**Air Quality Analysis and Prediction in Tamil Nadu**

|  |  |
| --- | --- |
| **Date** | **28-10-2023** |
| **Team ID** | **3886** |
| **Project Name** | **Air Quality Analysis and Predicion in Tamil Nadu** |
| **Team Members** | **Santosh M Sriranjani M Sripoornimadevi T. S**  **Trisha J V** |

#### Problem Statement:

**Overview:**

The problem statement revolves around improving the accuracy of an existing predictive model. The existing model may have limitations in terms of prediction accuracy, especially when dealing with complex and dynamic datasets. Incorporate machine learning algorithms to enhance prediction accuracy. Optimize data collection and preprocessing techniques. Perform advanced feature engineering to extract valuable insights from the data. Select and train models that can adapt to various scenarios. Leverage geographic analysis to uncover location-specific patterns and trends.

#### INTRODUCTION:

Air quality analysis and predictions in Tamil Nadu involve the systematic assessment of ambient air pollution levels and forecasting future trends to safeguard public health and the environment. This process combines data collection from air quality monitoring stations, satellite imagery, and meteorological data to generate insights into pollutant concentrations and their potential impact. These predictions aid in making informed decisions for pollution control measures, urban planning, and public health initiatives. By analyzing historical data and employing advanced modeling techniques, authorities can mitigate the adverse effects of air pollution, reduce health risks, and promote sustainable development throughout the state of Tamil Nadu.

### Analysis Approach:

The project starts by importing necessary Python libraries for data analysis, including NumPy, Pandas, Matplotlib, Seaborn, and scikit-learn.

The dataset, which is presumably air quality data for Tamil Nadu in 2014, is loaded using Pandas from a CSV file.

The dataset is explored by examining its structure, summary statistics, and column names to get an initial understanding of the data.

### Data Exploration:

dataset.info() provides information about the dataset, including data types and missing values.

dataset.describe() presents summary statistics for each numerical column. dataset.columns lists the column names in the dataset.

### Data Visualization:

The project uses various visualization techniques to gain insights from the data:

Histogram for SO2 Levels: Visualizes the distribution of SO2 levels in Tamil Nadu.

Scatter Plot of NO2 vs. RSPM/PM10: Examines the relationship between NO2 and RSPM/PM10 levels.

Bar Chart for State-wise SO2 Levels: Displays the average SO2 levels for different states in Tamil Nadu.

Pairplot and Histograms: Further explores the relationships between variables and distributions of different pollutants.

Correlation Matrix: Calculates and visualizes the correlation between air quality parameters.

### Calculating Averages:

Average levels of SO2, NO2, and RSPM/PM10 are calculated for different cities or areas within Tamil Nadu. The average levels provide insights into pollution variations across the region.

### Creating Visualizations:

Bar plots are used to visualize the average levels of SO2, NO2, and RSPM/PM10 in different areas. This provides a clear visual representation of pollution levels across various locations.

#### Insights into Air Pollution Trends:

The project uses visualizations to illustrate the distribution and trends in air quality parameters.

The correlation matrix helps identify any relationships between pollutants.

The bar charts for different areas show variations in pollution levels, helping to pinpoint areas with higher pollution.

**Code implementation:**

# IMPORTING LIBRARIES

import numpy as np import pandas as pd import os

import matplotlib.pyplot as plt

%matplotlib inline import seaborn as sns import warnings

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

LOADING THE DATASET

dataset= pd.read\_csv("C:\\Users\\SANTH\\Downloads\\ cpcb\_dly\_aq\_tamil\_nadu-2014.csv")

# DATA EXPLORATION:

0

1

2

3

4

... 2874

2875

Location of Monitoring Station Kathivakkam, Municipal Kalyana Mandapam, Chennai Kathivakkam, Municipal Kalyana Mandapam, Chennai Kathivakkam, Municipal Kalyana Mandapam, Chennai Kathivakkam, Municipal Kalyana Mandapam, Chennai Kathivakkam, Municipal Kalyana Mandapam, Chennai

...

Central Bus Stand, Trichy Central Bus Stand, Trichy

\

dataset

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Stn Code | Sampling Date |  | State | City/Town/Village/Area | \ |
| 0 | 38 | 01-02-2014 | Tamil | Nadu | Chennai |  |
| 1 | 38 | 01-07-2014 | Tamil | Nadu | Chennai |  |
| 2 | 38 | 21-01-2014 | Tamil | Nadu | Chennai |  |
| 3 | 38 | 23-01-2014 | Tamil | Nadu | Chennai |  |
| 4 | 38 | 28-01-2014 | Tamil | Nadu | Chennai |  |
| ... | ... | ... |  | ... | ... |  |
| 2874 | 773 | 12-03-2014 | Tamil | Nadu | Trichy |  |
| 2875 | 773 | 12-10-2014 | Tamil | Nadu | Trichy |  |
| 2876 | 773 | 17-12-2014 | Tamil | Nadu | Trichy |  |
| 2877 | 773 | 24-12-2014 | Tamil | Nadu | Trichy |  |
| 2878 | 773 | 31-12-2014 | Tamil | Nadu | Trichy |  |

2876 Central Bus Stand, Trichy

2877 Central Bus Stand, Trichy

2878 Central Bus Stand, Trichy

Agency \

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 0 | Tamilnadu | State | Pollution | Control | Board |
| 1 | Tamilnadu | State | Pollution | Control | Board |
| 2 | Tamilnadu | State | Pollution | Control | Board |
| 3 | Tamilnadu | State | Pollution | Control | Board |
| 4 | Tamilnadu | State | Pollution | Control | Board |
| ... |  |  |  |  | ... |
| 2874 | Tamilnadu | State | Pollution | Control | Board |
| 2875 | Tamilnadu | State | Pollution | Control | Board |
| 2876 | Tamilnadu | State | Pollution | Control | Board |
| 2877 | Tamilnadu | State | Pollution | Control | Board |
| 2878 | Tamilnadu | State | Pollution | Control | Board |
| Type of Location SO | | | | | |

2 NO2 RSPM/PM10 PM

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 2.5 |  | | | | | |
| 0 | Industrial | Area | 11.0 | 17.0 | 55.0 |  |
| NaN |  |  |  |  |  |  |
| 1 | Industrial | Area | 13.0 | 17.0 | 45.0 |  |
| NaN |  |  |  |  |  |  |
| 2 | Industrial | Area | 12.0 | 18.0 | 50.0 |  |
| NaN |  |  |  |  |  |  |
| 3 | Industrial | Area | 15.0 | 16.0 | 46.0 |  |
| NaN |  |  |  |  |  |  |
| 4 | Industrial | Area | 13.0 | 14.0 | 42.0 |  |
| NaN |  |  |  |  |  |  |
| ... |  | ... | ... | ... | ... | .. |
| . |  |  |  |  |  |  |

2874 Residential, Rural and other Areas 15.0 18.0 102.0

NaN

2875 Residential, Rural and other Areas 12.0 14.0 91.0

NaN

2876 Residential, Rural and other Areas 19.0 22.0 100.0

NaN

2877 Residential, Rural and other Areas 15.0 17.0 95.0

NaN

2878 Residential, Rural and other Areas 14.0 16.0 94.0

NaN

[2879 rows x 11 columns]

dataset.info()

<class 'pandas.core.frame.DataFrame'> RangeIndex: 2879 entries, 0 to 2878

Data columns (total 11 columns):

# Column Non-Null Count Dtype

1. Stn Code 2879 non-null int64
2. Sampling Date 2879 non-null object
3. State 2879 non-null object
4. City/Town/Village/Area 2879 non-null object
5. Location of Monitoring Station 2879 non-null object
6. Agency 2879 non-null object
7. Type of Location 2879 non-null object
8. SO2 2868 non-null float64
9. NO2 2866 non-null float64
10. RSPM/PM10 2875 non-null float64
11. PM 2.5 0 non-null float64 dtypes: float64(4), int64(1), object(6)

memory usage: 247.5+ KB dataset.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 |

dataset.columns

Index(['Stn Code', 'Sampling Date', 'State', 'City/Town/Village/Area', 'Location of Monitoring Station', 'Agency', 'Type of Location',

'SO2',

'NO2', 'RSPM/PM10', 'PM 2.5'],

dtype='object')

# DATA VISUALIZATION:

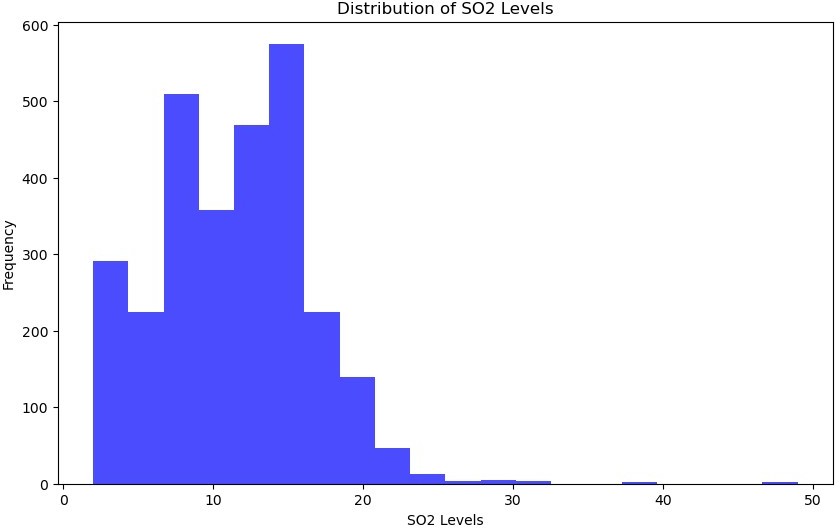
1.Histogram for SO2 levels

plt.figure(figsize=(10, 6))

plt.hist(dataset['SO2'], bins=20, color='blue', alpha=0.7) plt.title('Distribution of SO2 Levels')

plt.xlabel('SO2 Levels')

plt.ylabel('Frequency') plt.show()

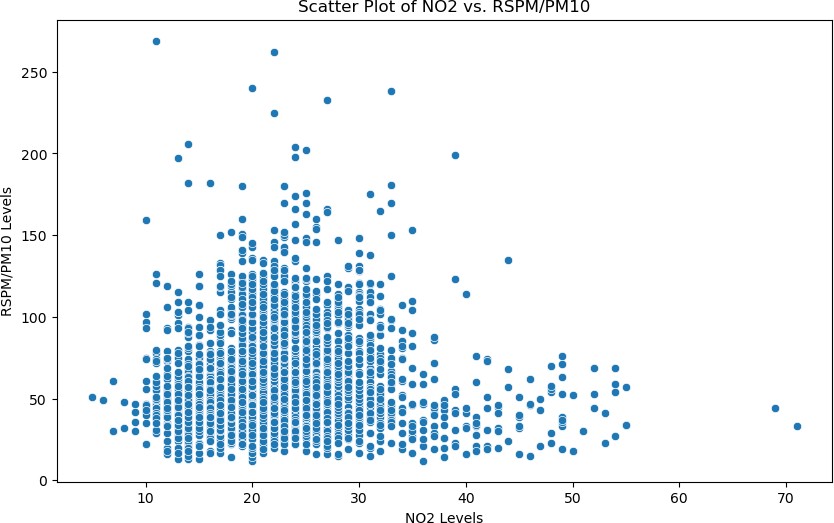


# 1.Scatter plot of NO2 vs. RSPM/PM10

plt.figure(figsize=(10, 6))

sns.scatterplot(x='NO2', y='RSPM/PM10', data=dataset) plt.title('Scatter Plot of NO2 vs. RSPM/PM10') plt.xlabel('NO2 Levels')

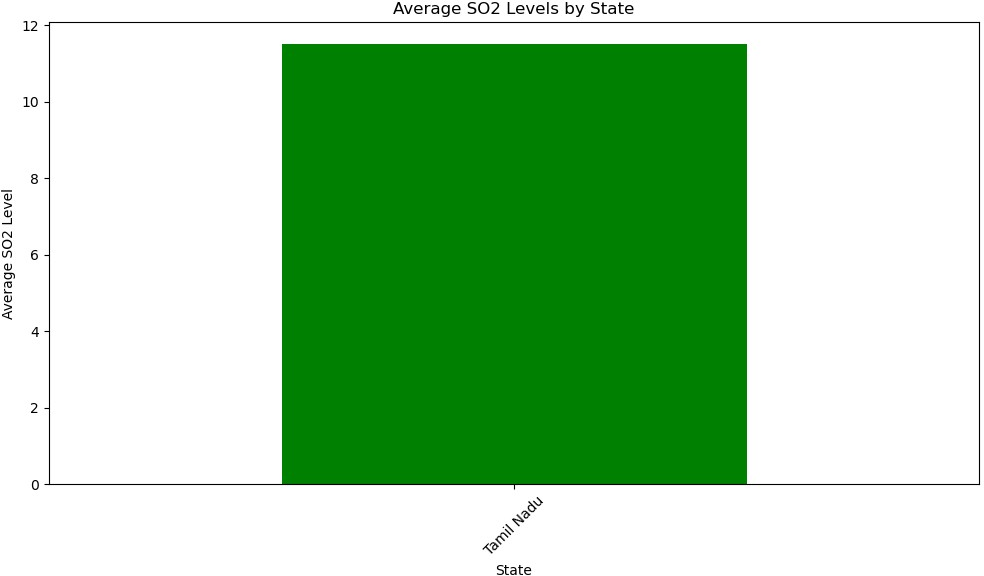
plt.ylabel('RSPM/PM10 Levels') plt.show()



2.Bar chart for State-wise SO2 levels

statewise\_so2 = dataset.groupby('State') ['SO2'].mean().sort\_values(ascending=False) plt.figure(figsize=(12, 6)) statewise\_so2.plot(kind='bar', color='green') plt.title('Average SO2 Levels by State') plt.xlabel('State')

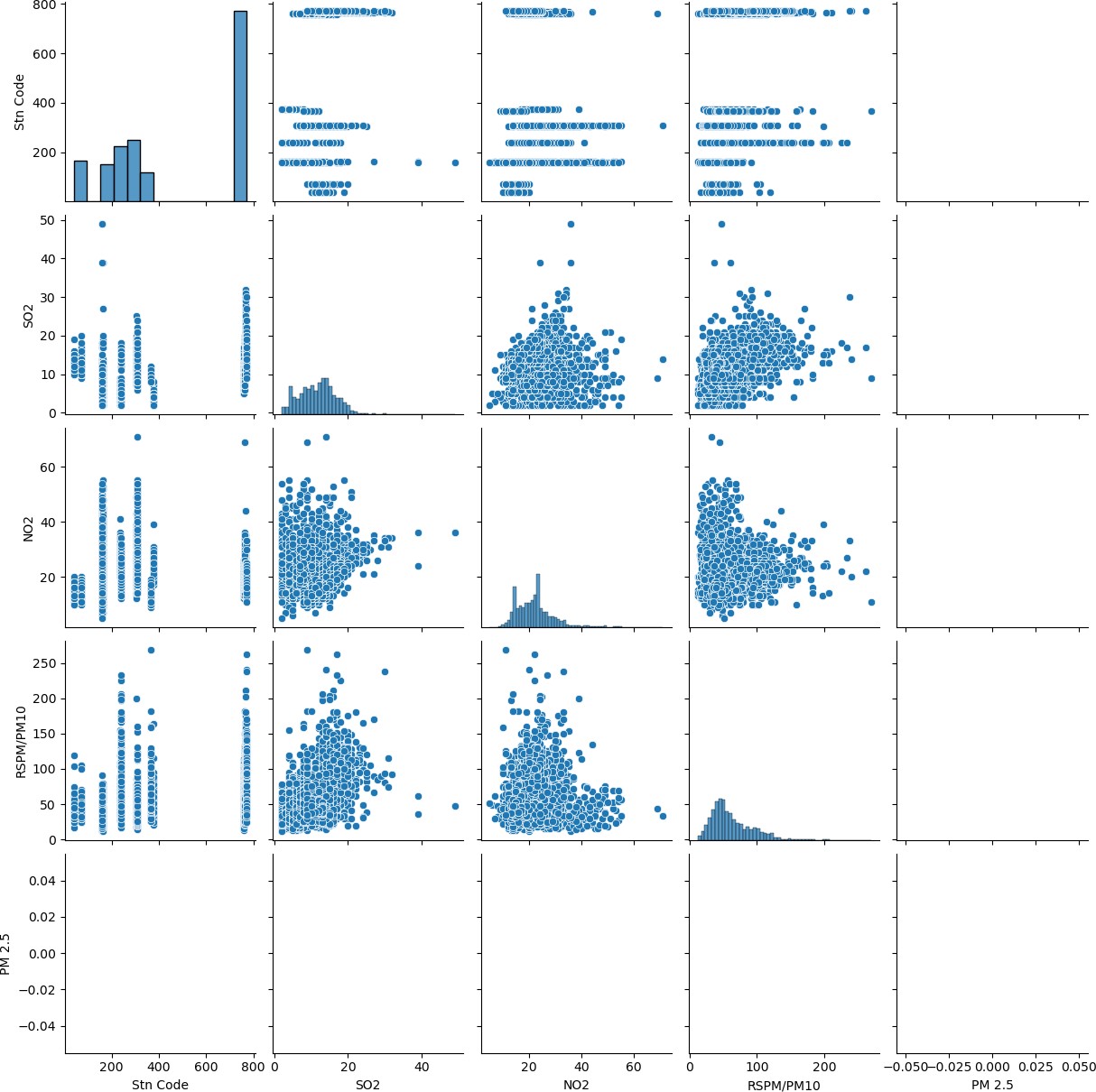
plt.ylabel('Average SO2 Level') plt.xticks(rotation=45) plt.show()



plt.figure(figsize=(12,8)) sns.pairplot(dataset)

<seaborn.axisgrid.PairGrid at 0x207303588e0>

<Figure size 1200x800 with 0 Axes>



dataset.hist(figsize=(10,8))

array([[<AxesSubplot:title={'center':'Stn Code'}>,

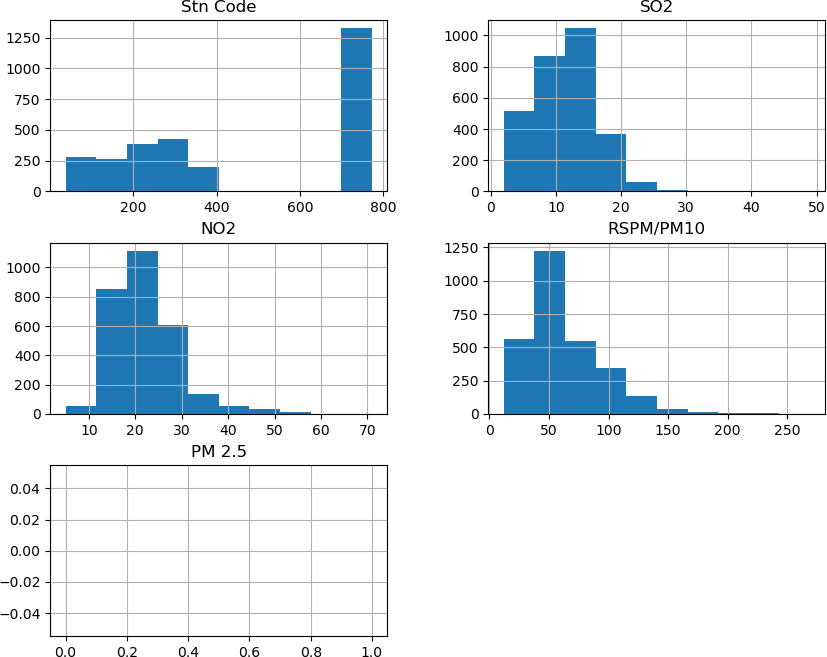
<AxesSubplot:title={'center':'SO2'}>],

[<AxesSubplot:title={'center':'NO2'}>,

<AxesSubplot:title={'center':'RSPM/PM10'}>],

[<AxesSubplot:title={'center':'PM 2.5'}>, <AxesSubplot:>]],

dtype=object)



# Visualising Correlation:

dataset.corr()

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Stn Code | SO2 | NO2 | RSPM/PM10 | PM 2.5 |
| Stn Code 1.000000 | 0.263537 | -0.043257 | 0.336190 | NaN |
| SO2 0.263537 | 1.000000 | 0.078246 | 0.445152 | NaN |
| NO2 -0.043257 | 0.078246 | 1.000000 | 0.068277 | NaN |
| RSPM/PM10 0.336190 | 0.445152 | 0.068277 | 1.000000 | NaN |
| PM 2.5 NaN | NaN | NaN | NaN | NaN |

calculating Averages:

so2\_by\_area = dataset.groupby('City/Town/Village/Area') ['SO2'].mean().sort\_values(ascending=False)

no2\_by\_area = dataset.groupby('City/Town/Village/Area') ['NO2'].mean().sort\_values(ascending=False)

rspm\_pm10\_by\_area = dataset.groupby('City/Town/Village/Area') ['RSPM/PM10'].mean().sort\_values(ascending=False) print("Average SO2 levels by City/Town/Village/Area:") print(so2\_by\_area)

print("\nAverage NO2 levels by City/Town/Village/Area:") print(no2\_by\_area)

print("\nAverage RSPM/PM10 levels by City/Town/Village/Area:") print(rspm\_pm10\_by\_area)

Average SO2 levels by City/Town/Village/Area: City/Town/Village/Area

|  |  |
| --- | --- |
| Trichy | 15.293956 |
| Madurai | 13.319728 |
| Chennai | 13.014042 |
| Thoothukudi | 12.989691 |
| Cuddalore | 8.965986 |
| Mettur | 8.429268 |
| Salem | 8.114504 |
| Coimbatore | 4.541096 |

Name: SO2, dtype: float64

Average NO2 levels by City/Town/Village/Area: City/Town/Village/Area

|  |  |
| --- | --- |
| Salem | 28.664122 |
| Madurai | 25.768707 |
| Coimbatore | 25.325342 |
| Mettur | 23.185366 |
| Chennai | 22.088442 |
| Cuddalore | 19.710884 |
| Trichy | 18.695055 |
| Thoothukudi | 18.512027 |

Name: NO2, dtype: float64

Average RSPM/PM10 levels by City/Town/Village/Area: City/Town/Village/Area

|  |  |
| --- | --- |
| Trichy | 85.054496 |
| Thoothukudi | 83.458904 |
| Salem | 62.954198 |
| Cuddalore | 61.881757 |
| Chennai | 58.998000 |
| Mettur | 52.721951 |
| Coimbatore | 49.217241 |
| Madurai | 45.724490 |

Name: RSPM/PM10, dtype: float64

# creating visualization:

import matplotlib.pyplot as plt import seaborn as sns

so2\_by\_area = dataset.groupby('City/Town/Village/Area') ['SO2'].mean().sort\_values(ascending=False)

no2\_by\_area = dataset.groupby('City/Town/Village/Area') ['NO2'].mean().sort\_values(ascending=False)

rspm\_pm10\_by\_area = dataset.groupby('City/Town/Village/Area') ['RSPM/PM10'].mean().sort\_values(ascending=False)

fig, axes = plt.subplots(3, 1, figsize=(10, 15))

sns.barplot(x=so2\_by\_area.values, y=so2\_by\_area.index, ax=axes[0], color='blue')

axes[0].set\_title('Average SO2 Levels by Area') axes[0].set\_xlabel('Average SO2 Level') axes[0].set\_ylabel('Area')

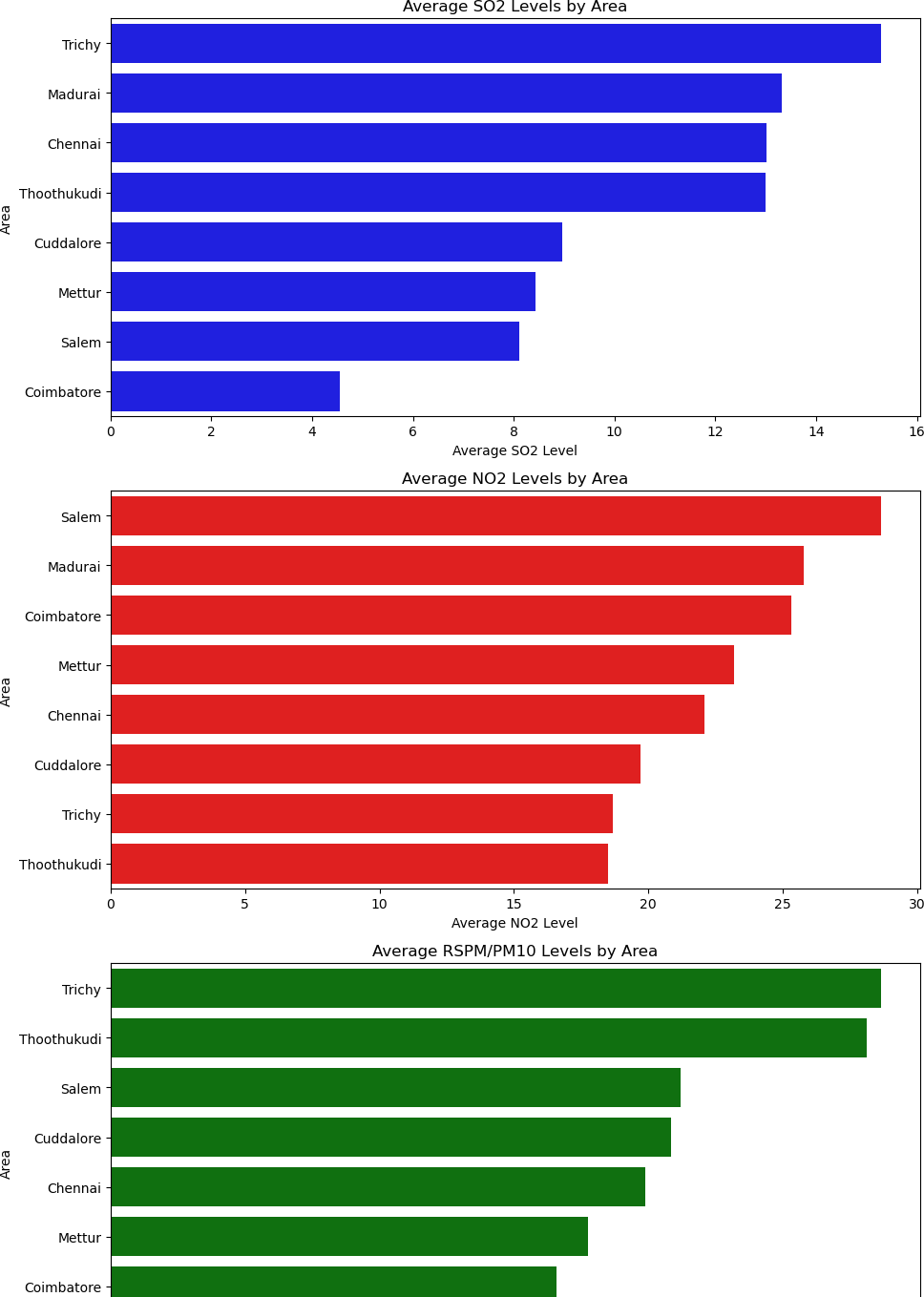
sns.barplot(x=no2\_by\_area.values, y=no2\_by\_area.index, ax=axes[1], color='red')

axes[1].set\_title('Average NO2 Levels by Area') axes[1].set\_xlabel('Average NO2 Level') axes[1].set\_ylabel('Area')

sns.barplot(x=rspm\_pm10\_by\_area.values, y=rspm\_pm10\_by\_area.index, ax=axes[2], color='green')

axes[2].set\_title('Average RSPM/PM10 Levels by Area') axes[2].set\_xlabel('Average RSPM/PM10 Level') axes[2].set\_ylabel('Area')

plt.tight\_layout() plt.show()



## Insights into air pollution:

* The histogram of SO2 levels indicates the distribution of this pollutant, helping to identify the range of SO2 levels recorded.
* The scatter plot of NO2 vs. RSPM/PM10 can reveal any potential correlations or patterns between these two pollutants.
* State-wise SO2 level comparisons show which areas within Tamil Nadu experience higher average SO2 pollution.
* The correlation matrix helps in understanding the relationships between different pollutants, which can be crucial for identifying potential sources of pollution.
* Average pollutant levels by area provide a clear picture of pollution disparities within Tamil Nadu, helping to pinpoint areas that might require more attention

in terms of air quality management.

The combination of data exploration, visualization, and statistical analysis in

this code offers a comprehensive overview of air pollution trends and pollution levels in Tamil Nadu, which can be valuable for policymakers and researchers working to address air quality issues in the region.

## Conclusion:

The analysis provides an overview of air quality in Tamil Nadu for the year 2014.It shows the distribution of SO2 levels and the relationship between NO2 and RSPM/PM10 levels.The bar chart highlighting the average SO2 levels by state identifies regions with higher pollution levels, which can be used for further investigation and targeted interventions.The analysis of average levels by City/Town/Village/Area allows for a localized understanding of air quality, which can be valuable for local authorities and residents.The correlation

analysis can help in identifying any significant relationships between air quality parameters, which can inform policies and actions to improve air quality.