**IBM NAAN MUDHALVAN - PHASE 4**

**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.

**Objectives:**

The core objectives of this project are as follows:

* **Data Analysis and Visualization**: Utilize IBM Cognos and Python to perform comprehensive data analysis and visualization to understand historical trends in air quality in Tamil Nadu. By exploring this data, we aim to identify patterns and correlations that can inform policy decisions and interventions.
* **Identify Pollution Hotspots**: Using spatial and temporal analysis, we will identify regions and times with elevated pollution levels. This information is crucial for targeting mitigation efforts and resource allocation.
* **Predictive Modeling**: Develop a predictive model using Python and relevant libraries. This model will estimate RSPM/PM10 levels based on the concentrations of SO2 and NO2. Accurate predictions will help authorities take proactive measures to address air quality issues.

**Dataset Link:**[**https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014**](https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014)

**Importing the necessary python libraries:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

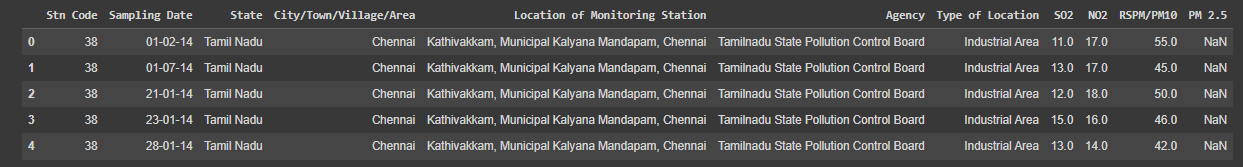
import seaborn as sns

**Importing the csv file:**

data = pd.read\_csv('cpcb\_dly\_aq\_tamil\_nadu-2014.csv')

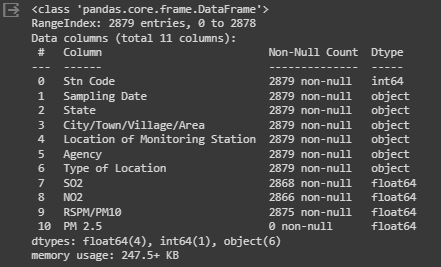
data.head()

**Output:**



data.info()

**Output:**



**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')

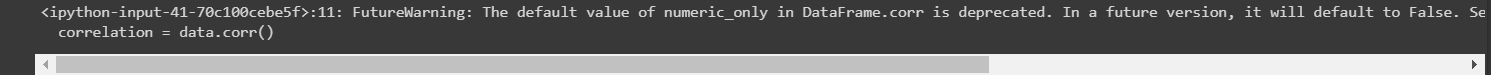
# Summary statistics of the numerical columns

summary\_stats = data.describe()

# Correlation between variables

correlation = data.corr()

**Output:**



**Visualizations:**

**Distribution of SO2, NO2, RSP****M/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)

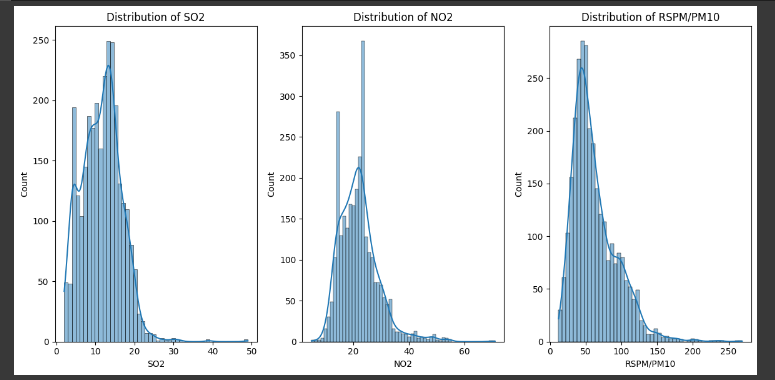
sns.histplot(data['RSPM/PM10'].dropna(), kde=True)

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)

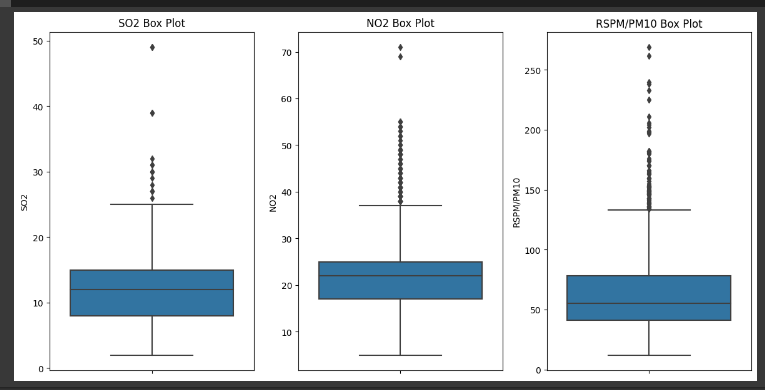
sns.boxplot(data=data, y='RSPM/PM10')

plt.title('RSPM/PM10 Box Plot')

plt.tight\_layout()

plt.show()

**Output:**



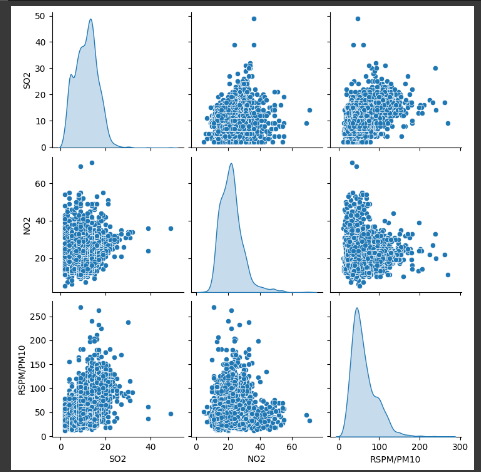
**Pairplot to visualize relationships between variables:**

# Pairplot to visualize relationships between variables

sns.pairplot(data=data, vars=['SO2', 'NO2', 'RSPM/PM10'], diag\_kind='kde')

plt.show()

**Output:**



**State-wise air quality analysis:**

# State-wise air quality analysis

statewise\_avg = data.groupby('State')[['SO2', 'NO2', 'RSPM/PM10']].mean().reset\_index()

plt.figure(figsize=(12, 6))

sns.barplot(data=statewise\_avg, x='State', y='SO2', color='skyblue', label='SO2')

sns.barplot(data=statewise\_avg, x='State', y='NO2', color='lightcoral', label='NO2')

sns.barplot(data=statewise\_avg, x='State', y='RSPM/PM10', color='lightgreen', label='RSPM/PM10')

plt.title('State-wise Air Quality Analysis')

plt.xlabel('State')

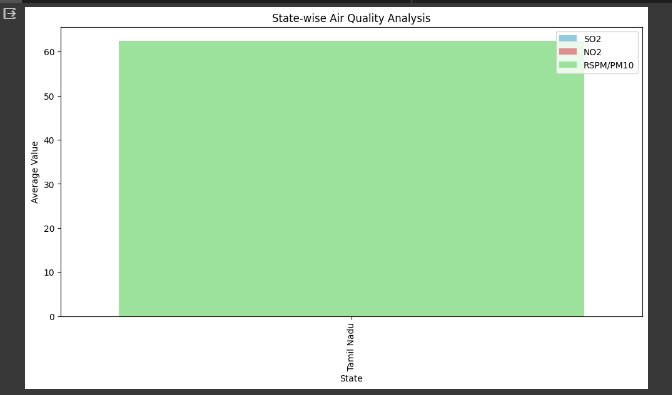
plt.ylabel('Average Value')

plt.legend()

plt.xticks(rotation=90)

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')

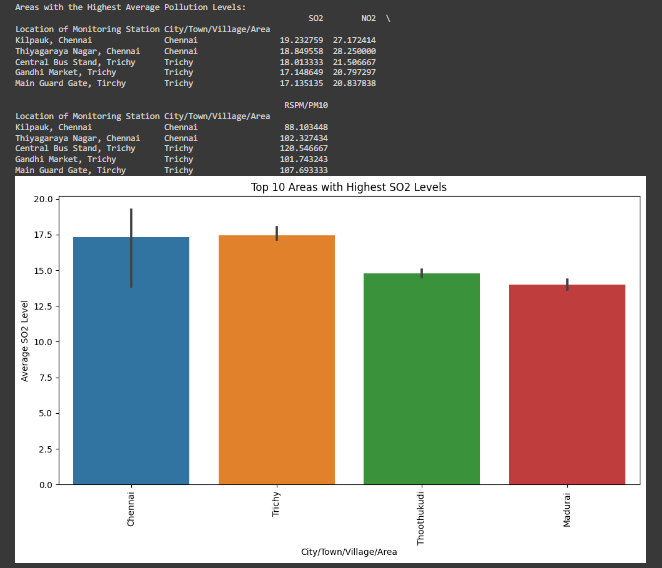
plt.xlabel('City/Town/Village/Area')

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')

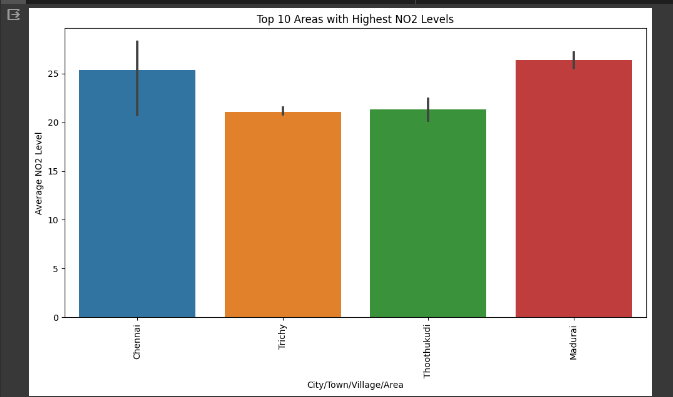
plt.xlabel('City/Town/Village/Area')

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')

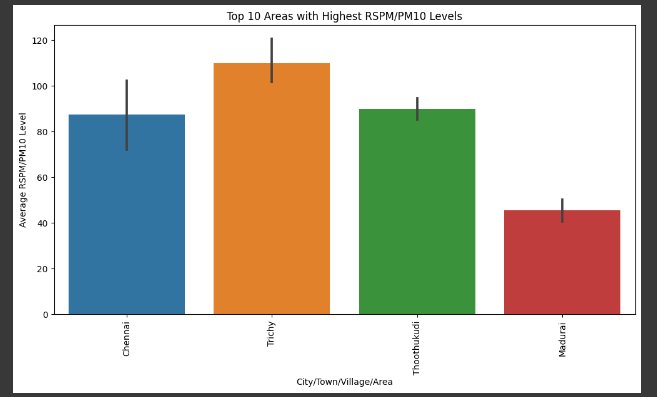
plt.xlabel('City/Town/Village/Area')

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')

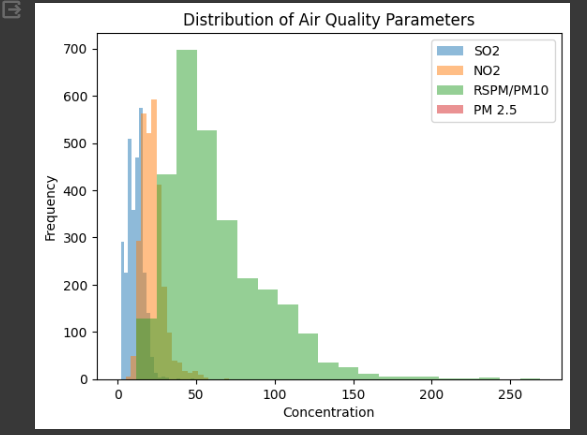
plt.ylabel('Frequency')

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

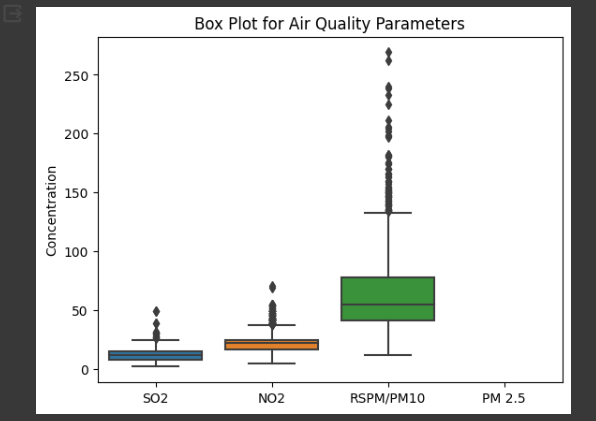
sns.boxplot(data=data[['SO2', 'NO2', 'RSPM/PM10', 'PM 2.5']], orient='v')

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')

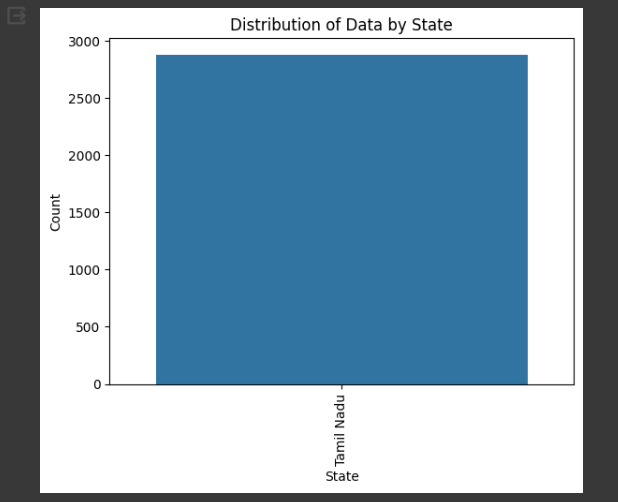
plt.ylabel('Count')

plt.title('Distribution of Data by State')

plt.xticks(rotation=90)

plt.show()

**Output:**



#Correlation Heatmap to explore relationships between numerical variables (SO2, NO2, RSPM/PM10, PM2.5):

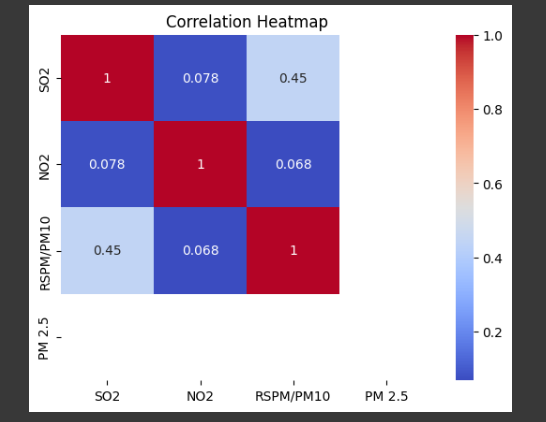
correlation\_matrix = data[['SO2', 'NO2', 'RSPM/PM10', 'PM 2.5']].corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Heatmap')

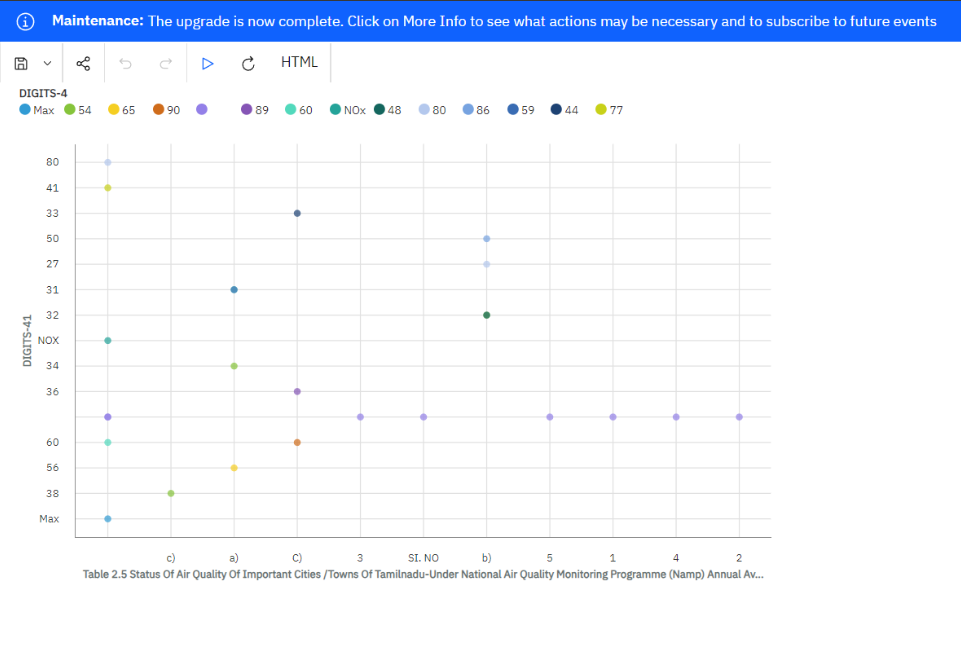
plt.show()

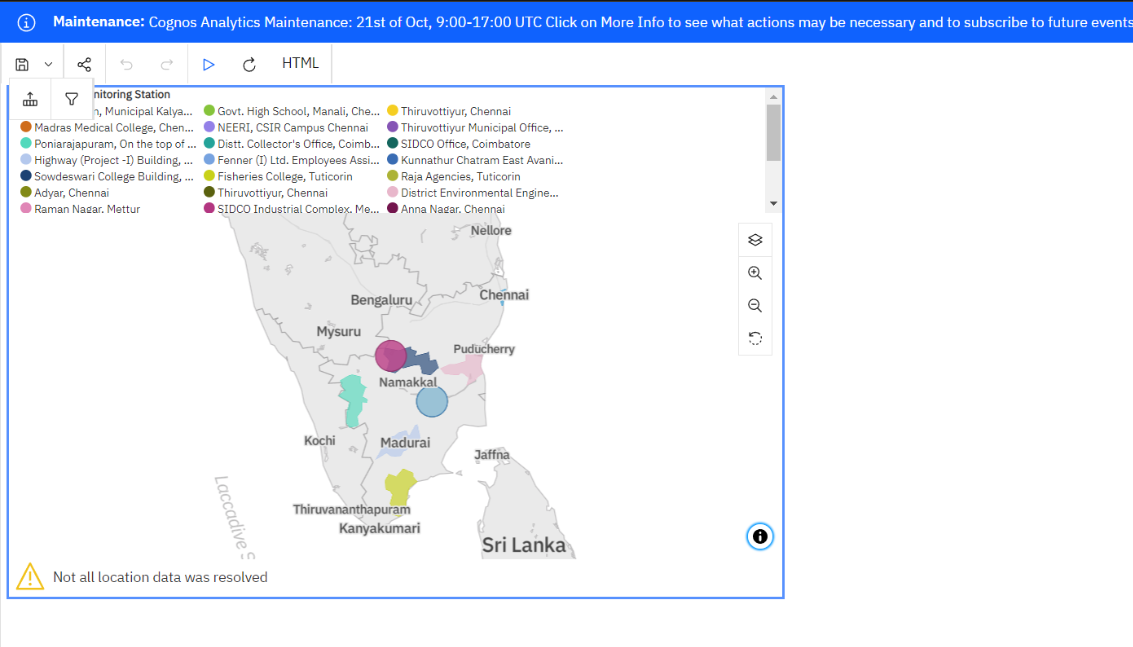
**Output:**



**Visualization using IBM Cognos:**







**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.