GOVERNMENT COLLEGE OF TECHNOLOGY, COIMBATORE-13

PROJECT NAME: **Air Quality Assessment TN**

Team Members:

1. Kokulavenkat K
2. Keerthana R
3. Kayalvizhi T
4. Karunasri A

Phase 5: PROJECT DOCUMENTATION AND SUBMISSION

Project Description:

Gather air quality data from monitoring stations across Tamil Nadu. This data may include measurements of pollutants like PM2.5, PM10, nitrogen dioxide (NO2), sulphur dioxide (SO2), ozone (O3), carbon monoxide (CO), and other relevant parameters.

Ensure the data is structured, accurate, and includes timestamps for each measurement. Aggregate data if multiple stations provide data for the same location. To visualise the data including line charts, bar graphs, heatmaps, and scatter plots and compare different pollutants. Geographic visualizations using tools like GIS software or Python libraries like Folium to map air quality across different regions in Tamil Nadu.

Consider using color-coding to represent air quality levels and station locations on the map. Calculate AQI based on the collected data and the relevant formula for Tamil Nadu. To visualise the air quality find how different pollutants relate to each other and apply time series analysis techniques, autoregressive integrated moving average (ARIMA) or prophet for forecasting air quality.

Analysis Approach:

To accomplish the project objectives, we will follow these steps:

- Data Collection: Gather historical air quality data from monitoring stations in Tamil Nadu. This data will include information on pollutants like RSPM/PM10, SO2, and NO2, as well as associated timestamps and geographical coordinates.

- Data Preprocessing: Clean and preprocess the data, addressing issues such as missing values, outliers, and data consistency. This step is crucial to ensure the quality and reliability of the analysis.
- Data Analysis: Apply statistical analysis techniques to uncover air quality trends, seasonal variations, and anomalies. This will involve time-series analysis and hypothesis testing.
- Data Visualization: Select appropriate visualization techniques, such as line charts for trends, heatmaps for geographical patterns, and scatter plots for correlation analysis. Interactive visualizations will be created for dynamic exploration.
- GIS Integration: Utilize GIS tools to geospatially represent air quality data, enabling the identification of pollution hotspots and regional variations

Techniques Used:

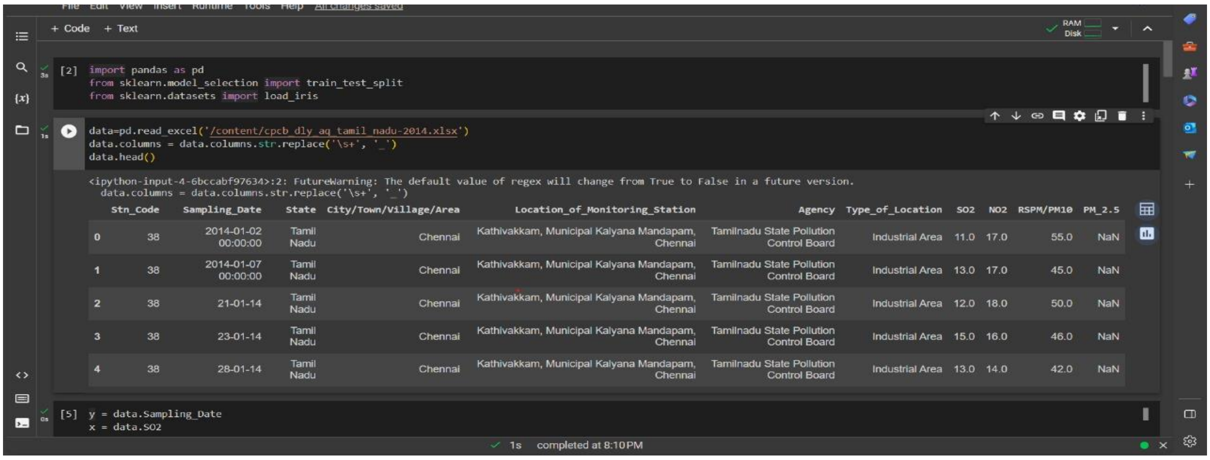
Here are some specific details about libraries that can be used for an air quality analysis project in Tamil Nadu:

- Pandas: Pandas is a fundamental library for data analysis and manipulation. You can use it to load, clean, and process air quality data in various formats such as CSV, Excel, or databases. This library is essential for data preprocessing.
- Matplotlib and Seaborn: Matplotlib and Seaborn are great choices for creating visualizations of air quality data. You can generate time series plots, scatter plots, bar charts, and maps to understand the trends and patterns in air quality over time and across locations in Tamil Nadu.
- NumPy: NumPy is essential for numerical operations. You can use it for various calculations, such as statistical analysis, averaging air quality values, and more.
- Tamil Nadu Pollution Control Board (TNPCB) API: If TNPCB or other relevant agencies provide air quality data through APIs, you can use these to fetch real time or historical data for your analysis.

LOADING A DATASET:

▪ Load a dataset into a pandas DataFrame using `pd.read_csv()` for CSV files or `pd.read_excel()` for Excel files.

▪ Ensure that 'data' is correctly loaded before applying the provided code for preprocessing.



```
[2] import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.datasets import load_iris
```

```
data=pd.read_excel('/content/cpcb_dly_aq_tamil_nadu-2014.xlsx')
data.columns = data.columns.str.replace('\s+', '_')
data.head()
```

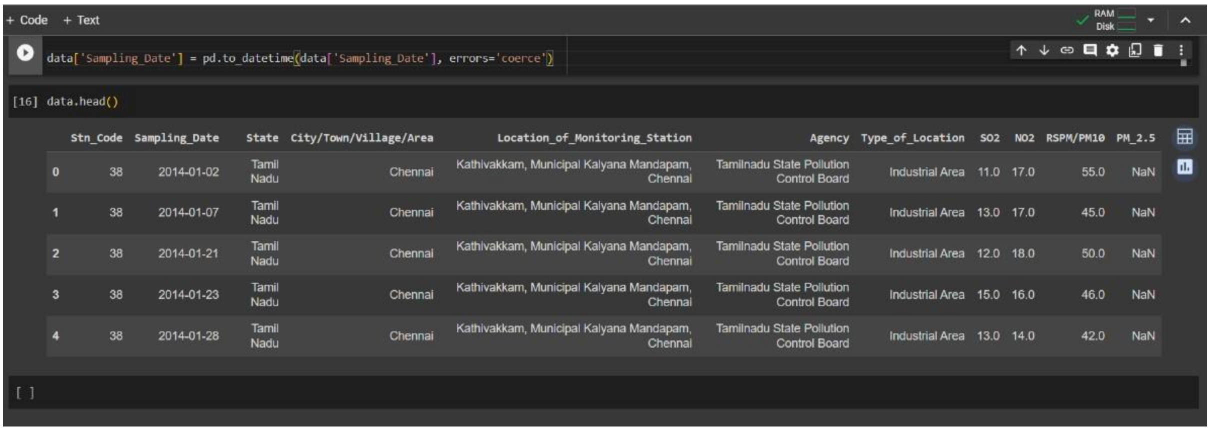
```
<ipython-input-4-6bccabf97634>:2: FutureWarning: The default value of regex will change from True to False in a future version.
  data.columns = data.columns.str.replace('\s+', '_')
```

	Stn_Code	Sampling_Date	State	City/Town/Village/Area	Location_of_Monitoring_Station	Agency	Type_of_Location	SO2	NO2	RSPM/PM10	PM_2.5
0	38	2014-01-02 00:00:00	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	2014-01-07 00:00:00	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN
2	38	21-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN
3	38	23-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN
4	38	28-01-14	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	NaN

```
[5] y = data.Sampling_Date
    x = data.SO2
```

DATA CONVERSION:

Convert the 'Sampling_Date' column to a datetime format, handling errors by using NaT.



```
data['Sampling_Date'] = pd.to_datetime(data['Sampling_Date'], errors='coerce')
```

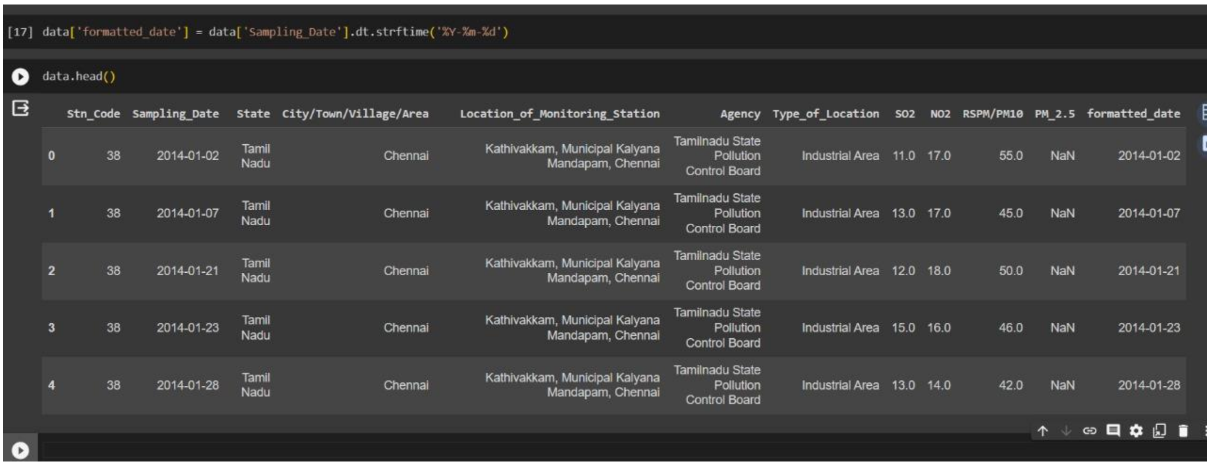
```
[16] data.head()
```

	Stn_Code	Sampling_Date	State	City/Town/Village/Area	Location_of_Monitoring_Station	Agency	Type_of_Location	SO2	NO2	RSPM/PM10	PM_2.5
0	38	2014-01-02	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	2014-01-07	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN
2	38	2014-01-21	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN
3	38	2014-01-23	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN
4	38	2014-01-28	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	NaN

```
[ ]
```

DATA FORMATTING:

Create a new 'formatted_date' column with a specific date format (YYYY-MM-DD)



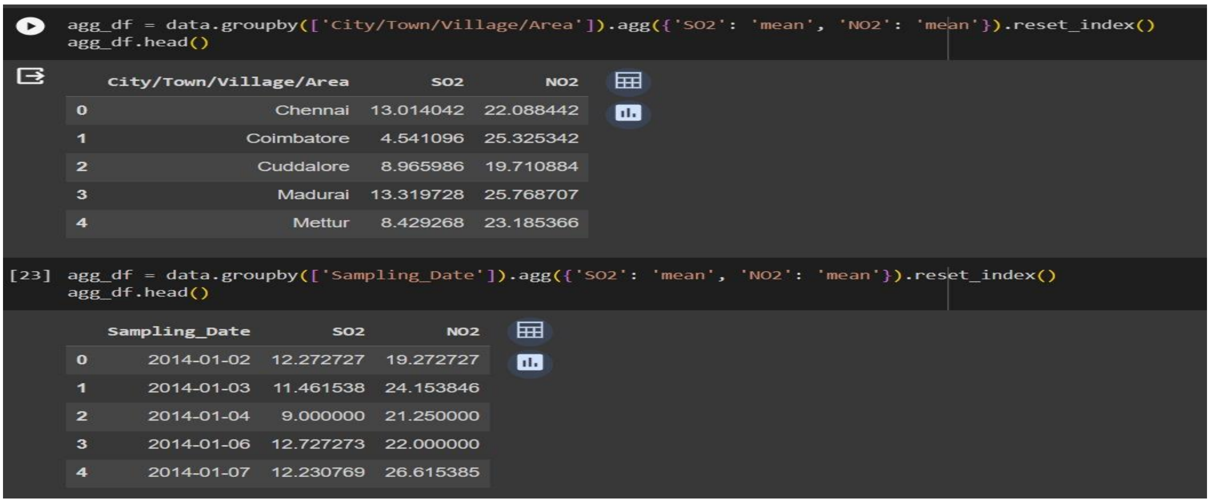
```
[17] data['formatted_date'] = data['Sampling_Date'].dt.strftime('%Y-%m-%d')
```

```
data.head()
```

	Stn_Code	Sampling_Date	State	City/Town/Village/Area	Location_of_Monitoring_Station	Agency	Type_of_Location	SO2	NO2	RSPM/PM10	PM_2.5	formatted_date
0	38	2014-01-02	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN	2014-01-02
1	38	2014-01-07	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN	2014-01-07
2	38	2014-01-21	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN	2014-01-21
3	38	2014-01-23	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN	2014-01-23
4	38	2014-01-28	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	NaN	2014-01-28

DATA AGGREGATION:

Aggregate data by 'City/Town/Village/Area' and 'Sampling_Date', calculating mean values for 'SO2' and 'NO2' for each group.



```
agg_df = data.groupby(['City/Town/Village/Area']).agg({'SO2': 'mean', 'NO2': 'mean'}).reset_index()
agg_df.head()
```

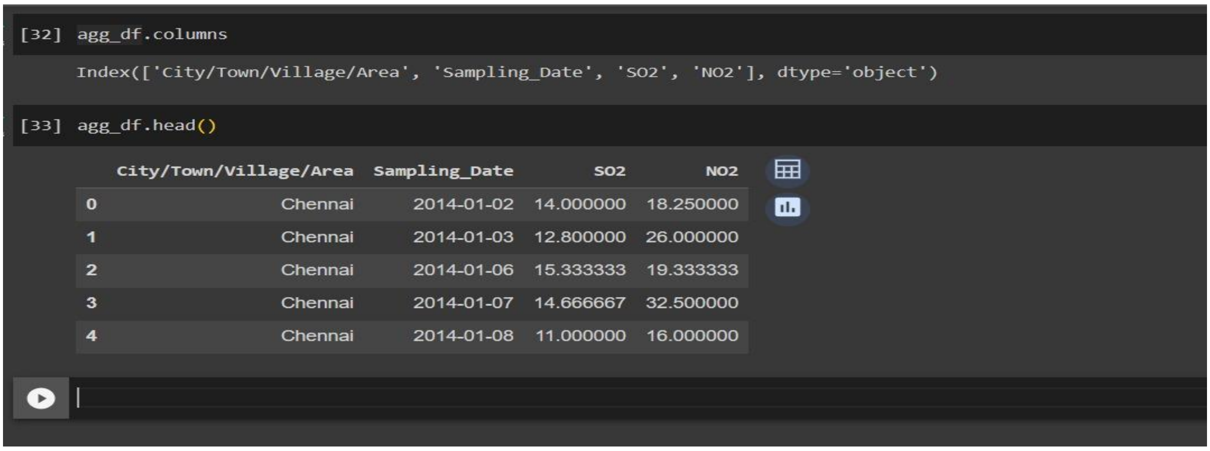
	City/Town/Village/Area	SO2	NO2
0	Chennai	13.014042	22.088442
1	Coimbatore	4.541096	25.325342
2	Cuddalore	8.965986	19.710884
3	Madurai	13.319728	25.768707
4	Mettur	8.429268	23.185366

```
[23] agg_df = data.groupby(['Sampling_Date']).agg({'SO2': 'mean', 'NO2': 'mean'}).reset_index()
     agg_df.head()
```

	Sampling_Date	SO2	NO2
0	2014-01-02	12.272727	19.272727
1	2014-01-03	11.461538	24.153846
2	2014-01-04	9.000000	21.250000
3	2014-01-06	12.727273	22.000000
4	2014-01-07	12.230769	26.615385

FILTERING UNWANTED DATA:

- Use double square brackets, like `[['NO2', 'SO2', 'City', 'Sampling_Date']]`.
- This creates a new DataFrame with only the selected columns.
- Useful for focusing on specific columns in analysis or data processing tasks



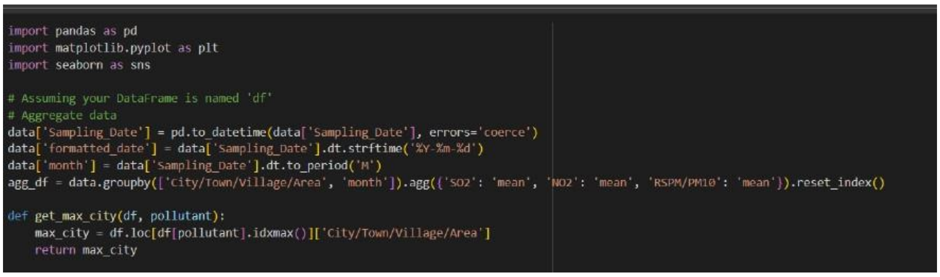
```
[32] agg_df.columns

     Index(['City/Town/Village/Area', 'Sampling_Date', 'SO2', 'NO2'], dtype='object')

[33] agg_df.head()
```

	City/Town/Village/Area	Sampling_Date	SO2	NO2
0	Chennai	2014-01-02	14.000000	18.250000
1	Chennai	2014-01-03	12.800000	26.000000
2	Chennai	2014-01-06	15.333333	19.333333
3	Chennai	2014-01-07	14.666667	32.500000
4	Chennai	2014-01-08	11.000000	16.000000

Implementation:



```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming your DataFrame is named 'df'
# Aggregate data
data['Sampling_Date'] = pd.to_datetime(data['Sampling_Date'], errors='coerce')
data['formatted_date'] = data['Sampling_Date'].dt.strftime('%Y-%m-%d')
data['month'] = data['Sampling_Date'].dt.to_period('M')
agg_df = data.groupby(['City/Town/Village/Area', 'month']).agg({'SO2': 'mean', 'NO2': 'mean', 'RSPM/PM10': 'mean'}).reset_index()

def get_max_city(df, pollutant):
    max_city = df.loc[df[pollutant].idxmax()]['City/Town/Village/Area']
    return max_city
```

This code creates a stacked area plot to visualize the air quality levels of three pollutants (RSPM/PM10, NO2, and SO2) over months in different cities. Here's a breakdown of the code:

1.Data Preparation

- The 'Sampling_Date' column is converted to a datetime format, and two additional columns ('formatted_date' and 'month') are created to store the formatted date and the month of the sampling.

- The data is then aggregated based on the city, month, and the mean values of the three pollutants (SO2, NO2, RSPM/PM10) are calculated.

python

agg_df = data.groupby(['City/Town/Village/Area', 'month']).agg({'SO2': 'mean', 'NO2': 'mean', 'RSPM/PM10': 'mean'}).reset_index()

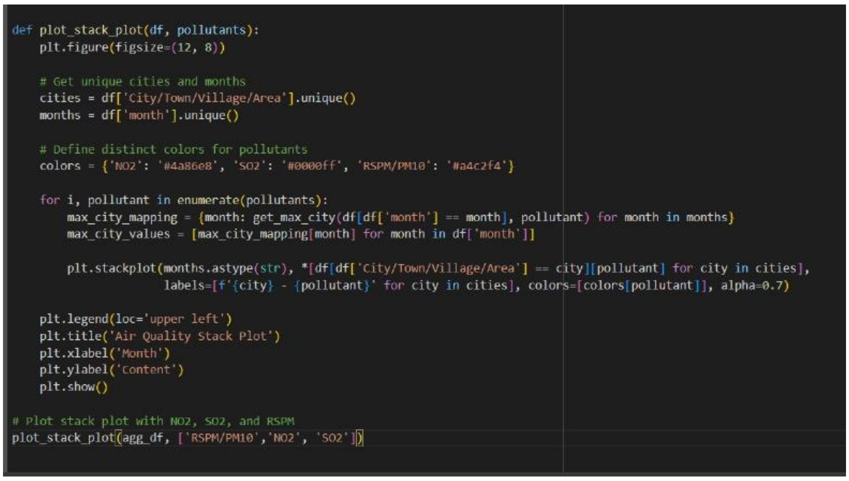
2.Function for Maximum City:

- A function (`get_max_city`) is defined to find the city with the maximum value for a given pollutant.

python

```
def get_max_city(df, pollutant):
    max_city = df.loc[df[pollutant].idxmax()]['City/Town/Village/Area']
    return max_city
```

Visualization Techniques:



```
def plot_stack_plot(df, pollutants):
    plt.figure(figsize=(12, 8))

    # Get unique cities and months
    cities = df['City/Town/Village/Area'].unique()
    months = df['month'].unique()

    # Define distinct colors for pollutants
    colors = {'NO2': '#4a86e8', 'SO2': '#0000ff', 'RSPM/PM10': '#a4c2f4'}

    for i, pollutant in enumerate(pollutants):
        max_city_mapping = {month: get_max_city(df[df['month'] == month], pollutant) for month in months}
        max_city_values = [max_city_mapping[month] for month in df['month']]

        plt.stackplot(months.astype(str), *[df[df['City/Town/Village/Area'] == city][pollutant] for city in cities],
                      labels=[f'{city} - {pollutant}' for city in cities], colors=[colors[pollutant]], alpha=0.7)

    plt.legend(loc='upper left')
    plt.title('Air Quality Stack Plot')
    plt.xlabel('Month')
    plt.ylabel('Content')
    plt.show()

# Plot stack plot with NO2, SO2, and RSPM
plot_stack_plot(agg_df, ['RSPM/PM10','NO2', 'SO2'])
```

1. Stacked Area Plot Function:

- The main function (`plot_stack_plot`) is defined to create the stacked area plot.

- Unique cities and months are extracted from the aggregated data.

- A dictionary (`colors`) is defined to map each pollutant to a specific color.

- The function iterates through the list of pollutants and creates a stacked area plot for each.

- The area plot is created using `stackplot` with alpha set for transparency.

- Legends, title, and labels are added to the plot.

python

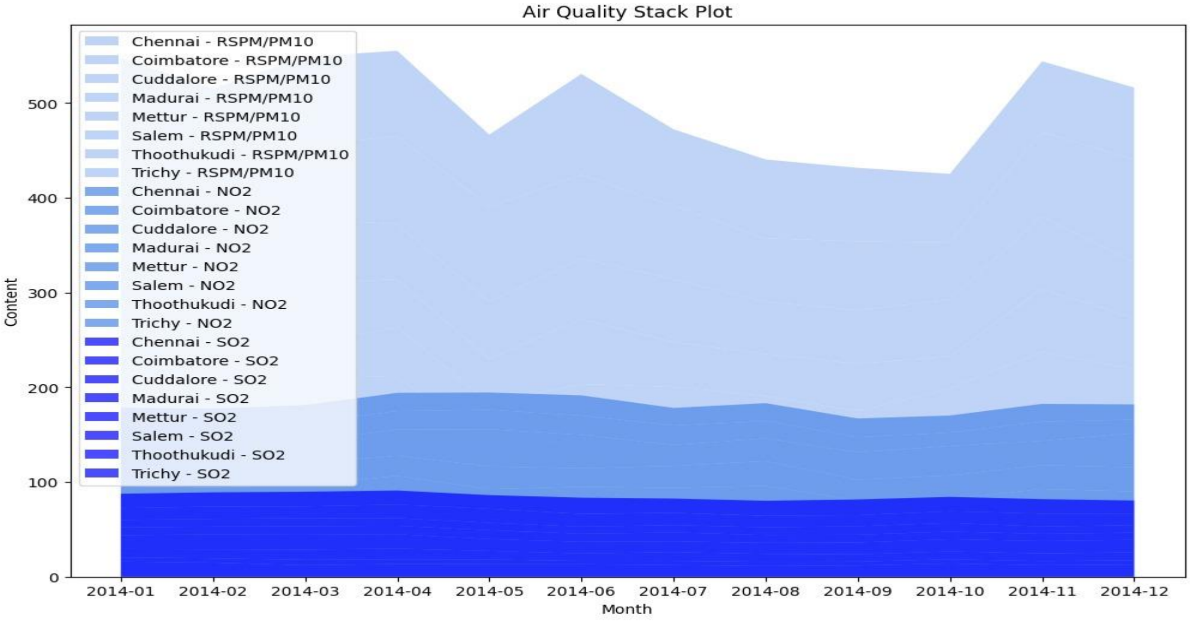
```
plot_stack_plot(agg_df, ['RSPM/PM10', 'NO2', 'SO2'])
```

2.Color Mapping:

- The colors for each pollutant are specified using hex color codes. In this case, RSPM/PM10 is represented by a shade of blue (`'#4a86e8'`), NO2 by a lighter blue (`'#0000ff`), and SO2 by a shade of green (`'#a4c2f4'`).

The resulting plot will show the air quality levels over months for each city, with different colors representing different pollutants. The transparency allows overlapping areas to be visible.

Insights:



RSPM/PM10:

From the month of febraury ,2014 rspm/PM10 levels seems increasing and it continues to grow steady till the mid days of the april and it declines like grooves till the early days of May and it found to be again increasing in the June month with slow pace and getting declined faster to the month of October and considers to be increasing at the earlier November and likely to be bend in the month of December.

SO2:

From the trends seen for SO2, the graph line tends to be increasing in the month of April and may unless it seems to be in slow pace, after the June, it oscillates now and then in between the months of July to December.

NO2:

Analysis done from the given trend says that no2 seems to be steady pace from January to December but likely to slightly bend down in the months of June and July.

Conclusion:

In conclusion, the analysis of air pollution trends and pollution levels in Tamil Nadu indicates a concerning pattern of increasing pollution levels, emphasizing the critical need for immediate environmental interventions and policy measures to ensure a sustainable and healthy future for the region.