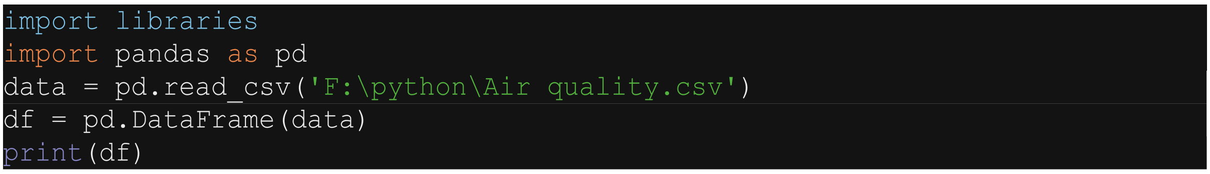
PHASE 3 - AIR QUALITY ANALYSIS AND PREDICTION IN TAMIL NADU

The project aims to analyze and visualize air quality data from monitoring stations in Tamil Nadu. The objective is to gain insights into air pollution trends, identify areas with high pollution levels, and develop a predictive model to estimate RSPM/PM10 levels based on SO2 and NO2 levels. This project involves defining objectives, designing the analysis approach, selecting visualization techniques, and creating a predictive model using Python and relevant libraries.

DEVELOPMENT PART 1:

Step 1: Data Loading

Data loading is the process of bringing external data into a format suitable for analysis. In this case, we've imported data in CSV format by utilizing the Pandas library and subsequently printed it to confirm the successful loading of the data.



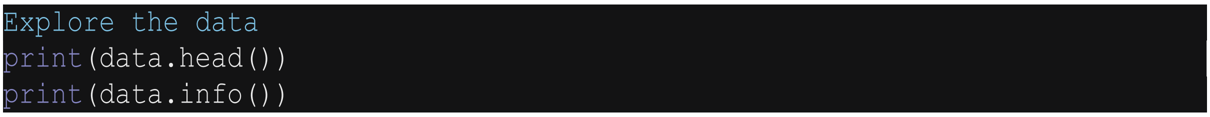
```
import libraries
import pandas as pd
data = pd.read_csv('F:\python\Air quality.csv')
df = pd.DataFrame(data)
print(df)
```

Step 2: Explore the data

Exploring the data using the head() and info() function is a process of initially examining a dataset to understand its structure, content, and quality.

head()- This function displays the first few rows of the dataset.

info()- It displays information about the data types of each column, the number of non-null entries, and the memory usage.

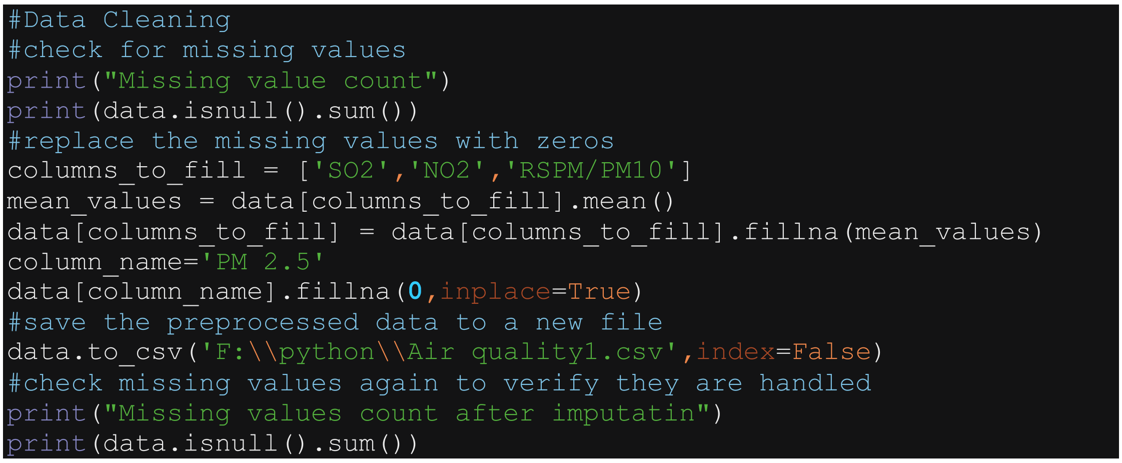


```
Explore the data
print(data.head())
print(data.info())
```

Step 3: Data cleaning

To address the issue of missing values in the provided dataset, we can resolve it by filling those missing values with mean.

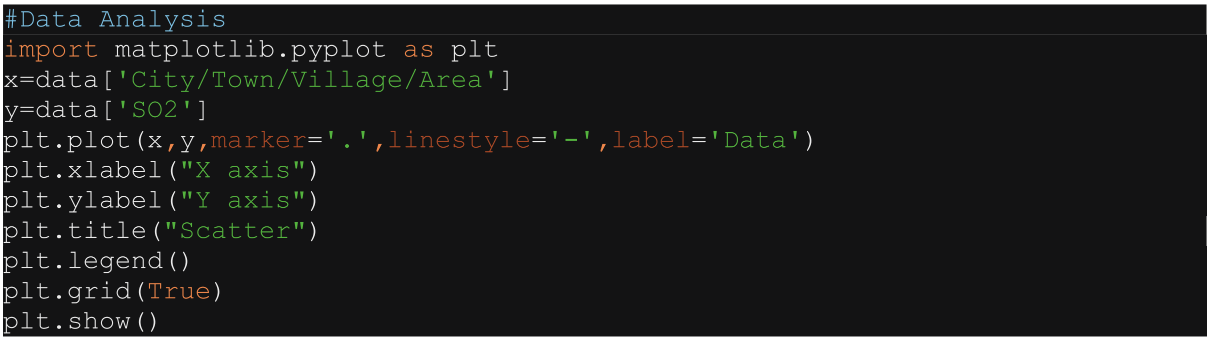
- ✓ Check whether the data set contain any missing values
- ✓ Replace the missing values with means
- ✓ Save the preprocessed data to a new file
- ✓ Check missing values again to verify they are handled



```
#Data Cleaning
#check for missing values
print("Missing value count")
print(data.isnull().sum())
#replace the missing values with zeros
columns_to_fill = ['SO2','NO2','RSPM/PM10']
mean_values = data[columns_to_fill].mean()
data[columns_to_fill] = data[columns_to_fill].fillna(mean_values)
column_name='PM 2.5'
data[column_name].fillna(0,inplace=True)
#save the preprocessed data to a new file
data.to_csv('F:\\python\\Air quality1.csv',index=False)
#check missing values again to verify they are handled
print("Missing values count after imputatin")
print(data.isnull().sum())
```

Step 4: Data Analysis

This analysis aims to visually assess patterns and variations in SO2 levels across different locations (City/Town/Village/Area). It helps identify areas with notably high or low SO2 pollution levels, providing insights into air quality variations across different areas.



```
#Data Analysis
import matplotlib.pyplot as plt
x=data['City/Town/Village/Area']
y=data['SO2']
plt.plot(x,y,marker='.',linestyle='-',label='Data')
plt.xlabel("X axis")
plt.ylabel("Y axis")
plt.title("Scatter")
plt.legend()
plt.grid(True)
plt.show()
```

Step 5: Scatter Plot

It creates the scatter plot with the specified data, axis labels, color, size, and title.The plot visually represents the relationship between SO2, NO2, and RSPM/PM10 levels, with color and marker size indicating RSPM/PM10 levels, making it easy to observer patterns and associations between these variables.

```
#scatter plot using plotly
# import plotly.express as px
fig = px.scatter(df, x='SO2', y='NO2', color='RSPM/PM10', size='RSPM/PM10',
                 labels={'SO2': 'SO2 Level', 'NO2': 'NO2 Level',
'RSPM/PM10': 'RSPM/PM10 Level'},
                 title='Scatter Plot of SO2 vs. NO2 with RSPM/PM10 Color
and Size'
                )

fig.show()
```

CODE:

```
#import libraries
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px

# Create a DataFrame from the provided data
data = pd.read_csv('F:\\python\\Air quality.csv')
df = pd.DataFrame(data)
print(df)

#Explore the data
print(data.head())
print(data.info())

#Access specific column
so2_column = data['SO2']
no2_column = data['NO2']
RSPM_column = data['RSPM/PM10']
date_column = data['Sampling Date']

#Data Cleaning
#check for missing values
print("Missing value count")
print(data.isnull().sum())
#replace the missing values with zeros
columns_to_fill = ['SO2','NO2','RSPM/PM10']
mean_values = data[columns_to_fill].mean()
data[columns_to_fill] = data[columns_to_fill].fillna(mean_values)
column_name='PM 2.5'
data[column_name].fillna(0,inplace=True)
#save the preprocessed data to a new file
data.to_csv('F:\\python\\Air quality1.csv',index=False)
#check missing values again to verify they are handled
print("Missing values count after imputation")
print(data.isnull().sum())

df = pd.DataFrame(data)
print(df)
```

```
#Data Analysis
x=data['City/Town/Village/Area']
y=data['SO2']
plt.plot(x,y,marker='.',linestyle='-',label='Data')
plt.xlabel("City")
plt.ylabel("SO2")
plt.title("Scatter")
plt.legend()
plt.grid(True)
plt.show()

#scatter plot using plotly
fig = px.scatter(df, x='SO2', y='NO2', color='RSPM/PM10', size='RSPM/PM10',
                 labels={'SO2': 'SO2 Level', 'NO2': 'NO2 Level', 'RSPM/PM10': 'RSPM/PM10 Level'},
                 title='Scatter Plot of SO2 vs. NO2 with RSPM/PM10 Color and Size'
                )

fig.show()
```

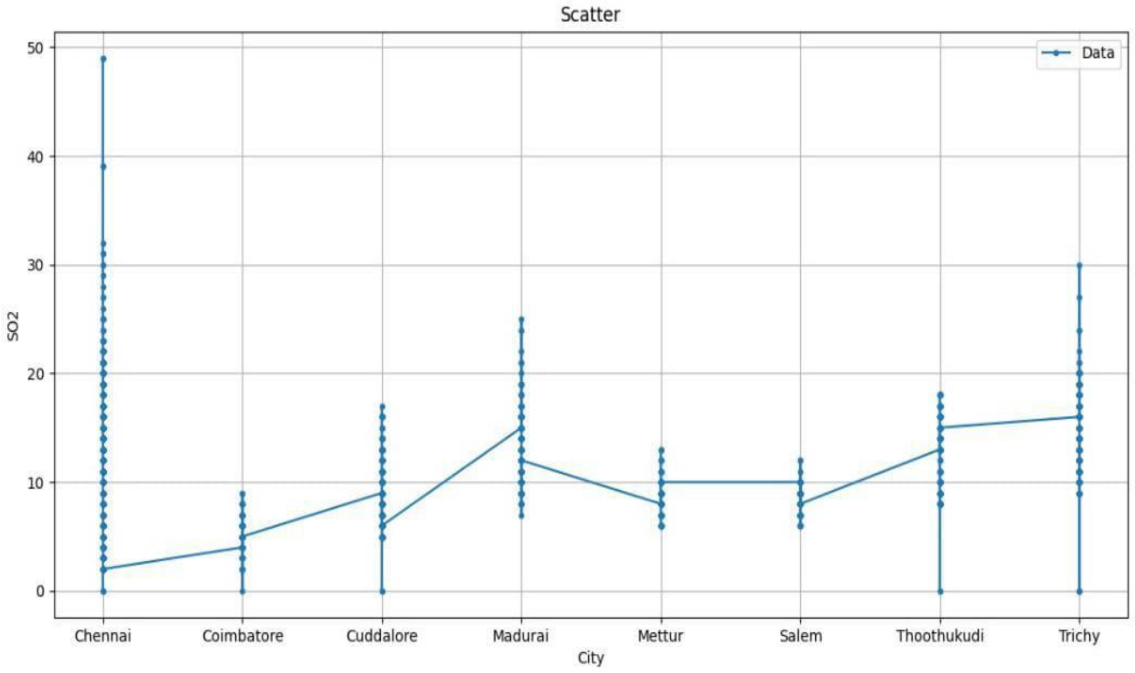
OUTPUT:

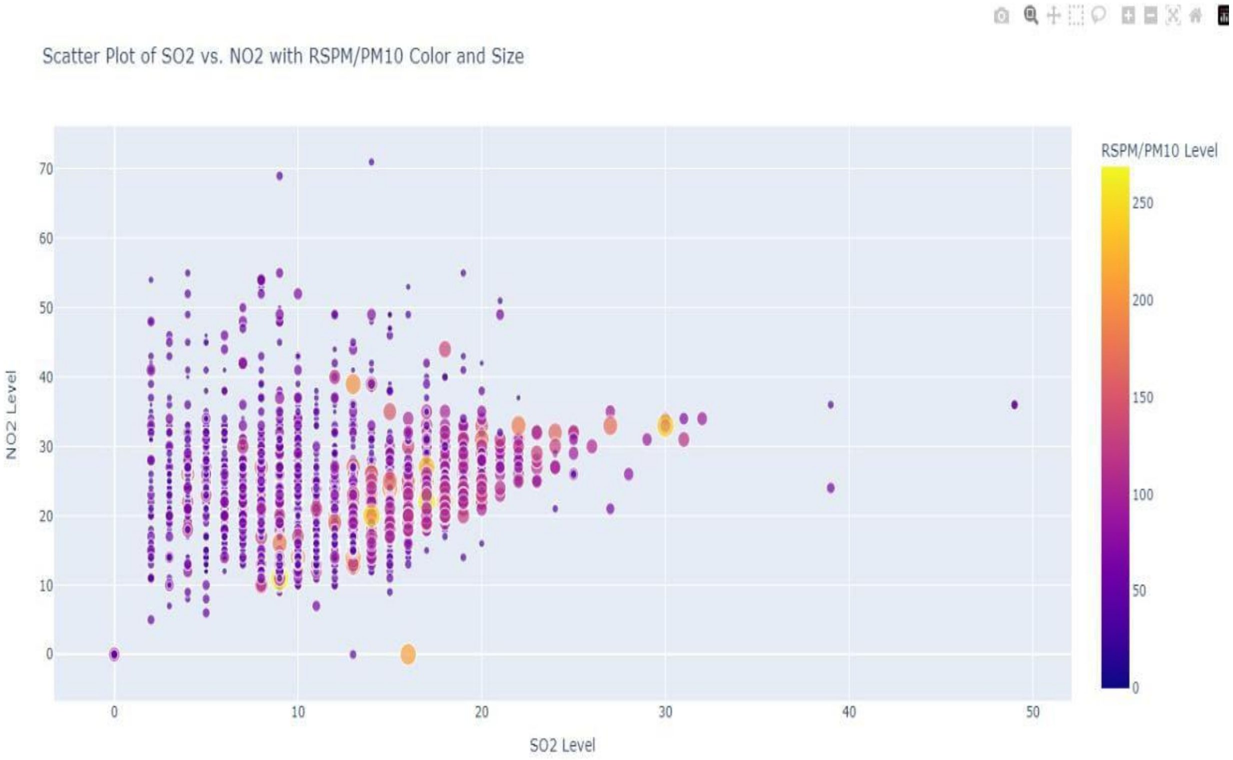
```
Stn Code Sampling Date       State  ...  NO2 RSPM/PM10 PM 2.5
0          38    01-02-2014  Tamil Nadu  ...  17.0      55.0    NaN
1          38    01-07-2014  Tamil Nadu  ...  17.0      45.0    NaN
2          38    21-01-2014  Tamil Nadu  ...  18.0      50.0    NaN
3          38    23-01-2014  Tamil Nadu  ...  16.0      46.0    NaN
4          38    28-01-2014  Tamil Nadu  ...  14.0      42.0    NaN
        ...           ...          ...  ...   ...       ...     ...
2874       773    12-03-2014  Tamil Nadu  ...  18.0     102.0    NaN
2875       773    12-10-2014  Tamil Nadu  ...  14.0      91.0    NaN
2876       773    17-12-2014  Tamil Nadu  ...  22.0     100.0    NaN
2877       773    24-12-2014  Tamil Nadu  ...  17.0      95.0    NaN
2878       773    31-12-2014  Tamil Nadu  ...  16.0      94.0    NaN
[2879 rows x 11 columns]
   Stn Code Sampling Date       State  ...  NO2 RSPM/PM10 PM 2.5
0        38    01-02-2014  Tamil Nadu  ...  17.0      55.0    NaN
1        38    01-07-2014  Tamil Nadu  ...  17.0      45.0    NaN
```

```
2       38    21-01-2014  Tamil Nadu  ...  18.0      50.0    NaN
3       38    23-01-2014  Tamil Nadu  ...  16.0      46.0    NaN
4       38    28-01-2014  Tamil Nadu  ...  14.0      42.0    NaN
[5 rows x 11 columns]
<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
None
Missing value count
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                            11
NO2                            13
RSPM/PM10                       4
PM 2.5                       2879
dtype: int64
Missing values count after imputatin
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
PM 2.5                          0
dtype: int64
     Stn Code Sampling Date       State  ...   NO2 RSPM/PM10 PM 2.5
0          38    01-02-2014  Tamil Nadu  ...  17.0      55.0    0.0
1          38    01-07-2014  Tamil Nadu  ...  17.0      45.0    0.0
2          38    21-01-2014  Tamil Nadu  ...  18.0      50.0    0.0
3          38    23-01-2014  Tamil Nadu  ...  16.0      46.0    0.0
4          38    28-01-2014  Tamil Nadu  ...  14.0      42.0    0.0
      ...           ...         ...  ...   ...       ...    ...
2874      773    12-03-2014  Tamil Nadu  ...  18.0     102.0    0.0
2875      773    12-10-2014  Tamil Nadu  ...  14.0      91.0    0.0
2876      773    17-12-2014  Tamil Nadu  ...  22.0     100.0    0.0
2877      773    24-12-2014  Tamil Nadu  ...  17.0      95.0    0.0
```

```
2878       773    31-12-2014  Tamil Nadu  ...  16.0      94.0    0.0
[2879 rows x 11 columns]
Backend TkAgg is interactive backend. Turning interactive mode on.
```


Scatter
Data
SO2
Chennai
Coimbatore
Cuddalore
Madurai
Mettur
Salem
Thoothukudi
Trichy
City


Scatter Plot of SO2 vs. NO2 with RSPM/PM10 Color and Size
NO2 Level
SO2 Level
RSPM/PM10 Level