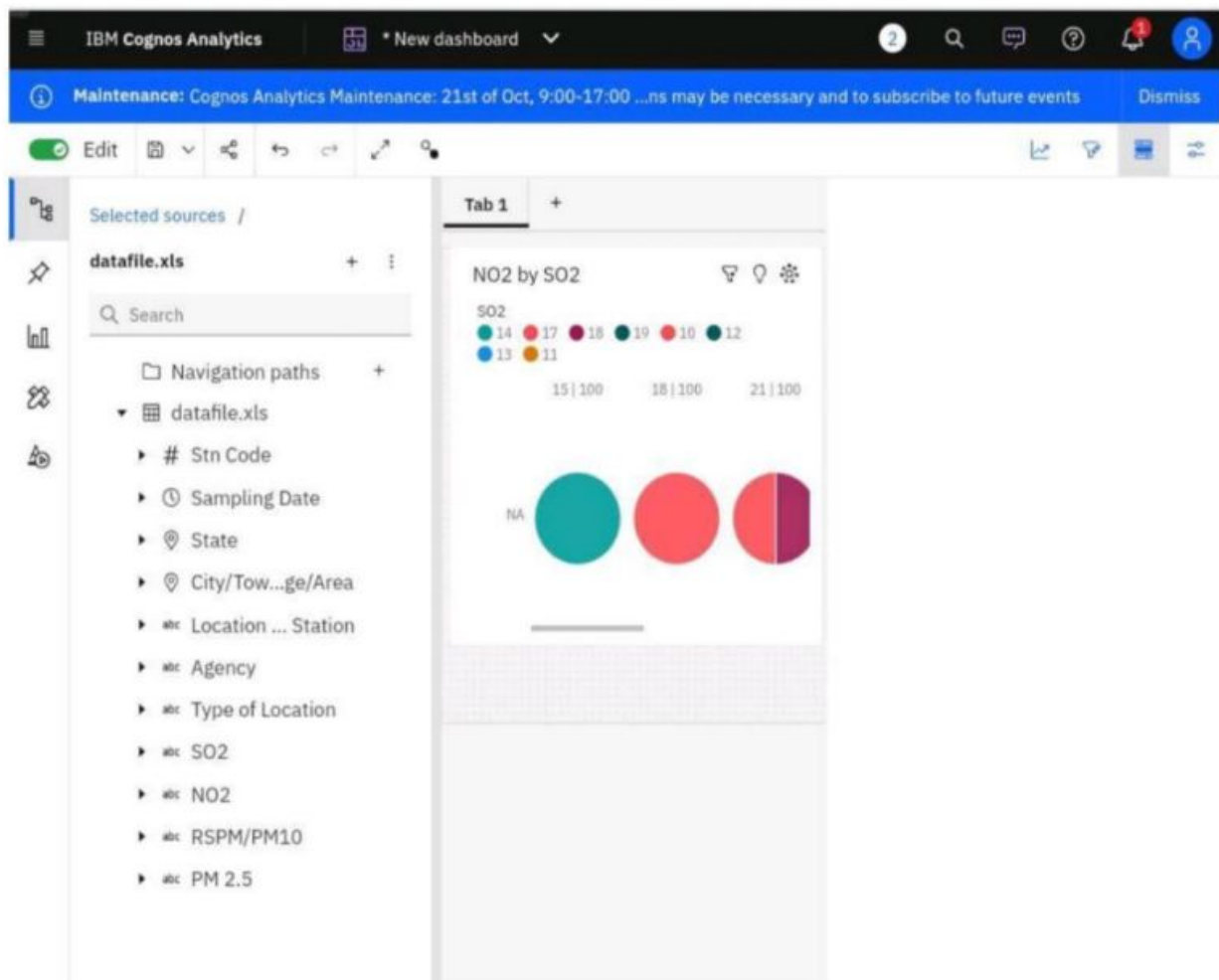


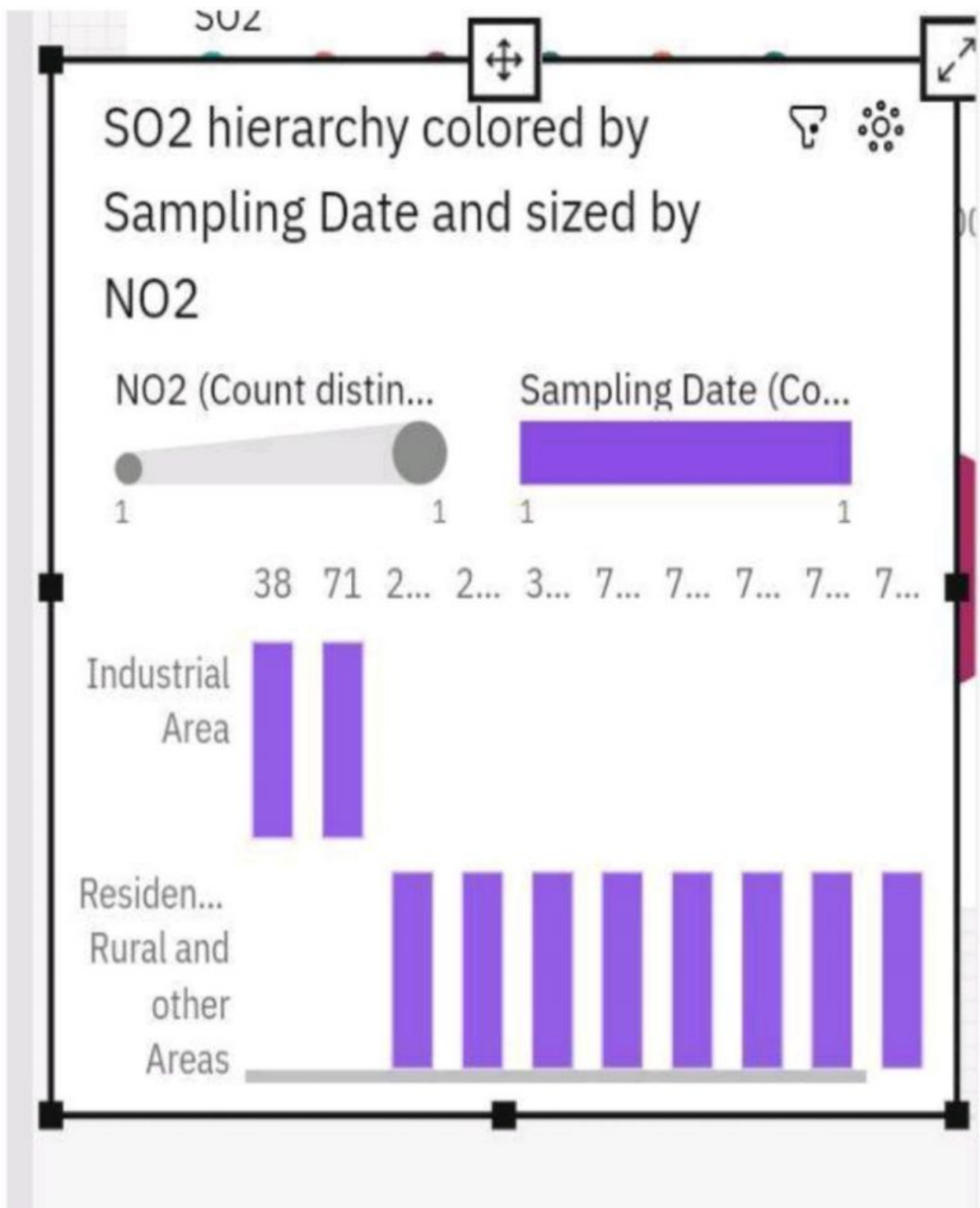
Air quality analysis in tamilnadu

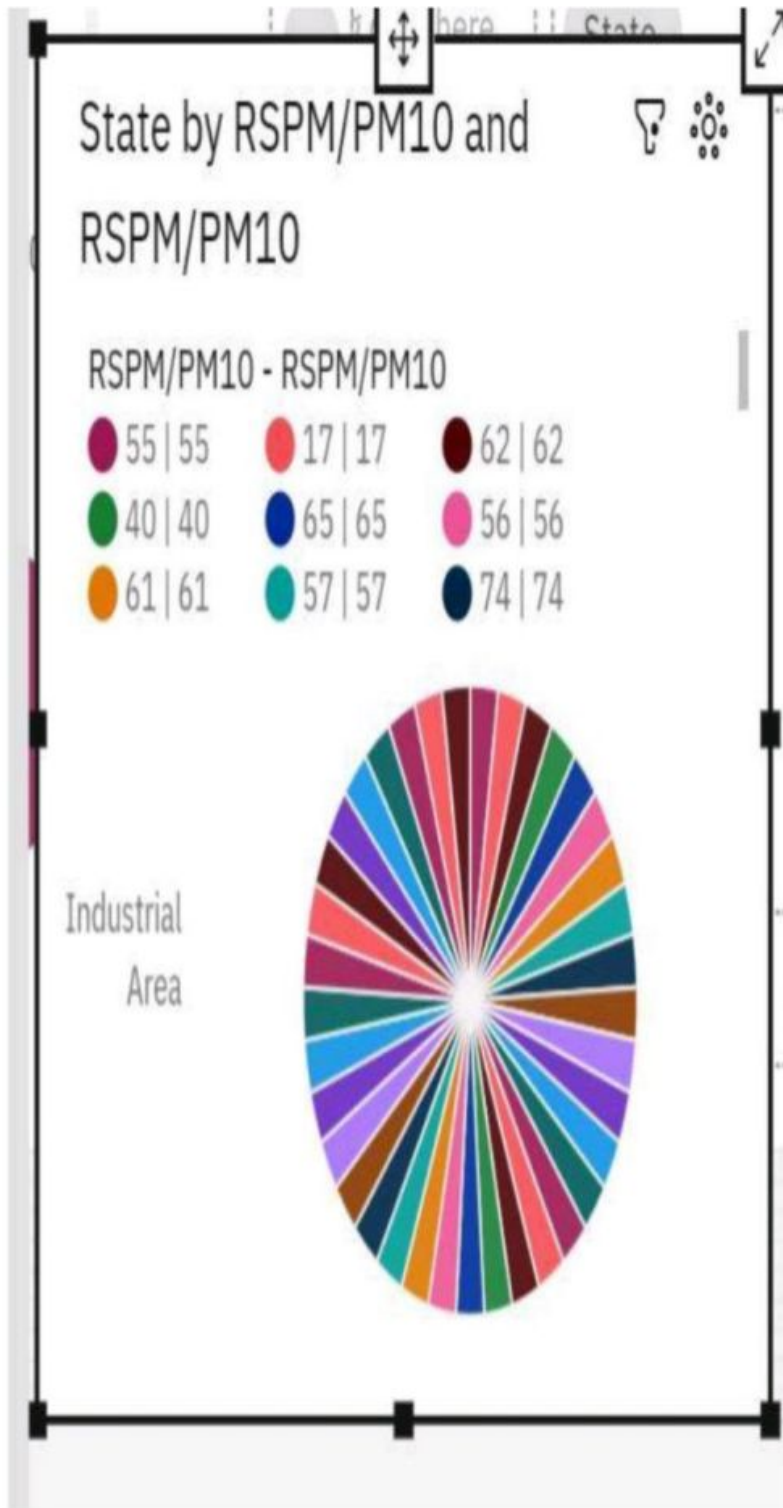
To perform the air quality analysis and to create visualizations

The MATLAB code helps preprocess the sensor data, provides functions to classify the data, and provides functions for visualizing the processed air quality data.

Calculated average SO₂,NO₂ and RSPM/PM₁₀ levels across different monitoring stations,cities and areas are as follows..







RSPM/PM10

Click or

Size*

State

Repeat (column)

Click or

Repeat (row)

Type of Locatio

Tab 1

City/Town/Village/Area and RSPM/PM10

City/Town/Village/...	RSPM/PM10
Chennai	104
	107
	115
	119
	123
	166
	17
	21

Columns*

- City/Town/Village/Area
- RSPM/PM10

Local filters

S02

Includes: 13

Creating visualizations using visualization libraries (example matplotlib and seaborn)

To perform air quality analysis and create visualizations using data visualization libraries such as Matplotlib and Seaborn, you will need to follow these steps:

1. Import the necessary libraries:

Python

Import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns

2. Load the air quality data into a Pandas DataFrame:

Python

```
Data = pd.read_csv('air_quality_data.csv')
```

3. Explore the data to understand its structure and available columns:

Python

```
Print(data.head()) # Display the first few rows of the DataFrame
```

```
Print(data.info()) # Get information about the DataFrame
```

4. Clean and preprocess the data if necessary (e.g., handle missing values, convert data types).
5. Perform analysis on the air quality data using various statistical methods and calculations.
6. Create visualizations using Matplotlib and Seaborn to gain insights from the data. Here are a few examples:
 - Line Plot: Visualize the trend of a specific air quality parameter over time.

Python

```
Plt.plot(data['Date'], data['PM2.5'])
```

```
Plt.xlabel('Date')
Plt.ylabel('PM2.5')
Plt.title('PM2.5 Trend Over Time')
Plt.show()
```

- Histogram: Display the distribution of a particular air quality parameter.

```
Python
Sns.histplot(data['O3'], bins=20)
Plt.xlabel('Ozone Level')
Plt.ylabel('Frequency')
Plt.title('Distribution of Ozone Levels')
Plt.show()
```

- Box Plot: Compare the air quality parameters across different categories.

```
Python
Sns.boxplot(x='Location', y='PM10', data=data)
Plt.xlabel('Location')
```

```
Plt.ylabel('PM10')
```

```
Plt.title('Comparison of PM10 Levels across Locations')
```

```
Plt.show()
```

- Heatmap: Visualize the correlation between different air quality parameters.

Python

```
Corr_matrix = data[['PM2.5', 'PM10', 'O3', 'CO', 'SO2',  
'NO2']].corr()
```

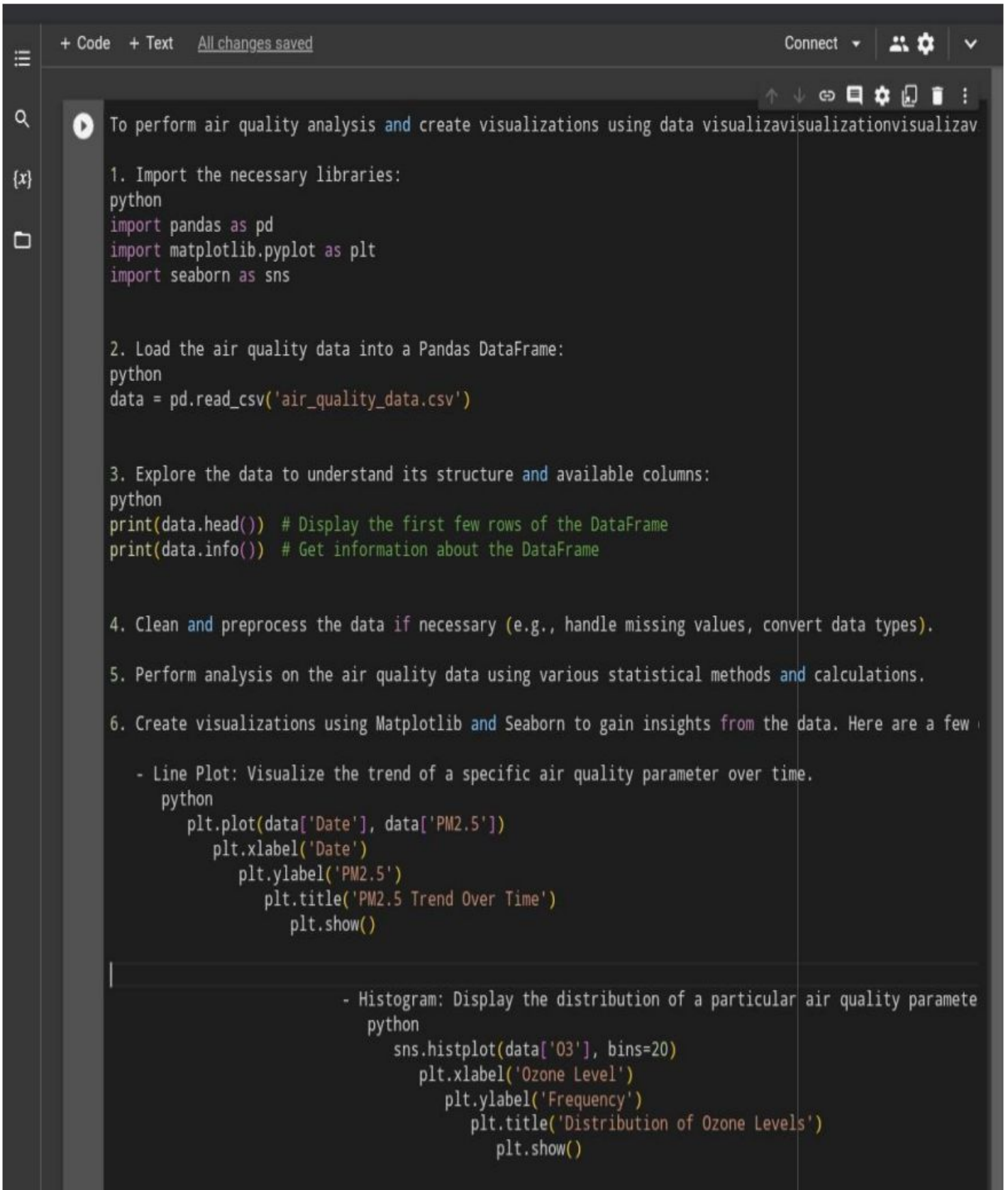
```
Sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
```

```
Plt.title('Correlation Between Air Quality Parameters')
```

```
Plt.show()
```

7. Customize the visualizations as per your requirements by adjusting colors, labels, titles, and other parameters.

Importing this code in Jupiter platform



Histogram: Display the distribution of a particular air quality parameter.

python

```
sns.histplot(data['O3'], bins=20)
plt.xlabel('Ozone Level')
plt.ylabel('Frequency')
plt.title('Distribution of Ozone Levels')
plt.show()
```

- Box Plot: Compare the air quality parameters across different categories.

python

```
sns.boxplot(x='Location', y='PM10', data=data)
plt.xlabel('Location')
plt.ylabel('PM10')
plt.title('Comparison of PM10 Levels across Locations')
plt.show()
```

- Heatmap: Visualize the correlation between different

python

```
corr_matrix = data[['PM2.5', 'PM10', 'O3', 'CO',
sns.heatmap(corr_matrix, annot=True, cmap='co
plt.title('Correlation Between Air Quality
plt.show()
```

```
+ Code + Text All changes saved Connect
visualization libraries such as Matplotlib and Seaborn, you will need

onvert data types).
and calculations.
he data. Here are a few examples:
time.

lar air quality parameter.

vels')

r quality parameters across different categories.
ion', y='PM10', data=data)
ion')
10')
Comparison of PM10 Levels across Locations')
()

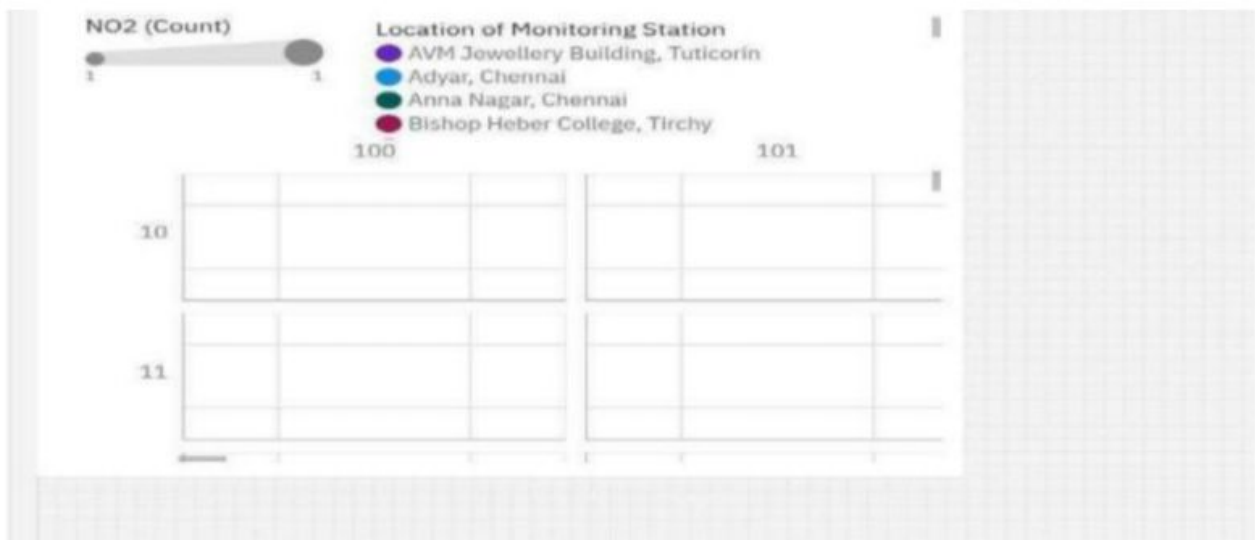
Heatmap: Visualize the correlation between different air quality parameters.
python
corr_matrix = data[['PM2.5', 'PM10', 'O3', 'CO', 'SO2', 'NO2']].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Between Air Quality Parameters')
plt.show()

7. Customize the visualizations as per your requirements by adjusting colors, labels,
Remember to interpret the visualizations and draw meaningful conclusions from the ana
7. Customize the visualizations as per your requirements by adjusting colors, labels,
7. Customize the visualizations as per your requirements by adjusting colors, labels,
Remember to interpret the visualizations and draw meaningful conclusions from the ana
```

Referred dataset link

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

Pollution trends and area with high pollution level found by using monitoring system with sampling date.



Conclusion

By this analysis we have concluded the high pollution level area through given sample dataset and it is implemented in ibm and Jupiter platform for better understanding and it greatly helps us to develop our project innovatively.