**Air Quality Analysis in Tamilnadu**

**Team Members**

R.Kiruthiga

V.Kavitha

S.Kavitha

E.Praneetha

**Description:**

This script analyzes and visualizes air quality data for Tamil Nadu, India, using the `pandas` and `plotly` libraries. It loads the data from a CSV file, performs time series analysis, and creates a line plot for different air pollutants.

**Importing Libraries:**

The script begins by importing the necessary Python libraries for data analysis and visualization:

- `pandas`: A powerful data manipulation and analysis library.

- `plotly.express`: A high-level interface for creating a variety of complex plots.

- `plotly.io`: A library for configuring Plotly settings.

- `plotly.graph\_objects`: A library for creating complex and sophisticated plots.

**Setting Plotly Template:**

The default plotly template is set to "plotly\_white" to provide a white background for the plots, enhancing their visibility.

**Loading Data:**

The script loads air quality data from a CSV file located at "/kaggle/input/airqualityintamilnadu/cpcb\_dly\_aq\_tamil\_nadu-2014.csv" into a pandas DataFrame named `data`.

**Data Processing:**

- The 'Sampling Date' column in the DataFrame is converted to datetime format using the `pd.to\_datetime()` function, facilitating time-based analysis.

**Data Summary:**

A summary of the data, including descriptive statistics, is printed using the `describe()` method to provide insights into the general characteristics of the dataset.

**Creating Plotly Figure:**

An empty plotly figure (`fig`) is created, which will be populated with line plots representing the concentrations of different pollutants over time.

**Plotting Time Series Data:**

- Line plots are generated for each pollutant (SO2, NO2, and RSPM/PM10) using a loop to iterate through the list of pollutants.

- The 'Sampling Date' is set as the x-axis, and the corresponding pollutant concentration values are set as the y-axis for each plot.

**Updating Layout and Labels:**

The plot's layout is updated with a title, x-axis label, and y-axis label for better interpretation and understanding of the visualization.

**Displaying the Figure:**

The final plot is displayed using the `show()` function to visualize the time series data for different air pollutants.

**Conclusion:**

This script serves as a basic framework for loading, processing, and visualizing air quality data, enabling users to gain insights into pollutant concentrations over time. Adjustments can be made to customize the visualization further or to accommodate different datasets and requirements.

**Code**

import pandas as pd

import plotly.express as px

import plotly.io as pio

import plotly.graph\_objects as go

pio.templates.default = "plotly\_white"

data = pd.read\_csv("/kaggle/input/airqualityintamilnadu/cpcb\_dly\_aq\_tamil\_nadu-2014.csv")

print(data.head())

data['Sampling Date'] = pd.to\_datetime(data['Sampling Date'])

print(data.describe())

fig = go.Figure()

for pollutant in ['SO2','NO2','RSPM/PM10']:

    fig.add\_trace(go.Scatter(x=data['Sampling Date'], y=data[pollutant], mode='lines',

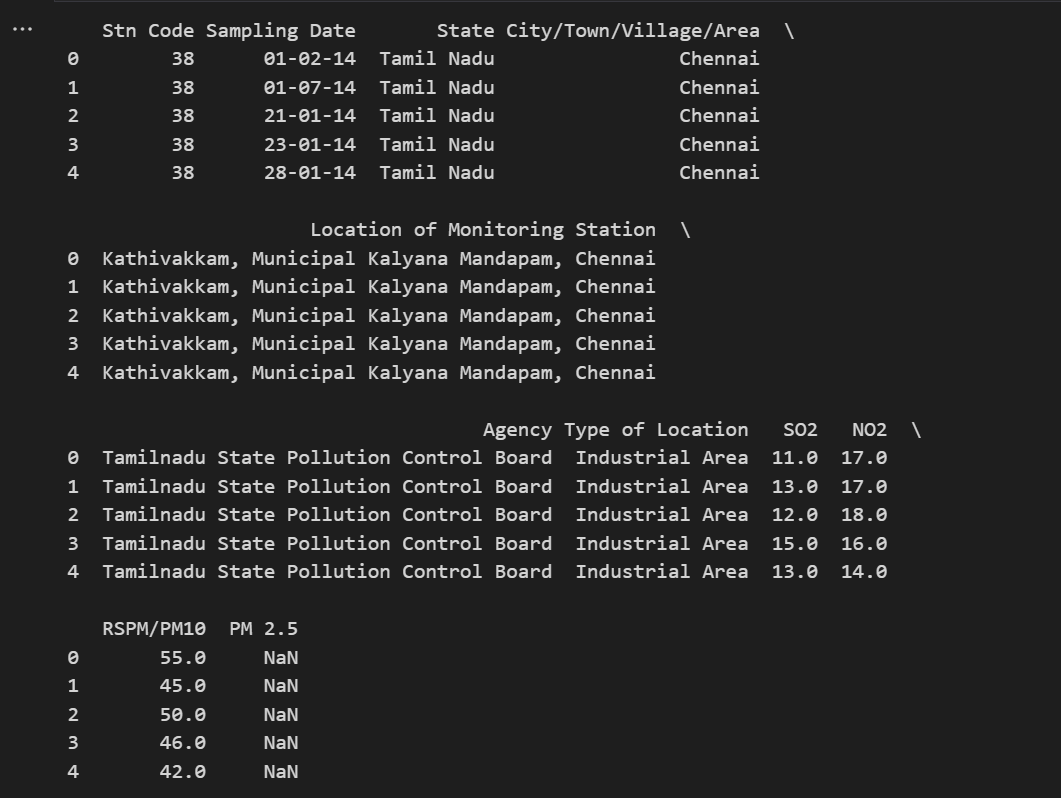
                             name=pollutant))

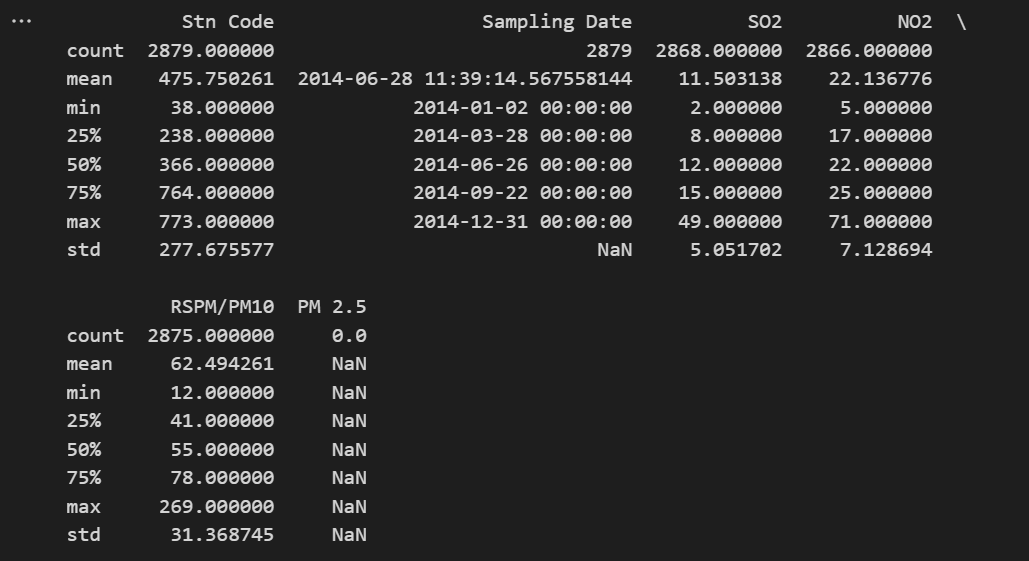
fig.update\_layout(title='Time Series Analysis of Air Pollutants in Tamilnadu',

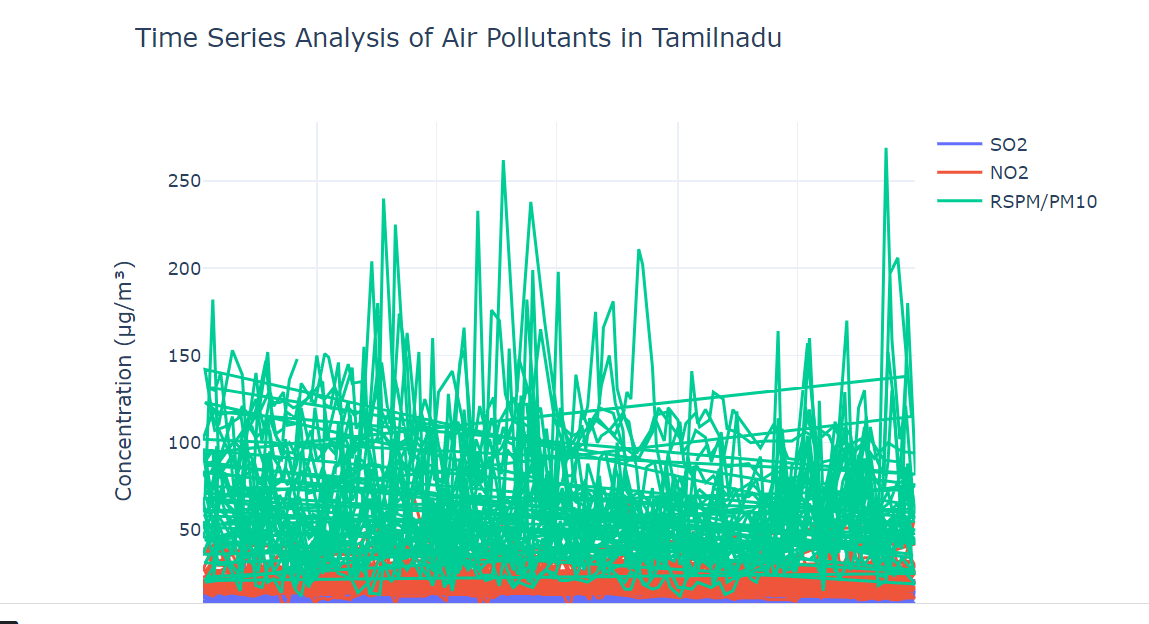
                  xaxis\_title='Date', yaxis\_title='Concentration (µg/m³)')

fig.show()

**Output:**







**Conclusion:**

This script serves as a basic framework for loading, processing, and visualizing air quality data, enabling users to gain insights into pollutant concentrations over time. Adjustments can be made to customize the visualization further or to accommodate different datasets and requirements.