SKILL UP

IBM

AIR QUALITY ANALYSIS IN TAMIL NADU

DAC_Phase3

Team Members:

1. **Nithyasri V G**
2. **Sakshi Rajesh Bhavsar**
3. **Mathew P**
4. **Sarfraj Ansari**

AIR QUALITY ANALYSIS IN TAMIL NADU

Project Definition:

The primary objective of this project is to conduct a comprehensive assessment and visualization of air quality data collected from monitoring stations situated across Tamil Nadu. Through this project, we aim to extract valuable insights into the prevailing patterns of air pollution, identify regions exhibiting elevated levels of pollution, and establish a predictive model capable of estimating RSPM/PM10 levels. This analysis will be based on the concentrations of SO2 and NO2.

Phase 3: Development Part 1

-Building the project by loading and preprocessing the dataset.
-Begin the analysis by loading and preprocessing the air quality dataset.
-Loading the dataset using Python and data manipulation libraries (e.g., pandas)

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

Python libraries

Importing Python libraries

import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O

import matplotlib.pyplot as plt

%matplotlib inline

import seaborn as sns

Importing Python libraries

```
<!-- Importing Python libraries involves bringing external code modules into a Python program to access predefined functions,
classes, or data structures for specific tasks or functionalities. -->
```

In [28]:

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
```

Loding dataset from csv file.

```
df = pd.read_csv("air_quality_data.csv")

print(df)
```

Loding dataset from csv file.

<!-- Loading dataset from a CSV file is the process of importing structured data stored in a comma-separated format for analysis, manipulation, or further processing in data-related tasks -->

In [27]:
```
df = pd.read_csv("air_quality_data.csv")
print(df)
```

```
      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
```

```
                                          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   SO2   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]
```

df.shape

```
In [5]: # Representing the dimensions of a DataFrame.
        # It provides the number of rows and columns in the DataFrame.
        df.shape

Out[5]: (2879, 11)
```

Describing and defining imported dataset

print(df.head())

```
In [8]: print(df.head())

   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

                    Location of Monitoring Station  \
0  Kathivakkam, Municipal Kalyana Mandapam, Chennai
1  Kathivakkam, Municipal Kalyana Mandapam, Chennai
2  Kathivakkam, Municipal Kalyana Mandapam, Chennai
3  Kathivakkam, Municipal Kalyana Mandapam, Chennai
4  Kathivakkam, Municipal Kalyana Mandapam, Chennai

                                  Agency Type of Location   SO2   NO2  \
0  Tamilnadu State Pollution Control Board  Industrial Area  11.0  17.0
1  Tamilnadu State Pollution Control Board  Industrial Area  13.0  17.0
2  Tamilnadu State Pollution Control Board  Industrial Area  12.0  18.0
3  Tamilnadu State Pollution Control Board  Industrial Area  15.0  16.0
4  Tamilnadu State Pollution Control Board  Industrial Area  13.0  14.0

   RSPM/PM10  PM 2.5
0       55.0     NaN
1       45.0     NaN
2       50.0     NaN
3       46.0     NaN
4       42.0     NaN
```

print(df.tail())

```
In [9]: print(df.tail())

      Stn Code Sampling Date       State City/Town/Village/Area  \
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

     Location of Monitoring Station                                  Agency  \
2874      Central Bus Stand, Trichy  Tamilnadu State Pollution Control Board
2875      Central Bus Stand, Trichy  Tamilnadu State Pollution Control Board
2876      Central Bus Stand, Trichy  Tamilnadu State Pollution Control Board
2877      Central Bus Stand, Trichy  Tamilnadu State Pollution Control Board
2878      Central Bus Stand, Trichy  Tamilnadu State Pollution Control Board

                        Type of Location   SO2   NO2  RSPM/PM10  PM 2.5
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
```

df.describe()

```
In [10]: df.describe()
```

Out[10]:

	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

df.info()

```
In [11]: df.info()

<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
```

Data preprocessing

import pandas as pd

df = pd.read_csv("air_quality_data.csv")

df.drop(labels=['Stn Code','Location of Monitoring Station','Agency'], axis = 1, inplace = True)

df.sample(5)

Data preprocessing

<!-- Data preprocessing: Preparing raw data for analysis by cleaning, transforming, and organizing it to enhance quality, relevance, and usability. -->

```
In [71]: import pandas as pd
df = pd.read_csv("air_quality_data.csv")
df.drop(labels=['Stn Code','Location of Monitoring Station','Agency'], axis = 1, inplace = True)
df.sample(5)
```

Out[71]:

	Sampling Date	State	City/Town/Village/Area	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
40	06-12-2014	Tamil Nadu	Chennai	Industrial Area	14.0	18.0	39.0	NaN
1735	07-05-2014	Tamil Nadu	Madurai	Industrial Area	9.0	24.0	35.0	NaN
1990	21-01-2014	Tamil Nadu	Mettur	Industrial Area	10.0	27.0	67.0	NaN
477	09-11-2014	Tamil Nadu	Chennai	Residential, Rural and other Areas	14.0	27.0	64.0	NaN
2046	08-07-2014	Tamil Nadu	Mettur	Industrial Area	9.0	69.0	44.0	NaN

df.isnull().sum()

```
In [45]: df.isnull().sum()

Out[45]: Sampling Date                0
         State                        0
         City/Town/Village/Area       0
         Type of Location             0
         SO2                         11
         NO2                         13
         RSPM/PM10                    4
         PM 2.5                    2879
         dtype: int64
```

Omitting PM 2.5 column from the dataset:

```
import pandas as pd

df = pd.read_csv("air_quality_data.csv")

new_df=df.drop(labels=['PM 2.5'], axis = 1, inplace = True)

new_df=df.sample(2)
```

```
In [47]: # PM 2.5 has 2879 data missing. So omitting PM 2.5 column from the dataset.
         import pandas as pd
         df = pd.read_csv("air_quality_data.csv")
         new_df=df.drop(labels=['PM 2.5'], axis = 1, inplace = True)
         new_df=df.sample(2)

In [9]: print(new_df)

      Stn Code Sampling Date       State City/Town/Village/Area  \
2052       763    26-08-2014  Tamil Nadu                 Mettur
493        765    11-12-2014  Tamil Nadu                Chennai

        Location of Monitoring Station  \
2052  SIDCO Industrial Complex, Mettur
493                Anna Nagar, Chennai

                                     Agency  \
2052  Tamilnadu State Pollution Control Board
493   Tamilnadu State Pollution Control Board

                       Type of Location   SO2   NO2  RSPM/PM10
2052                    Industrial Area   9.0  24.0       44.0
493   Residential, Rural and other Areas  15.0  26.0       62.0
```

Fill the missing values:

df=new_df

df.dtypes

```
In [10]: # In order to fill the missing values:
         # the values are first need to be sorted in Chronological order
         df=new_df
         df.dtypes

Out[10]: Stn Code                            int64
         Sampling Date                      object
         State                              object
         City/Town/Village/Area             object
         Location of Monitoring Station     object
         Agency                             object
         Type of Location                   object
         SO2                               float64
         NO2                               float64
         RSPM/PM10                         float64
         dtype: object
```

Converting "object" data type to "datetime"

df['Sampling Date'] = pd.to_datetime(df['Sampling Date'],format='%d-%m-%Y')

df.info()

```
In [12]: # To sort based on dates, the date should be of "datetime" datatype.
         #So converting "object" data type to "datetime" datatype
         df['Sampling Date'] = pd.to_datetime(df['Sampling Date'],format='%d-%m-%Y')
         df.info()

         <class 'pandas.core.frame.DataFrame'>
         Index: 2 entries, 2052 to 493
         Data columns (total 10 columns):
          #   Column                          Non-Null Count  Dtype
         ---  ------                          --------------  -----
          0   Stn Code                        2 non-null      int64
          1   Sampling Date                   2 non-null      datetime64[ns]
          2   State                           2 non-null      object
          3   City/Town/Village/Area          2 non-null      object
          4   Location of Monitoring Station  2 non-null      object
          5   Agency                          2 non-null      object
          6   Type of Location                2 non-null      object
          7   SO2                             2 non-null      float64
          8   NO2                             2 non-null      float64
          9   RSPM/PM10                       2 non-null      float64
         dtypes: datetime64[ns](1), float64(3), int64(1), object(5)
         memory usage: 176.0+ bytes
```

```
df.sort_values(by='Sampling Date')
```

```
In [48]: df.sort_values(by='Sampling Date')
```

Out[48]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10
0	38	01-02-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0
2512	769	01-02-2014	Tamil Nadu	Trichy	Gandhi Market, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	16.0	19.0	102.0
508	764	01-02-2014	Tamil Nadu	Chennai	Adyar, Chennai	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	19.0	61.0
624	767	01-02-2014	Tamil Nadu	Chennai	Kilpauk, Chennai	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	18.0	23.0	83.0
2661	771	01-02-2014	Tamil Nadu	Trichy	Bishop Heber College, Tirchy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	11.0	14.0	53.0
...	...	...	...	...	...	...	...	...	...	...
623	764	31-12-2014	Tamil Nadu	Chennai	Adyar, Chennai	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	20.0	42.0
1785	307	31-12-2014	Tamil Nadu	Madurai	Fenner (I) Ltd. Employees Assiciation Building...	Tamilnadu State Pollution Control Board	Industrial Area	13.0	26.0	25.0
1588	760	31-12-2014	Tamil Nadu	Cuddalore	SIPCOT Industrial Complex, Cuddalore	Tamilnadu State Pollution Control Board	Industrial Area	6.0	17.0	44.0
2218	309	31-12-2014	Tamil Nadu	Salem	Sowdeswari College Building, Salem	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	8.0	29.0	57.0
2878	773	31-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.0	94.0

Handle missing data:

```
# data preprocessing to handle missing data
new_df['SO2'].fillna(method='ffill',inplace = True);
new_df['NO2'].fillna(method='ffill',inplace = True);
new_df['RSPM/PM10'].fillna(method='ffill',inplace = True);
print(df.head(2))
```

```
In [14]: # data preprocessing to handle missing data
         new_df['SO2'].fillna(method='ffill',inplace = True);
         new_df['NO2'].fillna(method='ffill',inplace = True);
         new_df['RSPM/PM10'].fillna(method='ffill',inplace = True);
         print(df.head(2))

      Stn Code Sampling Date       State City/Town/Village/Area  \
2052       763    2014-08-26  Tamil Nadu                 Mettur
493        765    2014-12-11  Tamil Nadu                Chennai

        Location of Monitoring Station  \
2052  SIDCO Industrial Complex, Mettur
493                Anna Nagar, Chennai

                                   Agency  \
2052  Tamilnadu State Pollution Control Board
493   Tamilnadu State Pollution Control Board

                       Type of Location   SO2   NO2  RSPM/PM10
2052                    Industrial Area   9.0  24.0       44.0
493   Residential, Rural and other Areas  15.0  26.0       62.0
```

df.isnull().sum()

```
In [71]: df.isnull().sum()

Out[71]: Stn Code                          0
         Sampling Date                     0
         State                             0
         City/Town/Village/Area            0
         Location of Monitoring Station    0
         Agency                            0
         Type of Location                  0
         SO2                               0
         NO2                               0
         RSPM/PM10                         0
         dtype: int64
```

Finding hidden missing values:

missing_val= (df == 0).astype(int).sum(axis=0)

print(missing_val)

```
In [51]: #Finding hidden missing values; eg:0
         missing_val= (df == 0).astype(int).sum(axis=0)
         print(missing_val)

         Stn Code                          0
         Sampling Date                     0
         State                             0
         City/Town/Village/Area            0
         Location of Monitoring Station    0
         Agency                            0
         Type of Location                  0
         SO2                               0
         NO2                               0
         RSPM/PM10                         0
         dtype: int64
```

Chronological order:

df.dtypes

```
In [19]: # sorted in Chronological order
         df.dtypes

Out[19]: Stn Code                                int64
         Sampling Date                    datetime64[ns]
         State                                   object
         City/Town/Village/Area                  object
         Location of Monitoring Station          object
         Agency                                  object
         Type of Location                        object
         SO2                                    float64
         NO2                                    float64
         RSPM/PM10                              float64
         dtype: object
```

Sorting the date wise:

df.sort_values(by='Sampling Date')

```
In [77]: # sorting the date wise
         df.sort_values(by='Sampling Date')
```

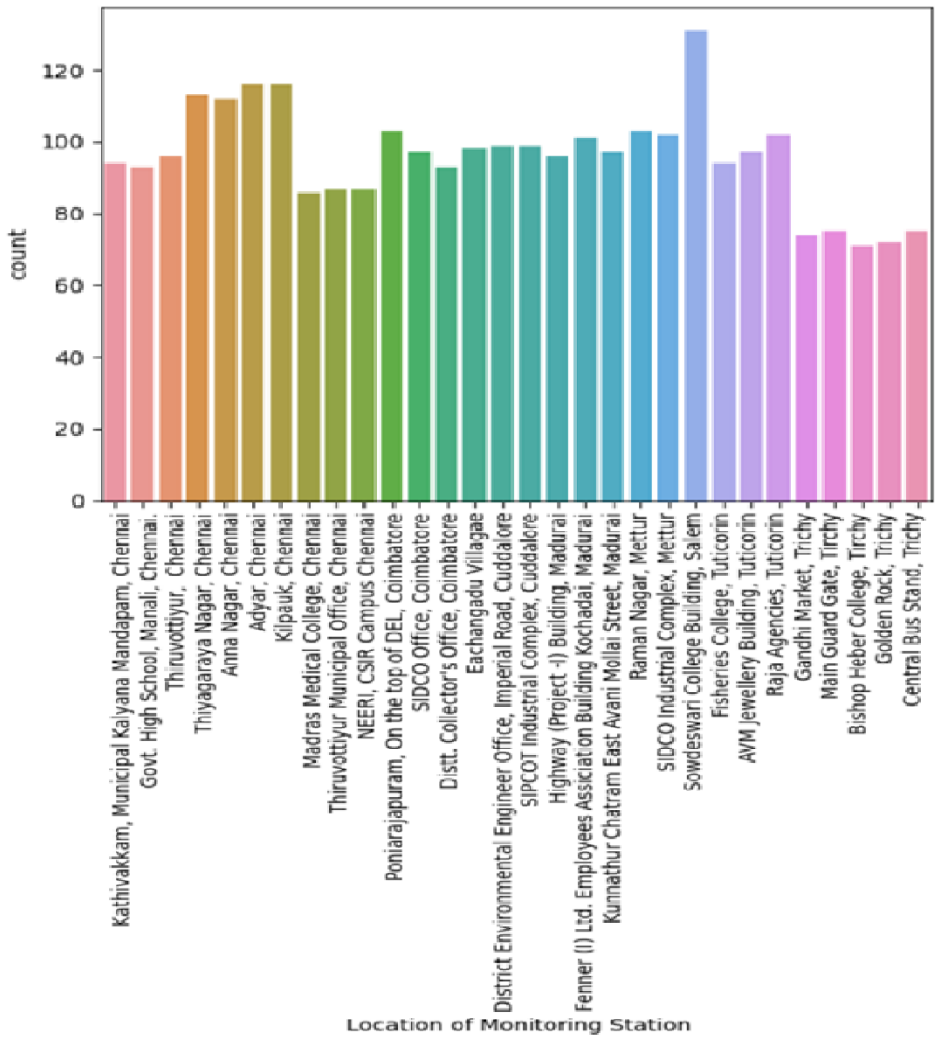
Out[77]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10
1151	237	2014-06-27	Tamil Nadu	Coimbatore	SIDCO Office, Coimbatore	Tamilnadu State Pollution Control Board	Industrial Area	3.0	29.0	39.0
1643	306	2014-07-28	Tamil Nadu	Madurai	Highway (Project -I) Building, Madurai	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	10.0	27.0	43.0

Modifying the insights data:

```
import pandas as pd

new_df = pd.read_csv("air_quality_data.csv")

datacount =sns.countplot(x ="Location of Monitoring Station",data =new_df);

datacount.set_xticklabels(datacount.get_xticklabels(), rotation=90);
```

```
In [29]: # Data Visualization
         # Modifying the insights datas
         import pandas as pd
         new_df = pd.read_csv("air_quality_data.csv")
         datacount =sns.countplot(x ="Location of Monitoring Station",data =new_df);
         datacount.set_xticklabels(datacount.get_xticklabels(), rotation=90);
```



Type of Location:

df['Type of Location'].unique()

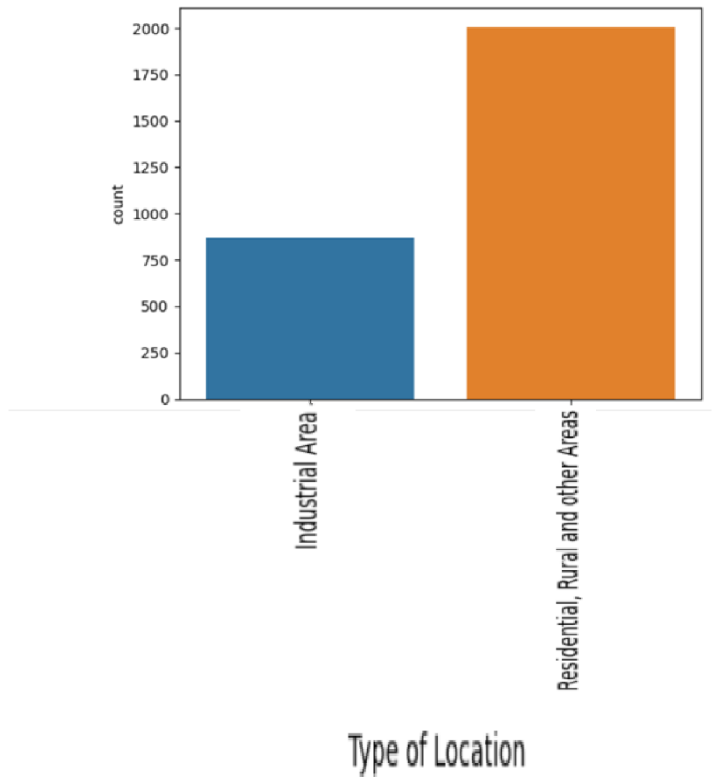
```
In [24]: # Type of Location
         df['Type of Location'].unique()

Out[24]: array(['Industrial Area', 'Residential, Rural and other Areas'],
               dtype=object)
```

Type of Location as Industrial & Residential area

```
import pandas as pd
new_df= pd.read_csv("air_quality_data.csv")
typ=sns.countplot(x ="Type of Location",data =new_df)
typ.set_xticklabels(typ.get_xticklabels(), rotation=90)
```

```
In [78]: import pandas as pd
         new_df= pd.read_csv("air_quality_data.csv")
         typ=sns.countplot(x ="Type of Location",data =new_df)
         typ.set_xticklabels(typ.get_xticklabels(), rotation=90)
```

```
Out[78]: [Text(0, 0, 'Industrial Area'),
          Text(1, 0, 'Residential, Rural and other Areas')]
```

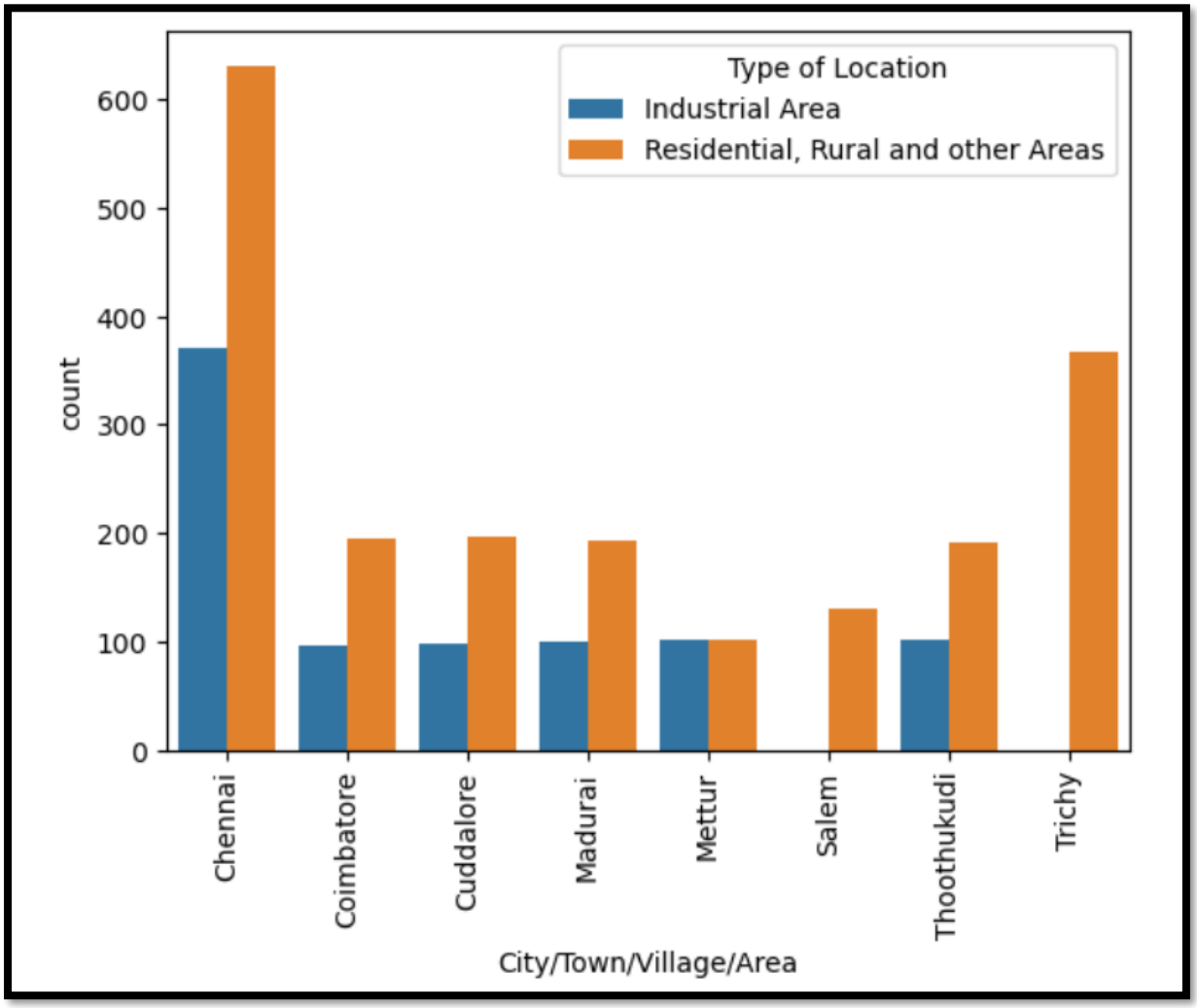


City area splitting Type of Location as Industrial & Residential area

datacount_ty =sns.countplot(x ='City/Town/Village/Area',hue ='Type of Location',data = df);

datacount_ty.set_xticklabels(datacount_ty.get_xticklabels(), rotation=90);

```
In [160]: datacount_ty =sns.countplot(x ='City/Town/Village/Area',hue ='Type of Location',data = df);
          datacount_ty.set_xticklabels(datacount_ty.get_xticklabels(), rotation=90);
```

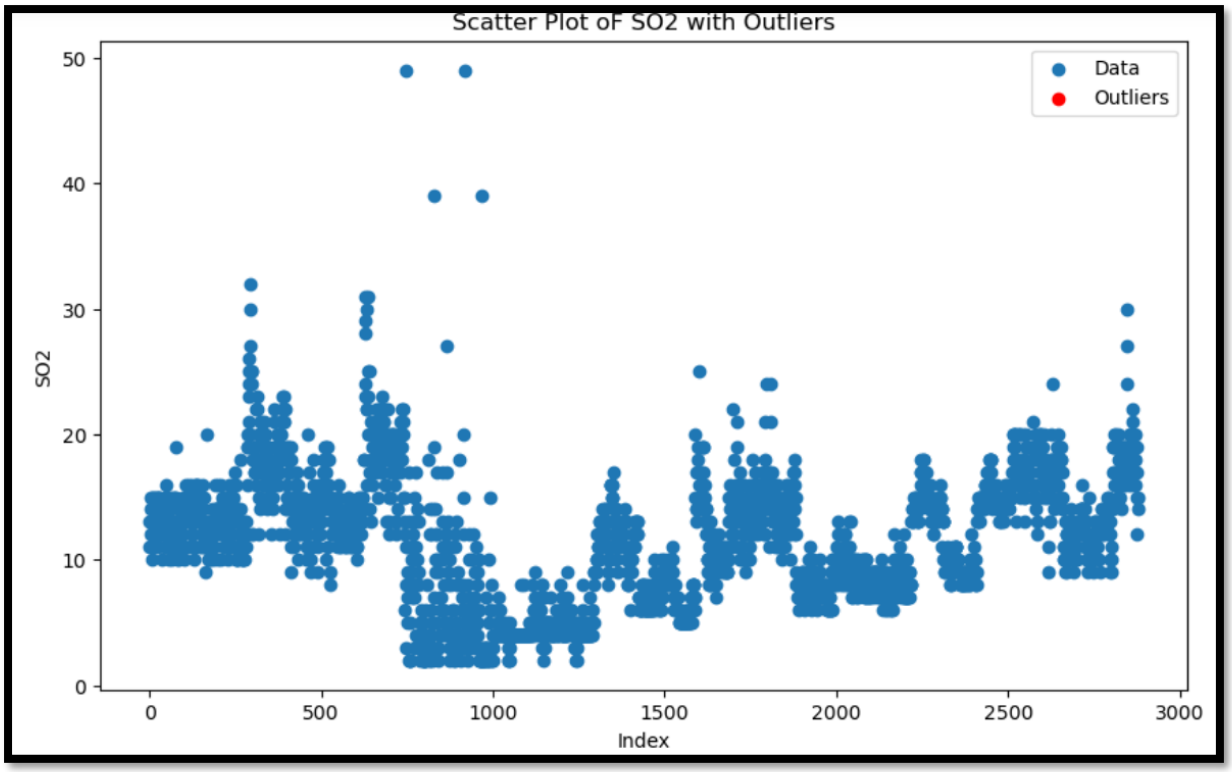


Checking of Outliers:

Scatter Plot of SO2 with Outliers:

```
from scipy.stats import zscore

column_name = 'SO2'

column_data = new_df[column_name]

z_scores = zscore(column_data)

threshold = 3

outliers = (z_scores > threshold) | (z_scores < -threshold)

# scatter plot

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

plt.scatter(new_df.index, column_data, label='Data')

plt.scatter(new_df.index[outliers], column_data[outliers], color='red', label='Outliers')

plt.xlabel('Index')

plt.ylabel(column_name)

plt.title('Scatter Plot oF SO2 with Outliers')

plt.legend()

plt.show()
```

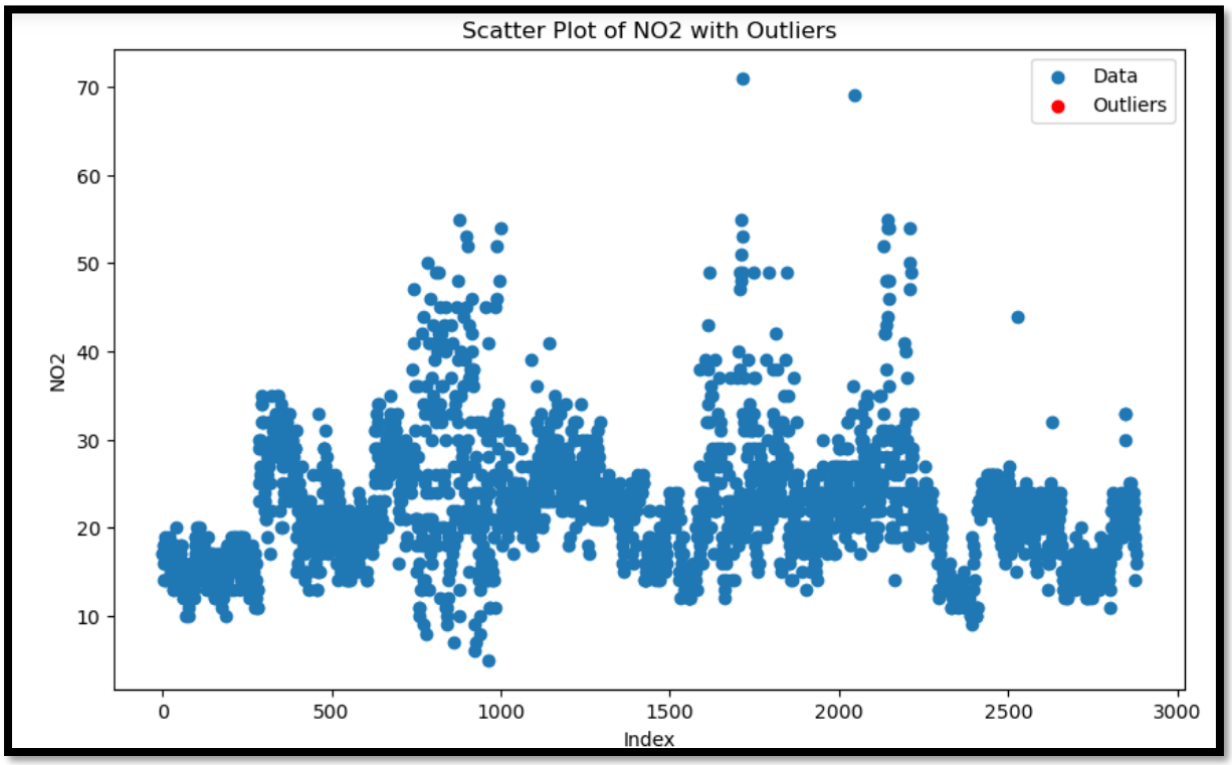
```
In [31]: # Checking of Outliers
         # SO2
         from scipy.stats import zscore
         column_name = 'SO2'
         column_data = new_df[column_name]
         z_scores = zscore(column_data)
         threshold = 3
         outliers = (z_scores > threshold) | (z_scores < -threshold)
         # scatter plot
         plt.figure(figsize=(10, 6))
         plt.scatter(new_df.index, column_data, label='Data')
         plt.scatter(new_df.index[outliers], column_data[outliers], color='red', label='Outliers')
         plt.xlabel('Index')
         plt.ylabel(column_name)
         plt.title('Scatter Plot oF SO2 with Outliers')
         plt.legend()
         plt.show()
```



Scatter Plot of NO2 with Outliers:

```
from scipy.stats import zscore
new_df = pd.read_csv("air_quality_data.csv")
column_name = 'NO2'
column_data = new_df[column_name]
z_scores = zscore(column_data)
threshold = 3
outliers = (z_scores > threshold) | (z_scores < -threshold)
# scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(new_df.index, column_data, label='Data')
plt.scatter(new_df.index[outliers], column_data[outliers], color='red', label='Outliers')
plt.xlabel('Index')
plt.ylabel(column_name)
plt.title('Scatter Plot of NO2 with Outliers')
plt.legend()
```

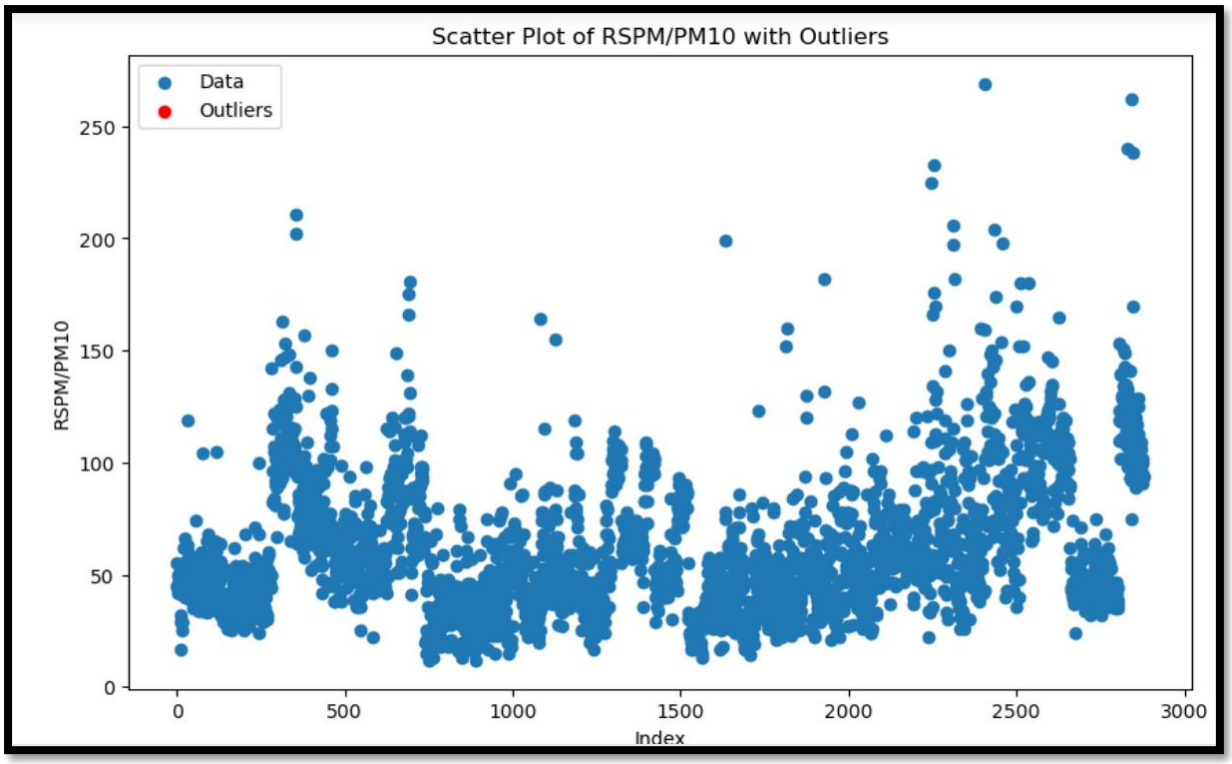
```
In [32]: # Checking of Outliers
         # NO2
         from scipy.stats import zscore
         new_df = pd.read_csv("air_quality_data.csv")
         column_name = 'NO2'
         column_data = new_df[column_name]
         z_scores = zscore(column_data)
         threshold = 3
         outliers = (z_scores > threshold) | (z_scores < -threshold)
         # scatter plot
         plt.figure(figsize=(10, 6))
         plt.scatter(new_df.index, column_data, label='Data')
         plt.scatter(new_df.index[outliers], column_data[outliers], color='red', label='Outliers')
         plt.xlabel('Index')
         plt.ylabel(column_name)
         plt.title('Scatter Plot of NO2 with Outliers')
         plt.legend()
         plt.show()
```



Scatter Plot of RSPM/PM10

```
#RSPM/PM10
# Checking of Outliers
from scipy.stats import zscore
new_df = pd.read_csv("air_quality_data.csv")
column_name = 'RSPM/PM10'
column_data = new_df[column_name]
z_scores = zscore(column_data)
threshold = 3
outliers = (z_scores > threshold) | (z_scores < -threshold)
# scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(new_df.index, column_data, label='Data')
plt.scatter(new_df.index[outliers], column_data[outliers], color='red', label='Outliers')
plt.xlabel('Index')
plt.ylabel(column_name)
plt.title('Scatter Plot of RSPM/PM10 with Outliers')
```

```
In [80]: # RSPM/PM10
         # Checking of Outliers
         from scipy.stats import zscore
         new_df = pd.read_csv("air_quality_data.csv")
         column_name = 'RSPM/PM10'
         column_data = new_df[column_name]
         z_scores = zscore(column_data)
         threshold = 3
         outliers = (z_scores > threshold) | (z_scores < -threshold)
         # scatter plot
         plt.figure(figsize=(10, 6))
         plt.scatter(new_df.index, column_data, label='Data')
         plt.scatter(new_df.index[outliers], column_data[outliers], color='red', label='Outliers')
         plt.xlabel('Index')
         plt.ylabel(column_name)
         plt.title('Scatter Plot of RSPM/PM10 with Outliers')
         plt.legend()
         plt.show()
```



Conclusion:

- Dataset imported successfully by importing various python libraries.
- Data describing and defining.
- Removing null values.
- Converting "object" data type to "datetime" datatype
- No Outliers found in data columns of -SO2, NO2, RSPM/PM10
- Hence Dataset is now cleaned and ready for further analysis process.