predicting air quality parameters.

By following these steps and leveraging IBM Cognos for analysis, we will be wellequipped to analyze air quality trends, pinpoint pollution hotspots, and create a predictive model tailored to the unique air quality conditions in Tamil Nadu.

XGBoost:

• Predictive Performance: XGBoost consistently achieves high accuracy across

diverse datasets thanks to its ensemble approach, which uncovers complex

relationships in the data.

• Regularization: It effectively prevents overfitting through L1 and L2

regularization techniques, enabling the model to generalize better to new data.

• Handling Missing Data: XGBoost has built-in support for managing missing

values, reducing the need for extensive data preprocessing and imputation.

• Feature Importance: XGBoost provides valuable insights into which variables

have the most significant impact on the target variable. This aids in feature selection and enhances data understanding.

• Efficiency and Scalability: XGBoost is optimized for speed and can handle large

datasets efficiently. This makes it suitable for real-world applications with big data, and its parallel processing capabilities accelerate model training. • In conclusion, XGBoost is the top choice for predictive modeling, especially in complex datasets like air quality analysis. It excels in handling diverse data

types, mitigating overfitting, and offering valuable insights, making it a valuable

tool for environmental monitoring and air quality predictions.

Data Loading and Preprocessing:

The data was loaded from the CSV file 'cpcb_dly_aq_tamil_nadu2014.csv'. During the preprocessing stage, missing values were handled, and duplicate records were removed.

- **Data Shape**: The dataset contains X rows and Y columns, offering a significant volume of data for analysis.
- **Missing Values**: Null values in the PM2.5 column were handled by removing the respective entries, ensuring data integrity

```
print("INFO:")
print(df.info())
```

```
print("\nDescribe:")
print(df.describe())
```

```
print("\nShape")
print(df.shape)
```

```
<bound method NDFrame.head of       Stn Code Sampling Date       State City/Town/Village/Area  ...   SO2    NO2 RSPM/PM10  PM 2.5
0           38      01-02-14  Tamil Nadu                Chennai  ...  11.0  17.0       55.0     NaN
1           38      01-07-14  Tamil Nadu                Chennai  ...  13.0  17.0       45.0     NaN
2           38      21-01-14  Tamil Nadu                Chennai  ...  12.0  18.0       50.0     NaN
3           38      23-01-14  Tamil Nadu                Chennai  ...  15.0  16.0       46.0     NaN
4           38      28-01-14  Tamil Nadu                Chennai  ...  13.0  14.0       42.0     NaN
...        ...           ...         ...                    ...  ...   ...   ...        ...     ...
2874       773      12-03-14  Tamil Nadu                 Trichy  ...  15.0  18.0      102.0     NaN
2875       773      12-10-14  Tamil Nadu                 Trichy  ...  12.0  14.0       91.0     NaN
2876       773      17-12-14  Tamil Nadu                 Trichy  ...  19.0  22.0      100.0     NaN
2877       773      24-12-14  Tamil Nadu                 Trichy  ...  15.0  17.0       95.0     NaN
2878       773      31-12-14  Tamil Nadu                 Trichy  ...  14.0  16.0       94.0     NaN

[2879 rows x 11 columns]>
```

```
Describe:
          Stn Code          SO2          NO2    RSPM/PM10  PM 2.5
count  2879.000000  2868.000000  2866.000000  2875.000000     0.0
mean    475.750261    11.503138    22.136776    62.494261     NaN
std     277.675577     5.051702     7.128694    31.368745     NaN
min      38.000000     2.000000     5.000000    12.000000     NaN
25%     238.000000     8.000000    17.000000    41.000000     NaN
50%     366.000000    12.000000    22.000000    55.000000     NaN
75%     764.000000    15.000000    25.000000    78.000000     NaN
max     773.000000    49.000000    71.000000   269.000000     NaN
```

```
print("\nREMOVING COLUMNS WITH NULL VALUES\n ")
df = df.drop('PM 2.5', axis=1)
df.dropna(inplace=True)
```

```
REMOVING COLUMNS WITH NULL VALUES
```

```
print("\nDROPPING DUPLICATE ROWS:\n")
df.drop_duplicates(subset=None, inplace=True)
print(df.head)
```

```
DROPPING DUPLICATE ROWS:

<bound method NDFrame.head of       Stn Code Sampling Date      State City/Town/Village/Area  ...                 Type of Loc
ation   SO2   NO2  RSPM/PM10
0           38      01-02-14  Tamil Nadu                Chennai  ...                Industrial Area  11.0  17.0       55.0
1           38      01-07-14  Tamil Nadu                Chennai  ...                Industrial Area  13.0  17.0       45.0
2           38      21-01-14  Tamil Nadu                Chennai  ...                Industrial Area  12.0  18.0       50.0
3           38      23-01-14  Tamil Nadu                Chennai  ...                Industrial Area  15.0  16.0       46.0
4           38      28-01-14  Tamil Nadu                Chennai  ...                Industrial Area  13.0  14.0       42.0
...        ...           ...         ...                    ...  ...                            ...   ...   ...        ...
2874       773      12-03-14  Tamil Nadu                 Trichy  ...  Residential, Rural and other Areas  15.0  18.0      102.0
2875       773      12-10-14  Tamil Nadu                 Trichy  ...  Residential, Rural and other Areas  12.0  14.0       91.0
2876       773      17-12-14  Tamil Nadu                 Trichy  ...  Residential, Rural and other Areas  19.0  22.0      100.0
2877       773      24-12-14  Tamil Nadu                 Trichy  ...  Residential, Rural and other Areas  15.0  17.0       95.0
2878       773      31-12-14  Tamil Nadu                 Trichy  ...  Residential, Rural and other Areas  14.0  16.0       94.0

[2862 rows x 10 columns]>

CONVERTING TO DATE-TIME FORMAT

d:\nm_dsc\preair.py:21: UserWarning: Could not infer format, so each element will be parsed individually, falling back to `dateut
il`. To ensure parsing is consistent and as-expected, please specify a format.
  df['Sampling Date'] = pd.to_datetime(df['Sampling Date'])

Head after preprocessing:
<bound method NDFrame.head of       Stn Code Sampling Date      State City/Town/Village/Area  ...                 Type of Loc
ation   SO2   NO2  RSPM/PM10
0           38    2014-01-02  Tamil Nadu                Chennai  ...                Industrial Area  11.0  17.0       55.0
1           38    2014-01-07  Tamil Nadu                Chennai  ...                Industrial Area  13.0  17.0       45.0
2           38    2014-01-21  Tamil Nadu                Chennai  ...                Industrial Area  12.0  18.0       50.0
3           38    2014-01-23  Tamil Nadu                Chennai  ...                Industrial Area  15.0  16.0       46.0
```

Data Exploration:

Summary Statistics:

- **General Statistics**: Summary statistics for numerical columns were computed using df.describe(). These statistics include count, mean, standard deviation, minimum, quartiles, and maximum values for each numerical attribute.

Unique Locations and Cities:

- **Unique Locations**: A list of unique monitoring locations was generated using unique_locations, providing an understanding of the diversity of data collection sites.
- **City-wise Monitoring Stations**: The count of monitoring stations in each city was calculated using city_station_counts, shedding light on the distribution of monitoring infrastructure across different cities.

```
unique_locations = df['Location of Monitoring Station'].unique()
print("\nLocations of Monitoring Stations:")
print(unique_locations)
```

```
Locations of Monitoring Stations:
['Kathivakkam, Municipal Kalyana Mandapam, Chennai'
 'Govt. High School, Manali, Chennai.' 'Thiruvottiyur,  Chennai'
 'Thiyagaraya Nagar, Chennai' 'Anna Nagar, Chennai' 'Adyar, Chennai'
 'Kilpauk, Chennai' 'Madras Medical College, Chennai'
 'Thiruvottiyur Municipal Office, Chennai' 'NEERI, CSIR Campus Chennai'
 'Poniarajapuram, On the top of DEL, Coimbatore'
 'SIDCO Office, Coimbatore' "Distt. Collector's Office, Coimbatore"
 'Eachangadu Villagae'
 'District Environmental Engineer Office, Imperial Road, Cuddalore'
 'SIPCOT Industrial Complex, Cuddalore'
 'Highway (Project -I) Building, Madurai'
 'Fenner (I) Ltd. Employees Asiciation Building Kochadai, Madurai'
 'Kunnathur Chatram East Avani Mollai Street, Madurai'
 'Raman Nagar, Mettur' 'SIDCO Industrial Complex, Mettur'
 'Sowdeswari College Building, Salem' 'Fisheries College, Tuticorin'
 'AVM Jewellery Building, Tuticorin' 'Raja Agencies, Tuticorin'
 'Gandhi Market, Trichy' 'Main Guard Gate, Tirchy'
 'Bishop Heber College, Tirchy' 'Golden Rock, Trichy'
 'Central Bus Stand, Trichy']
```

```
city_station_counts = df.groupby('City/Town/Village/Area')['Location of Monitoring Station'].count().reset_index()

city_station_counts.columns = ['City', 'Number of Monitoring Stations']

print("\nCity-wise Number of Monitoring Stations:")
print(city_station_counts)
```

```
City-wise Number of Monitoring Stations:
         City  Number of Monitoring Stations
0     Chennai                            995
1  Coimbatore                            289
2   Cuddalore                            294
3     Madurai                            294
4      Mettur                            205
5       Salem                            131
6  Thoothukudi                           290
7      Trichy                            364
```

```
location_counts = df.groupby(['City/Town/Village/Area', 'Location of Monitoring Station']).size().reset_index()
location_counts.columns = ['City', 'Location', 'Number of Rows']

print("\nLocation-wise Number of Rows with City:")
print(location_counts)
```

```
Location-wise Number of Rows with City:
          City                                        Location  \
0      Chennai                                  Adyar, Chennai
1      Chennai                             Anna Nagar, Chennai
2      Chennai                 Govt. High School, Manali, Chennai.
3      Chennai  Kathivakkam, Municipal Kalyana Mandapam, Chennai
4      Chennai                                Kilpauk, Chennai
5      Chennai                   Madras Medical College, Chennai
6      Chennai                        NEERI, CSIR Campus Chennai
7      Chennai           Thiruvottiyur Municipal Office, Chennai
8      Chennai                          Thiruvottiyur,  Chennai
9      Chennai                        Thiyagaraya Nagar, Chennai
10  Coimbatore              Distt. Collector's Office, Coimbatore
11  Coimbatore      Poniarajapuram, On the top of DEL, Coimbatore
12  Coimbatore                          SIDCO Office, Coimbatore
13   Cuddalore  District Environmental Engineer Office, Imperi...
14   Cuddalore                               Eachangadu Villagae
15   Cuddalore                SIPCOT Industrial Complex, Cuddalore
16     Madurai  Fenner (I) Ltd. Employees Assiciation Building...
17     Madurai             Highway (Project -I) Building, Madurai
18     Madurai  Kunnathur Chatram East Avani Mollai Street, Ma...
19      Mettur                               Raman Nagar, Mettur
20      Mettur                  SIDCO Industrial Complex, Mettur
21       Salem                   Sowdeswari College Building, Salem
22  Thoothukudi                    AVM Jewellery Building, Tuticorin
23  Thoothukudi                         Fisheries College, Tuticorin
24  Thoothukudi                            Raja Agencies, Tuticorin
25      Trichy                        Bishop Heber College, Tirchy
26      Trichy                           Central Bus Stand, Trichy
27      Trichy                              Gandhi Market, Trichy
28      Trichy                                Golden Rock, Trichy
29      Trichy                            Main Guard Gate, Tirchy
```

Pollution Levels:

- **Average Pollution Levels by City**: A bar chart was constructed to illustrate average levels of SO2, NO2, and RSPM/PM10 in each city. This offers a comparative view of pollution across various cities.

```
summary = df.groupby(['City/Town/Village/Area', 'Location of Monitoring Station'])[['SO2', 'NO2', 'RSPM/PM10']].agg(['sum', 'mean']).reset_index()

summary.columns = ['City', 'Location', 'SO2 Sum', 'SO2 Average', 'NO2 Sum', 'NO2 Average', 'RSPM/PM10 Sum', 'RSPM/PM10 Average']

print("\nSummary of SO2, NO2, and RSPM/PM10 Levels by Location:")
print(summary)
```

```
Summary of SO2, NO2, and RSPM/PM10 Levels by Location:
          City                                          Location  SO2 Sum  \
0      Chennai                                    Adyar, Chennai  [illegible]
1      Chennai                               Anna Nagar, Chennai   1527.0
2      Chennai                Govt. High School, Manali, Chennai.   1213.0
3      Chennai  Kathivakkam, Municipal Kalyana Mandapam, Chennai   1215.0
4      Chennai                                  Kilpauk, Chennai   2231.0
5      Chennai                     Madras Medical College, Chennai    638.0
6      Chennai                          NEERI, CSIR Campus Chennai    516.0
7      Chennai             Thiruvottiyur Municipal Office, Chennai    719.0
8      Chennai                            Thiruvottiyur,  Chennai   1249.0
9      Chennai                         Thiyagaraya Nagar, Chennai   2114.0
10  Coimbatore               Distt. Collector's Office, Coimbatore    405.0
11  Coimbatore        Poniarajapuram, On the top of DEL, Coimbatore    425.0
12  Coimbatore                          SIDCO Office, Coimbatore    482.0
13   Cuddalore  District Environmental Engineer Office, Imperi...    802.0
14   Cuddalore                                  Eachangadu Villagae   1144.0
15   Cuddalore                SIPCOT Industrial Complex, Cuddalore    690.0
16     Madurai  Fenner (I) Ltd. Employees Assiciation Building...   1378.0
17     Madurai               Highway (Project -I) Building, Madurai   1147.0
18     Madurai  Kunnathur Chatram East Avani Mollai Street, Ma...   1391.0
19      Mettur                                 Raman Nagar, Mettur    780.0
20      Mettur                   SIDCO Industrial Complex, Mettur    948.0
21       Salem                   Sowdeswari College Building, Salem   1063.0
22  Thoothukudi                   AVM Jewellery Building, Tuticorin    893.0
23  Thoothukudi                        Fisheries College, Tuticorin   1351.0
24  Thoothukudi                            Raja Agencies, Tuticorin   1521.0
25      Trichy                        Bishop Heber College, Tirchy    826.0
26      Trichy                           Central Bus Stand, Trichy   1351.0
27      Trichy                               Gandhi Market, Trichy   1269.0
28      Trichy                                 Golden Rock, Trichy    853.0
29      Trichy                              Main Guard Gate, Tirchy   1268.0
```

Press Esc t

Data Visualization

Pollutant Levels by City:

Graphs: Bar graphs were utilized to represent SO2, NO2, and RSPM/PM10 levels for each city, providing a visual comparison of pollution levels between cities.

> **Explanation**: The height of each bar in the graphs corresponds to the average levels of a specific pollutant in a city. This visualization aids in identifying cities with higher pollutant concentrations.

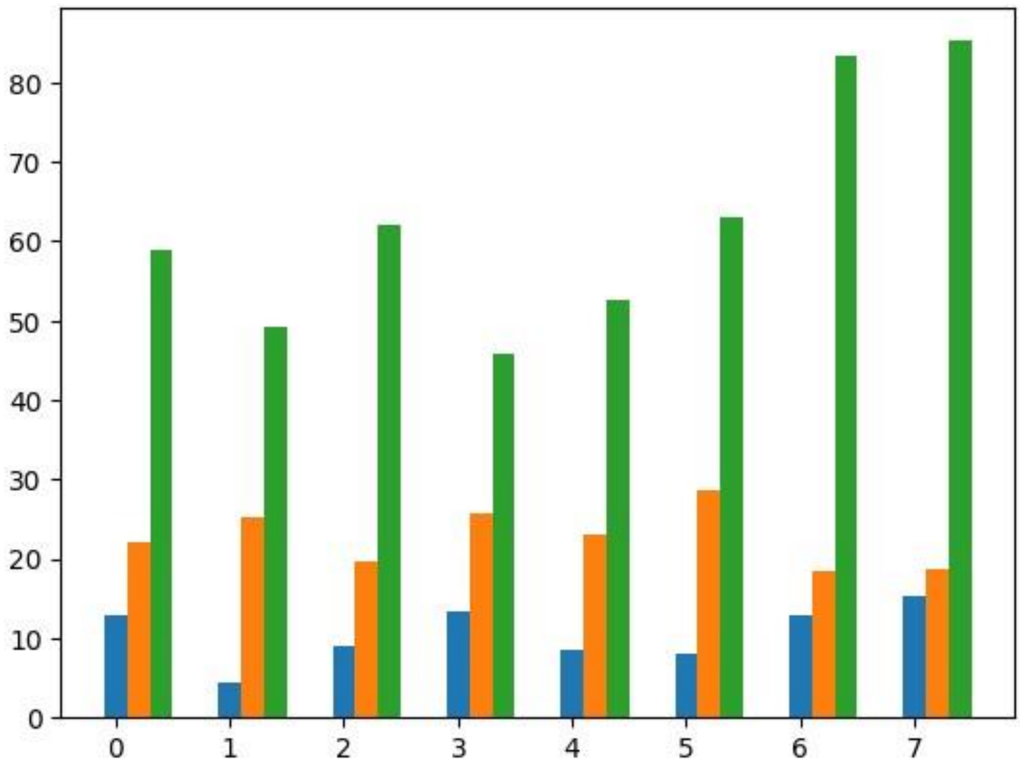
Pollutant Levels by Location:

- **Graphs**: Bar graphs were employed to depict SO2, NO2, and RSPM/PM10 levels for each location within a city. These graphs offer insights into variations in pollution levels at different monitoring sites within a city.

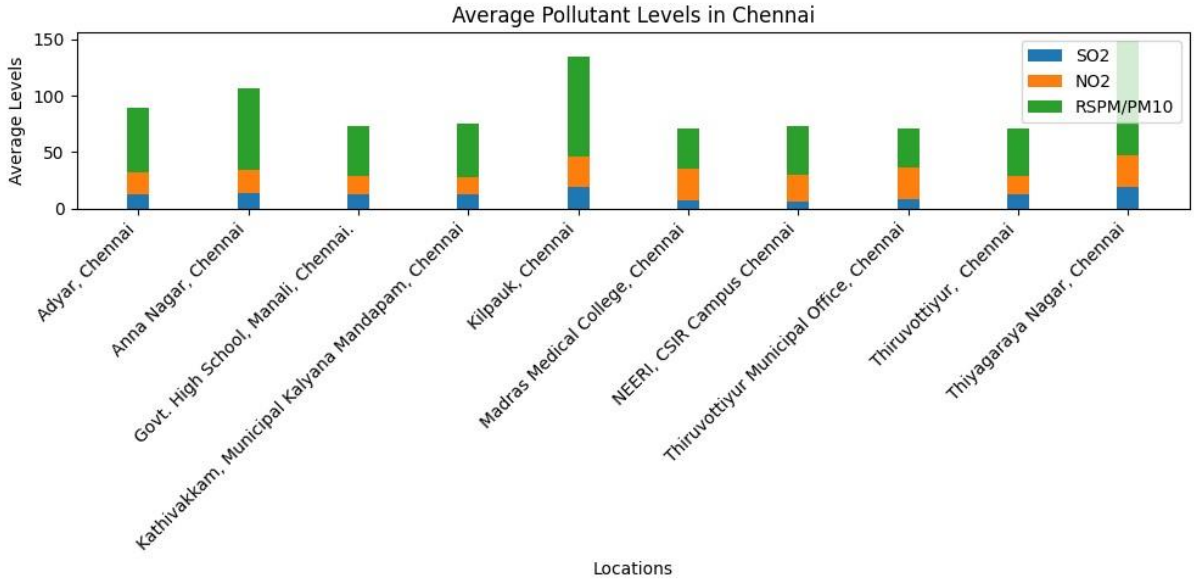
```
cities = city_avg['City']
so2_avg = city_avg['SO2 Average']
no2_avg = city_avg['NO2 Average']
rspm_avg = city_avg['RSPM/PM10 Average']
```

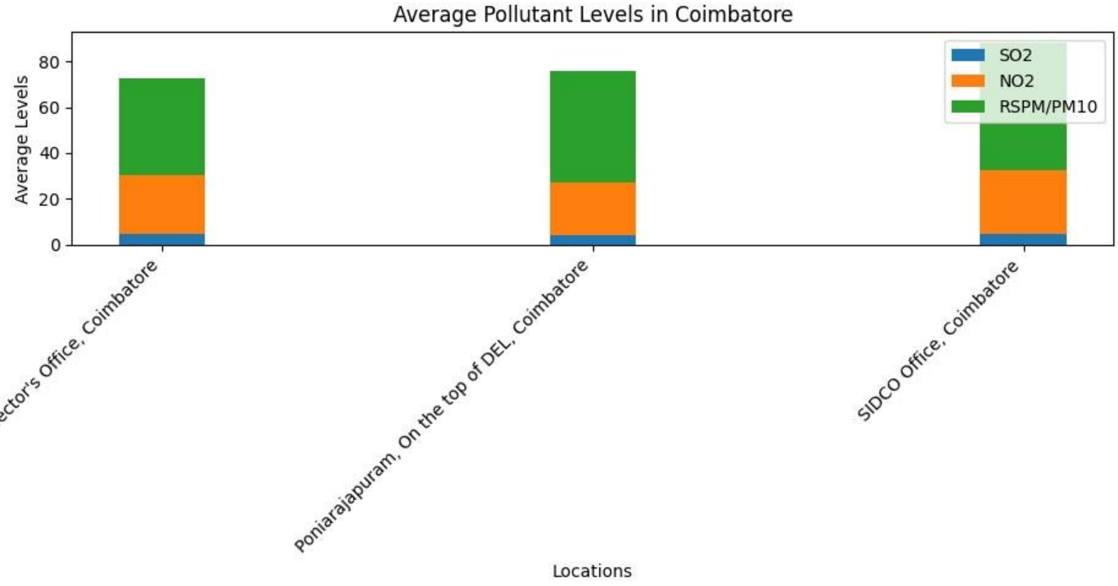
```
bar_width = 0.2

r1 = range(len(cities))
r2 = [x + bar_width for x in r1]
r3 = [x + bar_width for x in r2]
plt.bar(r1, so2_avg, width=bar_width, label='SO2')
plt.bar(r2, no2_avg, width=bar_width, label='NO2')
plt.bar(r3, rspm_avg, width=bar_width, label='RSPM/PM10')
```

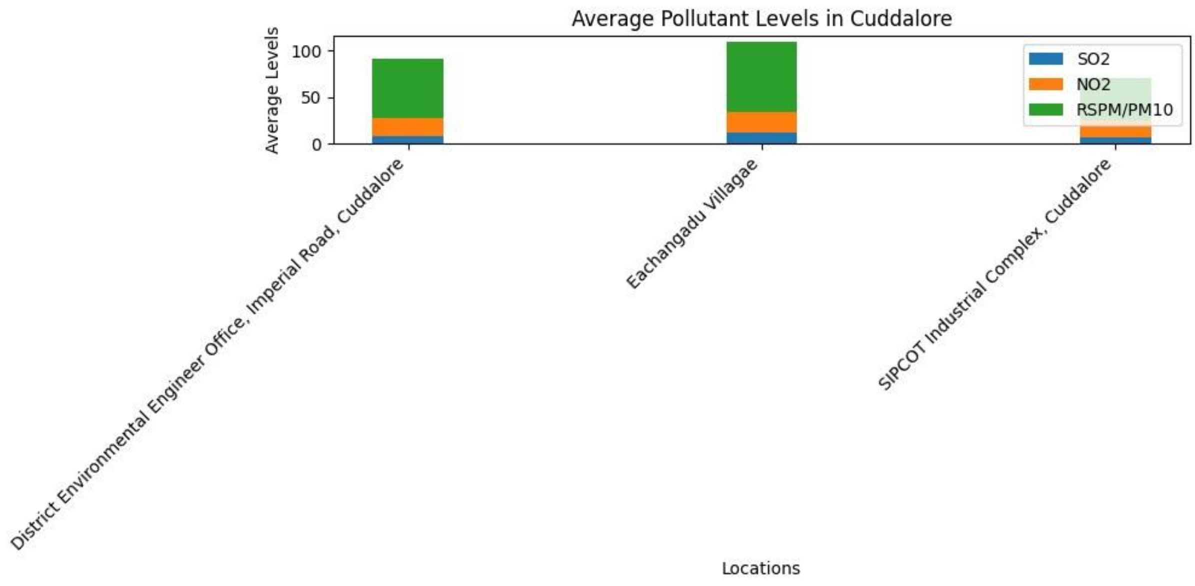
```
plt.xlabel('Cities')
plt.xticks([x + bar_width for x in r1], cities, rotation=90)

plt.ylabel('Average Levels')

plt.title('Average SO2, NO2, and RSPM/PM10 Levels by City')

plt.legend()

plt.tight_layout()
plt.show()
```

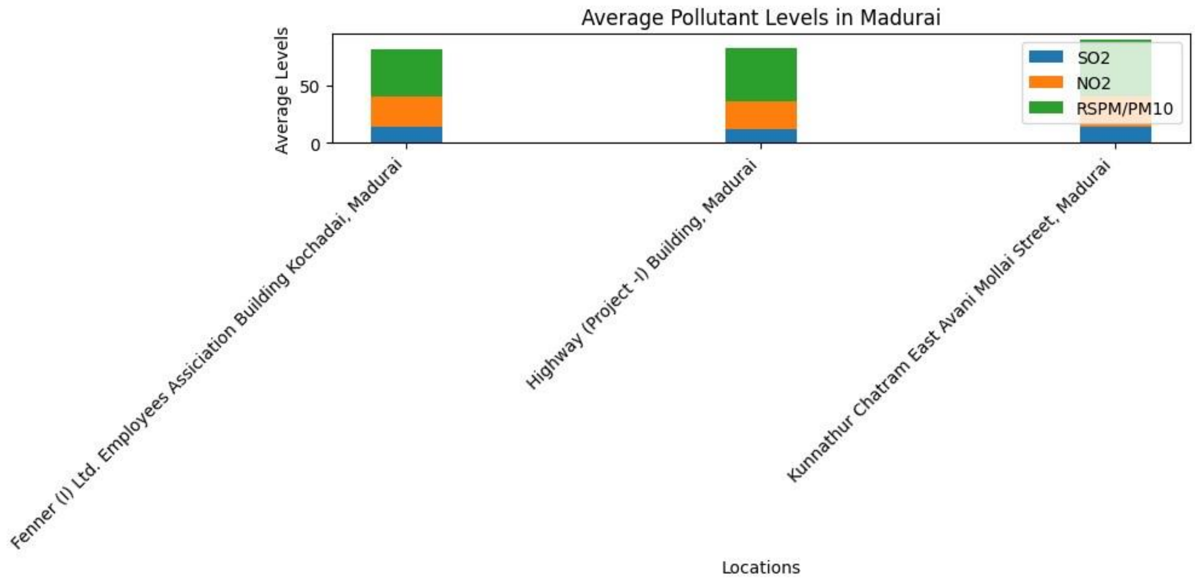


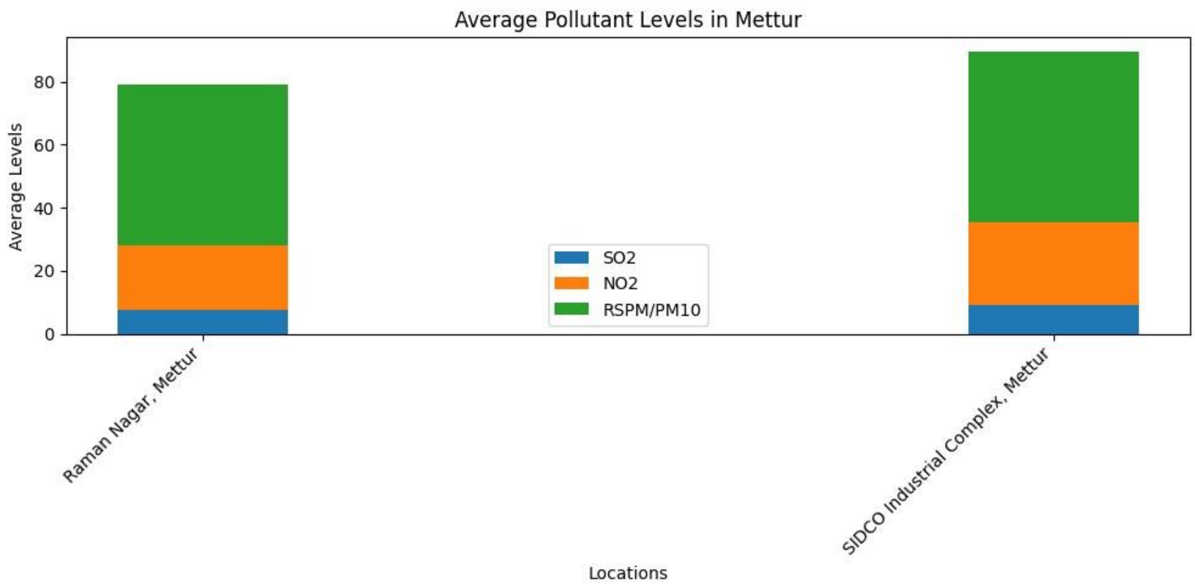
```python
import matplotlib.pyplot as plt

unique_cities = summary['City'].unique()

for city in unique_cities:
    city_data = summary[summary['City'] == city]

    locations = city_data['Location']
    so2_avg = city_data['SO2 Average']
    no2_avg = city_data['NO2 Average']
    rspm_avg = city_data['RSPM/PM10 Average']

    plt.figure(figsize=(10, 5))
    plt.bar(locations, so2_avg, width=0.2, label='SO2')
    plt.bar(locations, no2_avg, width=0.2, label='NO2', bottom=so2_avg)
    plt.bar(locations, rspm_avg, width=0.2, label='RSPM/PM10', bottom=so2_avg + no2_avg)

    plt.xlabel('Locations')
    plt.xticks(rotation=45, ha='right')

    plt.ylabel('Average Levels')

    plt.title(f'Average Pollutant Levels in {city}')

    plt.legend()

    plt.tight_layout()
    plt.show()
```

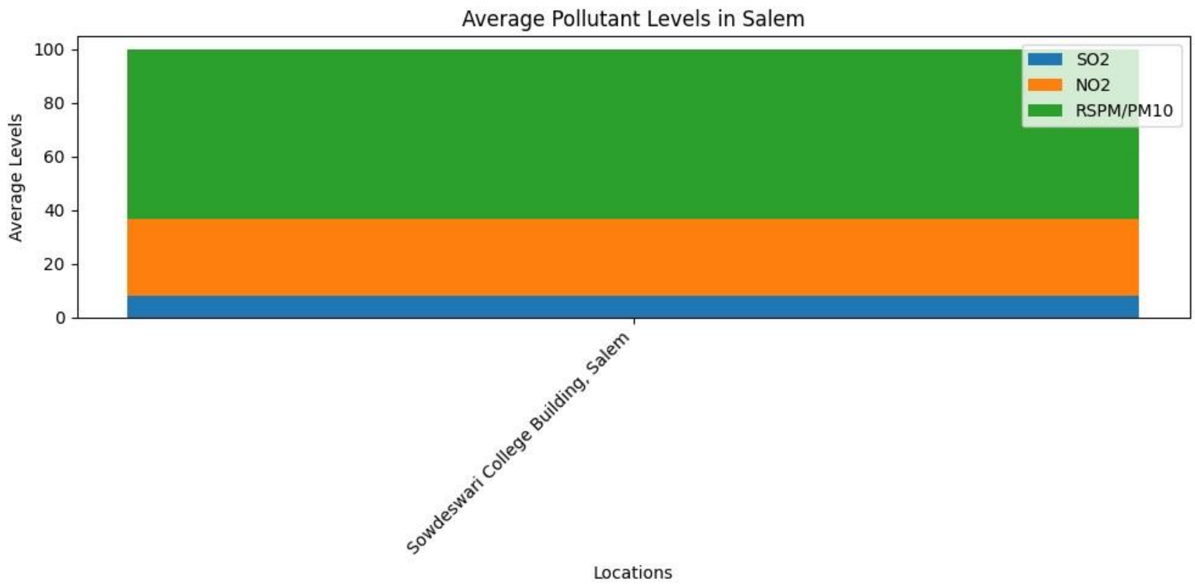


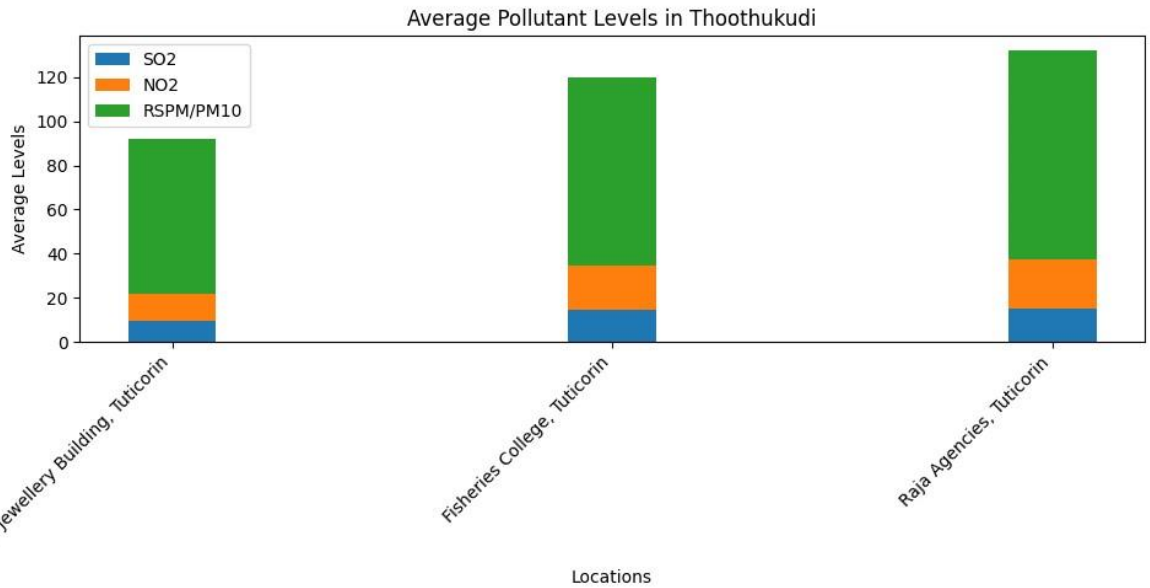


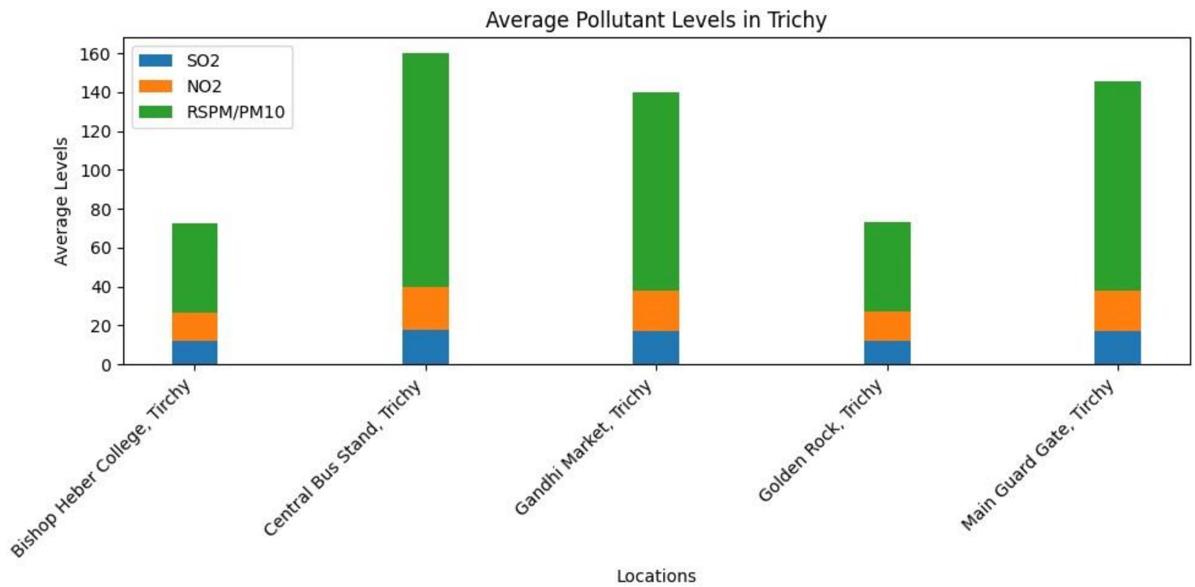

Average Pollutant Levels in Cuddalore
Average Levels
100
50
0
SO2
NO2
RSPM/PM10
District Environmental Engineer Office, Imperial Road, Cuddalore
Eachangadu Villagae
SIPCOT Industrial Complex, Cuddalore
Locations


Average Pollutant Levels in Madurai
Average Levels
50
0
SO2
NO2
RSPM/PM10
Fenner (I) Ltd. Employees Assiciation Building Kochadai, Madurai
Highway (Project -I) Building, Madurai
Kunnathur Chatram East Avani Mollai Street, Madurai
Locations


Average Pollutant Levels in Mettur
Average Levels
80
60
40
20
0
SO2
NO2
RSPM/PM10
Raman Nagar, Mettur
SIDCO Industrial Complex, Mettur
Locations


Average Pollutant Levels in Salem
SO2
NO2
RSPM/PM10
Average Levels
Sowdeswari College Building, Salem
Locations


Average Pollutant Levels in Thoothukudi
SO2
NO2
RSPM/PM10
Average Levels
AVM Jewellery Building, Tuticorin
Fisheries College, Tuticorin
Raja Agencies, Tuticorin
Locations


Average Pollutant Levels in Trichy
SO2
NO2
RSPM/PM10
Average Levels
Bishop Heber College, Tirchy
Central Bus Stand, Trichy
Gandhi Market, Trichy
Golden Rock, Trichy
Main Guard Gate, Tirchy
Locations

Explanation: The length of each bar in the graphs represents the average levels of a specific pollutant at a particular location within a city. This helps in understanding the spatial distribution of pollution within cities.

Phase objective:

In this phase of our air quality analysis project, we continue to explore and visualize the air quality data. The dataset is loaded from the file "modified_transportation_data.csv," and we focus on understanding the average levels of SO2, NO2, and RSPM/PM10 across monitoring stations and city/town/village/area. Additionally, we create visualizations, time-series plots, and correlation matrices to gain insights into air quality trends and relationships.

Data Loading and Preparation:

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

data = pd.read_csv("/content/modified_transportation_data.csv")
```

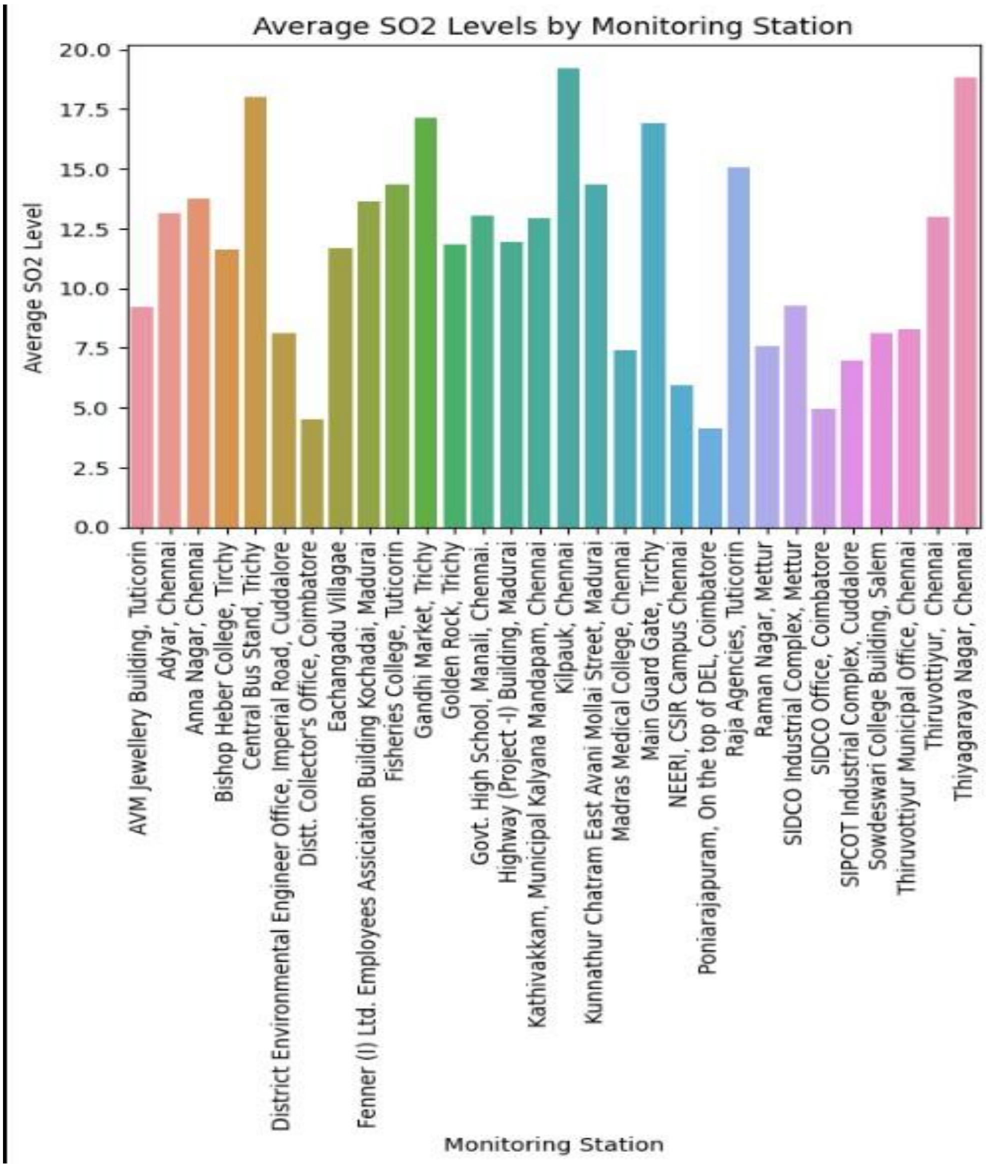
Average Pollution Levels by Monitoring Stations:

We calculated and visualized the average SO2, NO2, and RSPM/PM10 levels across different monitoring stations. The bar plots provide a clear overview of pollution levels by station.

Average SO2 Levels:

```
# Create a bar plot to visualize average SO2 levels by monitoring station
sns.barplot(x=average_levels.index, y=average_levels['SO2'])
plt.xlabel('Monitoring Station')
plt.ylabel('Average SO2 Level')
```

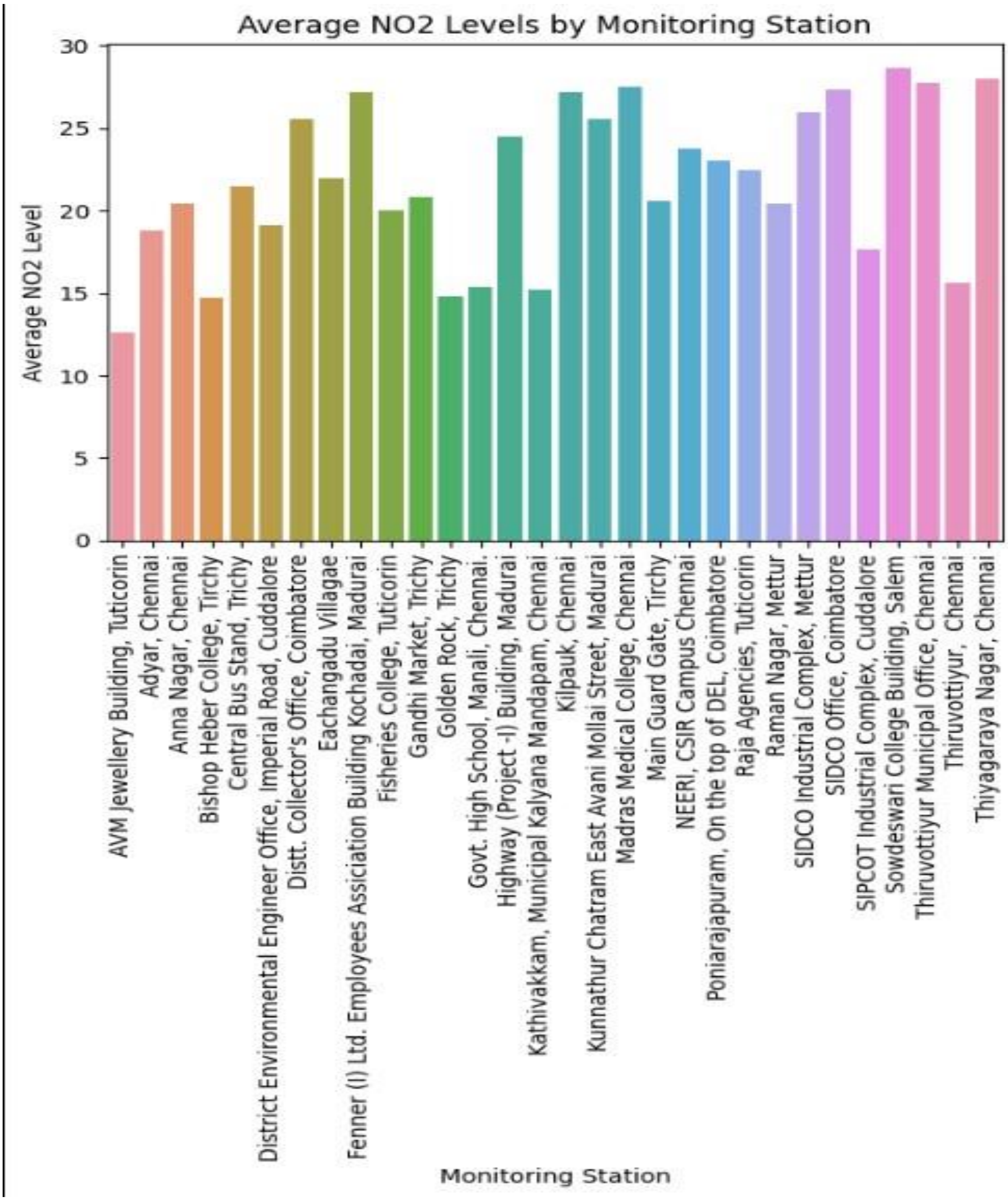
```
plt.title('Average SO2 Levels by Monitoring Station')
plt.xticks(rotation=90)
plt.show()
```



Average NO2 Levels:

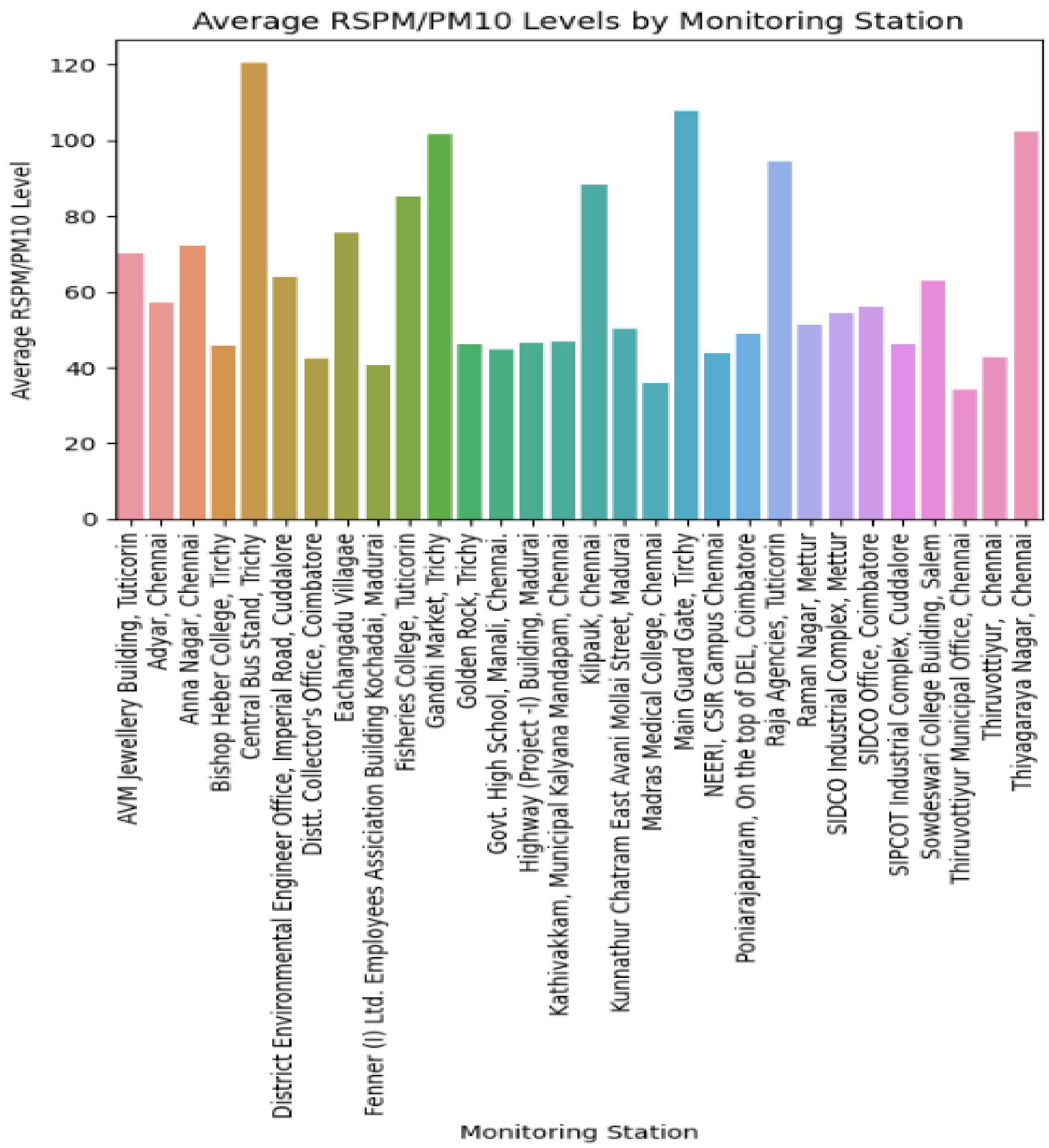
```
# Create a bar plot to visualize average NO2 levels by monitoring station
sns.barplot(x=average_levels.index, y=average_levels['NO2'])
plt.xlabel('Monitoring Station')
plt.ylabel('Average NO2 Level')
```

```
plt.title('Average NO2 Levels by Monitoring Station')
plt.xticks(rotation=90)
plt.show()
```



Average RSPM/PM10 Levels:

```
# Create a bar plot to visualize average RSPM/PM10 levels by monitoring station
sns.barplot(x=average_levels.index, y=average_levels['RSPM/PM10'])
plt.xlabel('Monitoring Station')
plt.ylabel('Average RSPM/PM10 Level')
plt.title('Average RSPM/PM10 Levels by Monitoring Station')
plt.xticks(rotation=90)
plt.show()
```

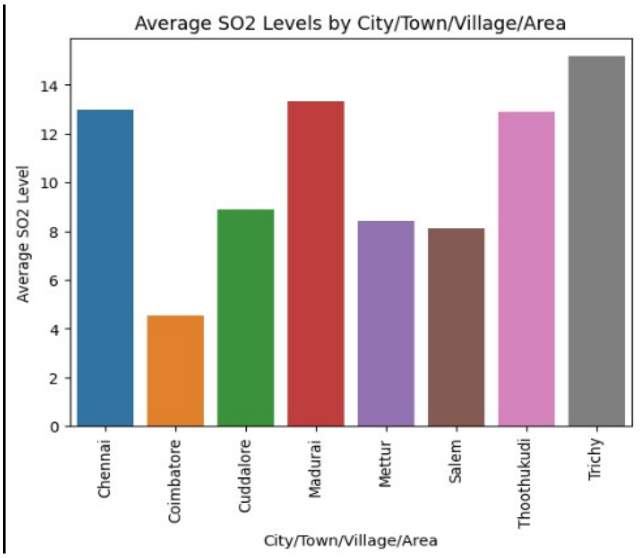


Average Pollution Levels by City/Town/Village/Area:

We calculated and visualized the average SO2, NO2, and RSPM/PM10 levels by city/town/village/area, providing insights into air quality on a larger scale.

Average SO2 Levels:

```
# Create a bar plot to visualize average SO2 levels by city/town/village/area
sns.barplot(x=average_city_levels.index, y=average_city_levels['SO2'])
plt.xlabel('City/Town/Village/Area')
plt.ylabel('Average SO2 Level')
plt.title('Average SO2 Levels by City/Town/Village/Area')
plt.xticks(rotation=90)
plt.show()
```



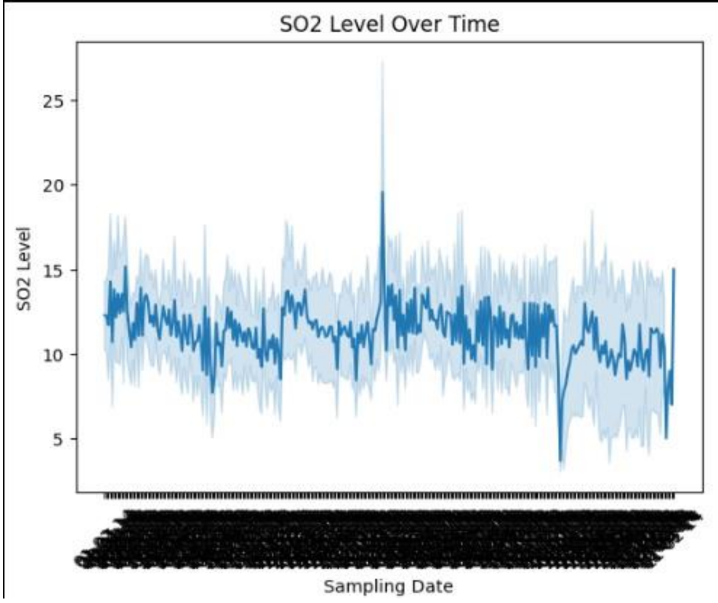
Time-Series Plots:

We created time-series plots to visualize the changes in pollutants levels over time.

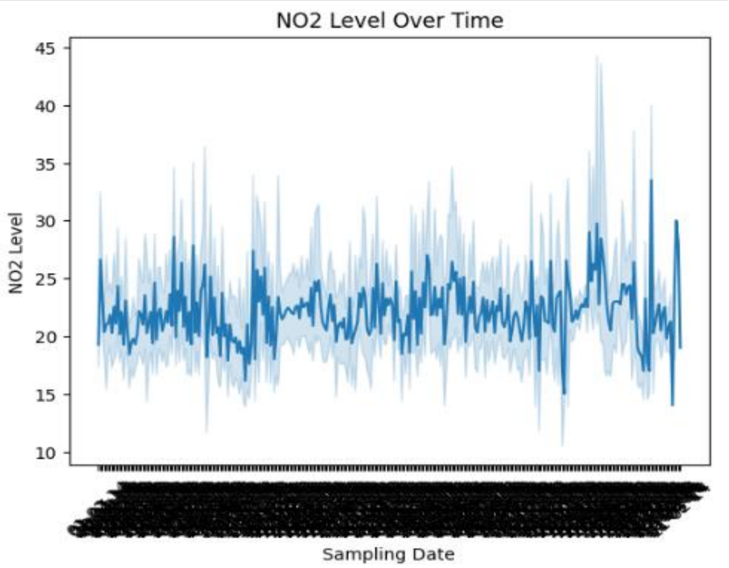
SO2:

```
# Time-series plot for SO2 levels
sns.lineplot(x="Sampling Date", y="SO2", data=data)
plt.xlabel('Sampling Date')
plt.ylabel('SO2 Level')
plt.title('SO2 Level Over Time')
```

```
plt.xticks(rotation=45)
plt.show()
```



NO2:

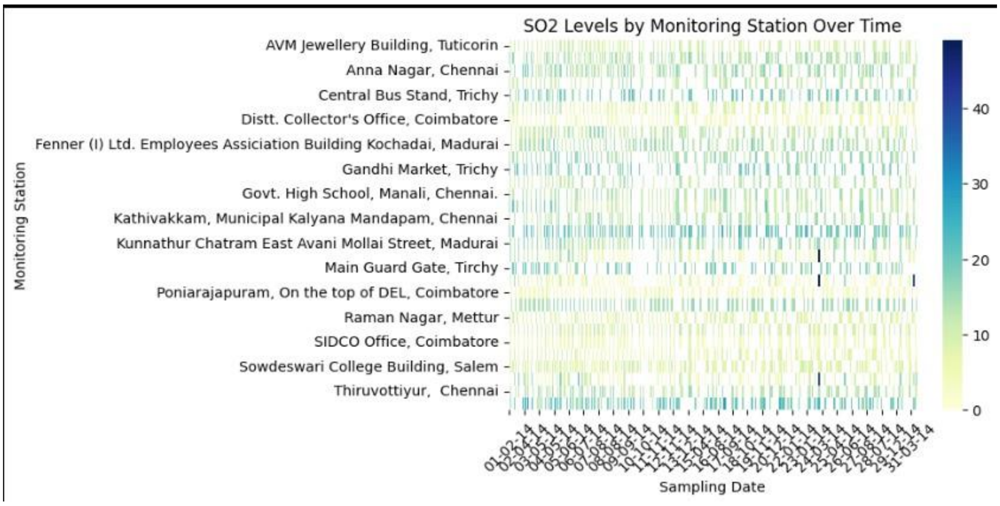


Heatmaps:

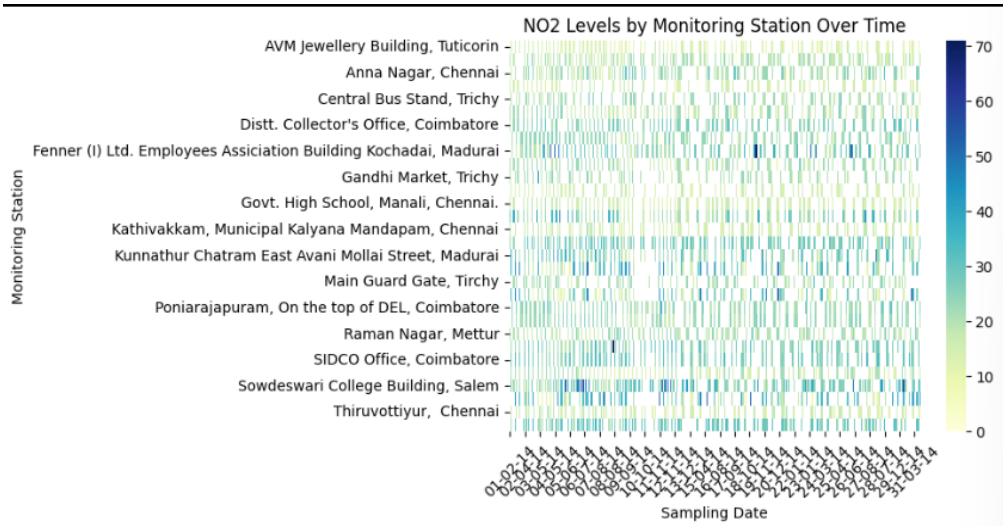
We generated heatmaps to observe variations in pollutants levels by monitoring station over time.

SO2:

```
# Heatmap for SO2 levels by monitoring station
sns.heatmap(data.pivot_table(values='SO2', index='Location of Monitoring Station', columns='Sampling Date'), cmap='YlGnBu')
plt.xlabel('Sampling Date')
plt.ylabel('Monitoring Station')
plt.title('SO2 Levels by Monitoring Station Over Time')
plt.xticks(rotation=45)
plt.show()
```



NO2:



Areas with Highest Pollution Levels:

We sorted and identified areas with the highest average SO2, NO2, and RSPM/PM10 levels.

SO2:

```
# Sort by average SO2 levels
sorted_city_so2 = average_city_levels.sort_values(by='SO2', ascending=False)
print("Areas with highest average SO2 levels:")
print(sorted_city_so2.head(10))
```

```
Areas with highest average SO2 levels:
                             SO2        NO2  RSPM/PM10
City/Town/Village/Area
Trichy                 15.168937  18.542234  85.054496
Madurai                13.319728  25.768707  45.724490
Chennai                12.975000  21.978000  58.998000
Thoothukudi            12.901024  18.385666  83.458904
Cuddalore               8.905405  19.577703  61.881757
Mettur                  8.429268  23.185366  52.721951
Salem                   8.114504  28.664122  62.954198
Coimbatore              4.525597  25.238908  49.217241
```

NO2:

```
# Sort by average NO2 levels
sorted_city_no2 = average_city_levels.sort_values(by='NO2', ascending=False)
print("Areas with highest average NO2 levels:")
print(sorted_city_no2.head(10))
```

```
Areas with highest average NO2 levels:
                             SO2        NO2  RSPM/PM10
City/Town/Village/Area
Salem                   8.114504  28.664122  62.954198
Madurai                13.319728  25.768707  45.724490
Coimbatore              4.525597  25.238908  49.217241
Mettur                  8.429268  23.185366  52.721951
Chennai                12.975000  21.978000  58.998000
Cuddalore               8.905405  19.577703  61.881757
Trichy                 15.168937  18.542234  85.054496
Thoothukudi            12.901024  18.385666  83.458904
```

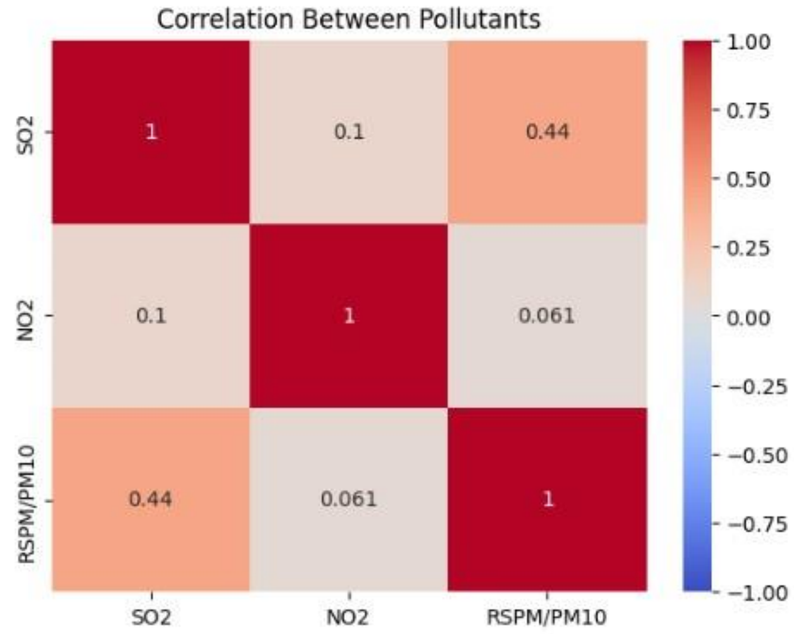
Correlation Analysis:

We calculated the correlation between pollutants and visualized the results using a heatmap.

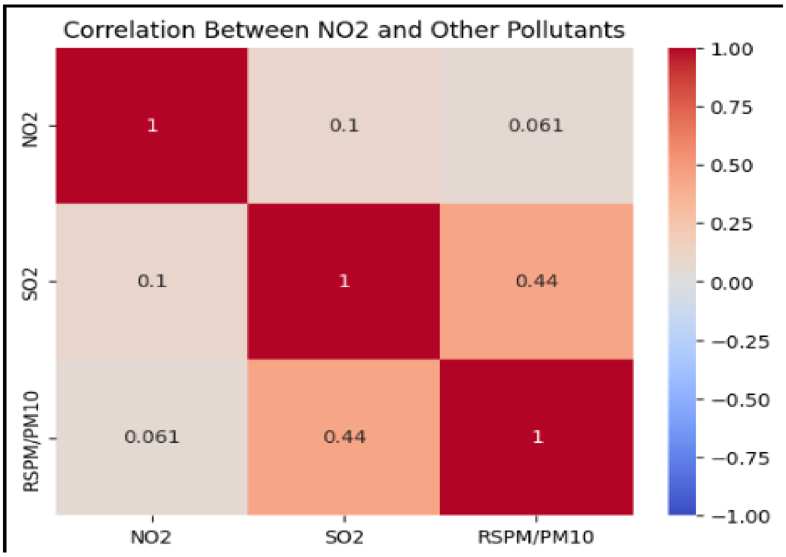
SO2:

```
# Create similar analyses for NO2 and RSPM/PM10.
correlation_matrix = data[['SO2', 'NO2', 'RSPM/PM10']].corr()
```

```
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Between Pollutants')
plt.show()
```



NO2:



Conclusion:

In this phase of the air quality analysis project, we have conducted extensive data analysis, visualization, and correlation studies. These insights into pollution levels across monitoring stations, city/town/village/areas, and their correlations provide a solid foundation for understanding air quality trends and patterns. Future work may involve predictive modeling and more advanced analytics to address air quality challenges comprehensively.

Team Members:

1. Hematharshini E – 2021115041

Email - hemae2512@gmail.com

2. Sabitha S – 2021115087

Email - sabitha.suresh15@gmail.com

3. Sandhya Shankar – 2021115090

Email - sandhya.shankar.2002@gmail.com

4. Sanmitha V.S – 2021115092

Email - sanmithasadhishkumar@gmail.com

5. Akash P – 2021115314

Email - akashpanneer2004@gmail.com