**Air Quality Analysis in Tamilnadu**

**Team Members**

Priyadharshini.B

Logaharshini.S

Sandhiya.A

Kirthana.M

**Description:**

This script conducts a comprehensive analysis and visualization of air quality data for Tamil Nadu, India. It calculates the Air Quality Index (AQI) based on predefined breakpoints, categorizes the AQI into different levels, and explores various aspects of air quality through visualizations, including time-based trends, pollutant concentrations, and correlations.

**Data Preprocessing and Calculation Functions:**

1. **Define AQI Breakpoints and Calculation Functions:**

This section defines the AQI breakpoints and two functions: `calculate\_aqi` to calculate the AQI for each pollutant and `calculate\_overall\_aqi` to determine the overall AQI based on the maximum AQI value among all pollutants.

1. **Categorize AQI:**

The script categorizes the calculated AQI values into predefined categories using the `categorize\_aqi` function.

**Visualizations:**

1. **AQI Over Time:**

This section visualizes the AQI over time using a bar plot, showing the variation of the AQI in Tamil Nadu in January.

1. **AQI Category Distribution Over Time:**

This section presents a histogram representing the distribution of AQI categories over time.

1. **Pollutant Concentrations:**

It displays a donut plot representing the concentration of different pollutants in Tamil Nadu.

1. **Correlation Between Pollutants:**

This section showcases the correlation matrix between the pollutants, providing insights into their relationships.

**5.Hourly Average AQI Trends:**

It illustrates the hourly average AQI trends for Tamil Nadu in January 2023 using a line plot.

1. **Average AQI by Day of the Week:**

This section presents a bar plot indicating the average AQI based on the day of the week, providing insights into weekly patterns of air quality.

**Key Features and Interpretations:**

- The script calculates the AQI to assess overall air quality and presents it through various visualizations.

- It identifies hourly and weekly patterns in air quality, enabling users to understand variations and potential trends.

- Pollutant concentrations are visualized to highlight the relative contributions of different pollutants to overall air quality.

- Correlation matrices between pollutants are included, providing insights into potential interdependencies between different pollutants.

**Potential Use Cases:**

**-** Environmental agencies and policymakers can utilize this analysis to make informed decisions about air quality management and regulation.

- Researchers and analysts can use these visualizations to identify patterns, trends, and potential areas for further investigation in the field of air quality research.

**Code:**

# Define AQI breakpoints and corresponding AQI values

aqi\_breakpoints = [

    (0, 12.0, 50), (12.1, 35.4, 100), (35.5, 55.4, 150),

    (55.5, 150.4, 200), (150.5, 250.4, 300), (250.5, 350.4, 400),

    (350.5, 500.4, 500)

]

def calculate\_aqi(pollutant\_name, concentration):

    for low, high, aqi in aqi\_breakpoints:

        if low <= concentration <= high:

            return aqi

    return None

def calculate\_overall\_aqi(row):

    aqi\_values = []

    pollutants = ['SO2','NO2','RSPM/PM10']

    for pollutant in pollutants:

        aqi = calculate\_aqi(pollutant, row[pollutant])

        if aqi is not None:

            aqi\_values.append(aqi)

    return max(aqi\_values)

# Calculate AQI for each row

data['AQI'] = data.apply(calculate\_overall\_aqi, axis=1)

# Define AQI categories

aqi\_categories = [

    (0, 50, 'Good'), (51, 100, 'Moderate'), (101, 150, 'Unhealthy for Sensitive Groups'),

    (151, 200, 'Unhealthy'), (201, 300, 'Very Unhealthy'), (301, 500, 'Hazardous')

]

def categorize\_aqi(aqi\_value):

    for low, high, category in aqi\_categories:

        if low <= aqi\_value <= high:

            return category

    return None

# Categorize AQI

data['AQI Category'] = data['AQI'].apply(categorize\_aqi)

print(data.head())

def categorize\_aqi(aqi\_value):

    for low, high, category in aqi\_categories:

        if low <= aqi\_value <= high:

            return category

    return None

# Categorize AQI

data['AQI Category'] = data['AQI'].apply(categorize\_aqi)

print(data.head())

# AQI over time

fig = px.bar(data, x="Sampling Date", y="AQI",

             title="AQI of Tamilnadu in January")

fig.update\_xaxes(title="Date")

fig.update\_yaxes(title="AQI")

fig.show()

fig = px.histogram(data, x="Sampling Date",

                    color="AQI Category",

                    title="AQI Category Distribution Over Time")

fig.update\_xaxes(title="Date")

fig.update\_yaxes(title="Count")

fig.show()

# Define pollutants and their colors

pollutants = ['SO2','NO2','RSPM/PM10']

pollutant\_colors = px.colors.qualitative.Plotly

# Calculate the sum of pollutant concentrations

total\_concentrations = data[pollutants].sum()

# Create a DataFrame for the concentrations

concentration\_data = pd.DataFrame({

    "Pollutant": pollutants,

    "Concentration": total\_concentrations

})

# Create a donut plot for pollutant concentrations

fig = px.pie(concentration\_data, names="Pollutant", values="Concentration",

             title="Pollutant Concentrations in Tamilnadu",

             hole=0.4, color\_discrete\_sequence=pollutant\_colors)

# Update layout for the donut plot

fig.update\_traces(textinfo="percent+label")

fig.update\_layout(legend\_title="Pollutant")

# Show the donut plot

fig.show()

# Correlation Between Pollutants

correlation\_matrix = data[pollutants].corr()

fig = px.imshow(correlation\_matrix, x=pollutants,

                 y=pollutants, title="Correlation Between Pollutants")

fig.show()

# Extract the hour from the date

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

# Calculate hourly average AQI

hourly\_avg\_aqi = data.groupby('Hour')['AQI'].mean().reset\_index()

# Create a line plot for hourly trends in AQI

fig = px.line(hourly\_avg\_aqi, x='Hour', y='AQI',

              title='Hourly Average AQI Trends in Tamilnadu (Jan 2023)')

fig.update\_xaxes(title="Hour of the Day")

fig.update\_yaxes(title="Average AQI")

fig.show()

# Average AQI by Day of the Week

data['Day\_of\_Week'] = data['Sampling Date'].dt.day\_name()

average\_aqi\_by\_day = data.groupby('Day\_of\_Week')['AQI'].mean().reindex(['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'])

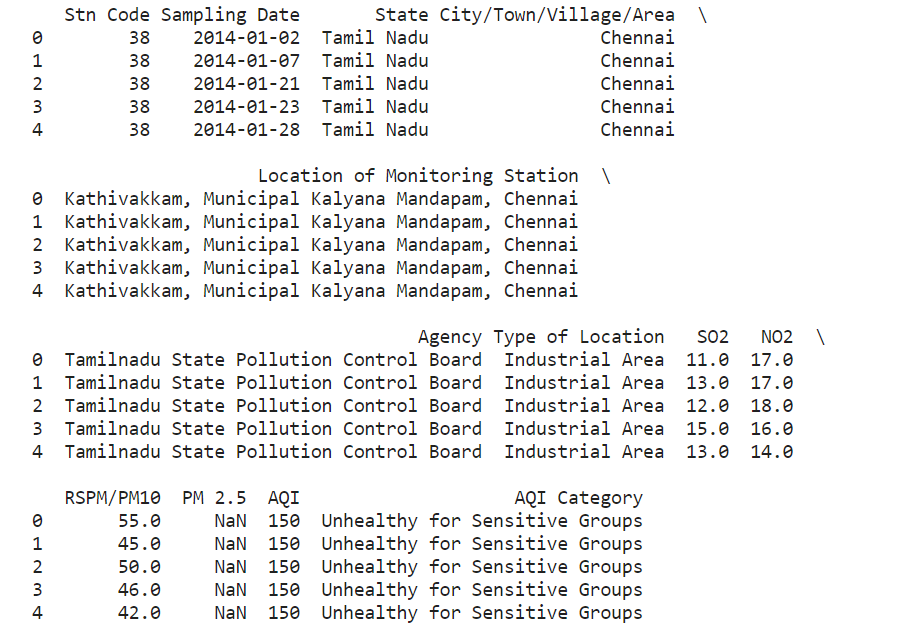
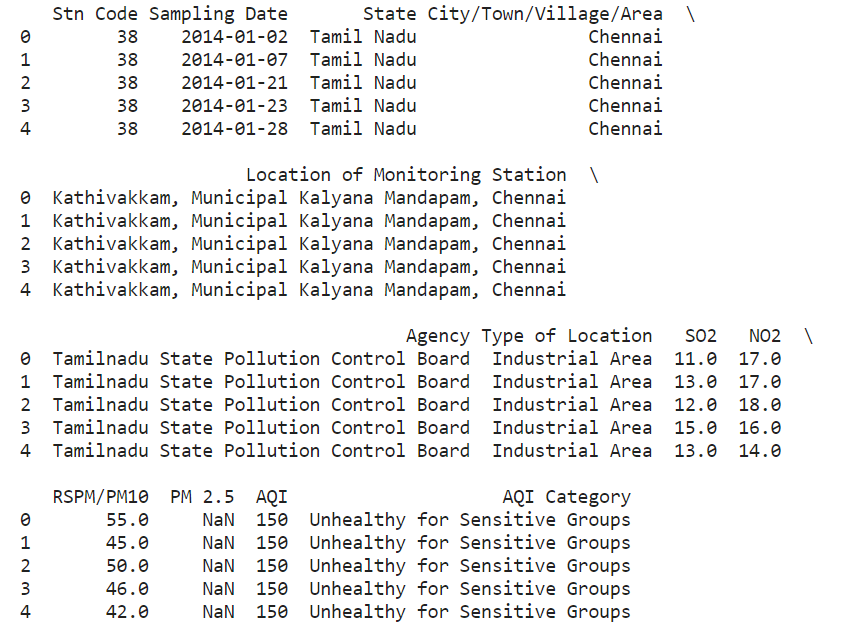
fig = px.bar(average\_aqi\_by\_day, x=average\_aqi\_by\_day.index, y='AQI',

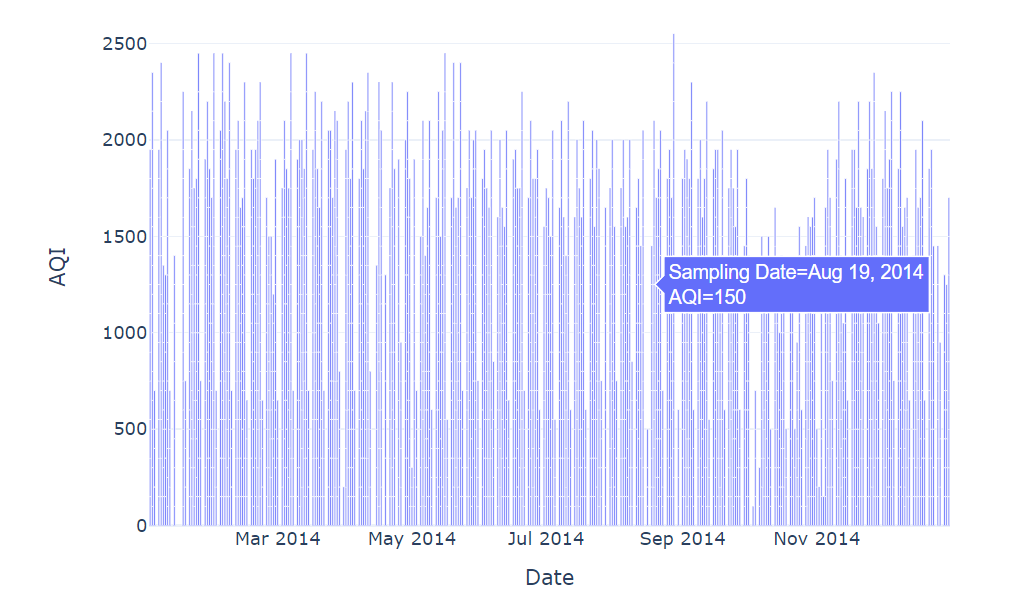
              title='Average AQI by Day of the Week')

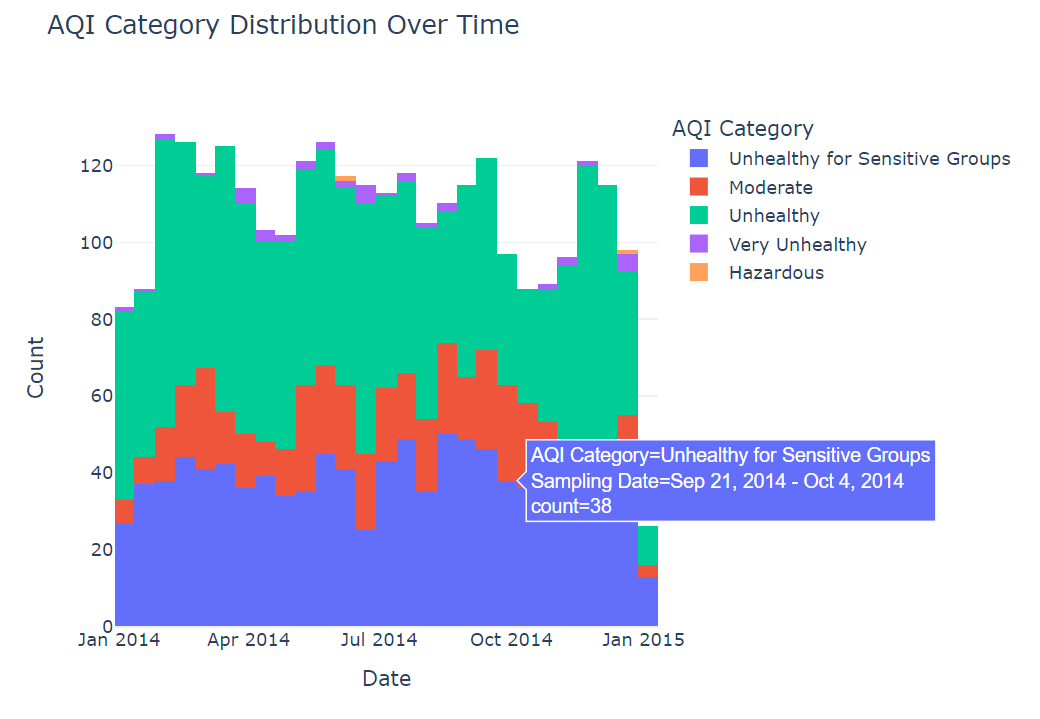
fig.update\_xaxes(title="Day of the Week")

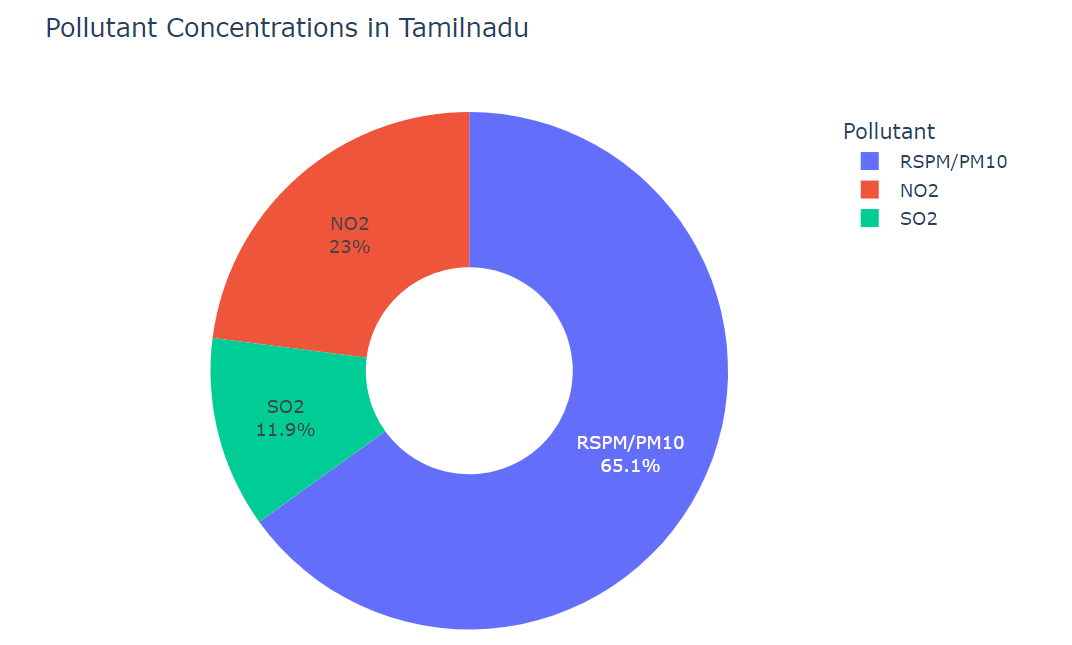
fig.update\_yaxes(title="Average AQI")

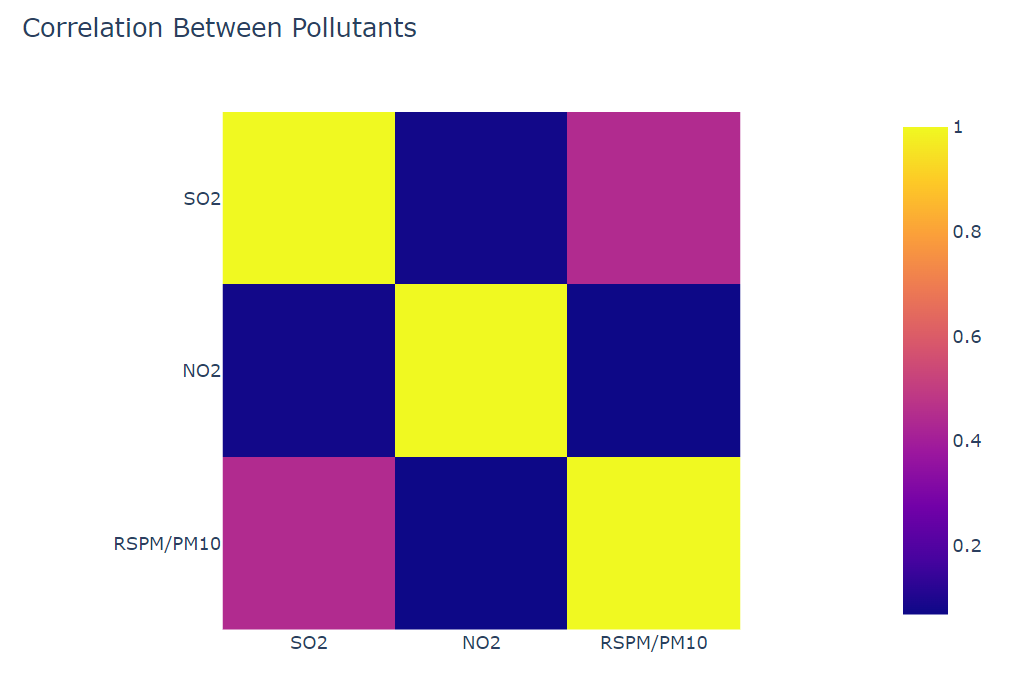
fig.show()

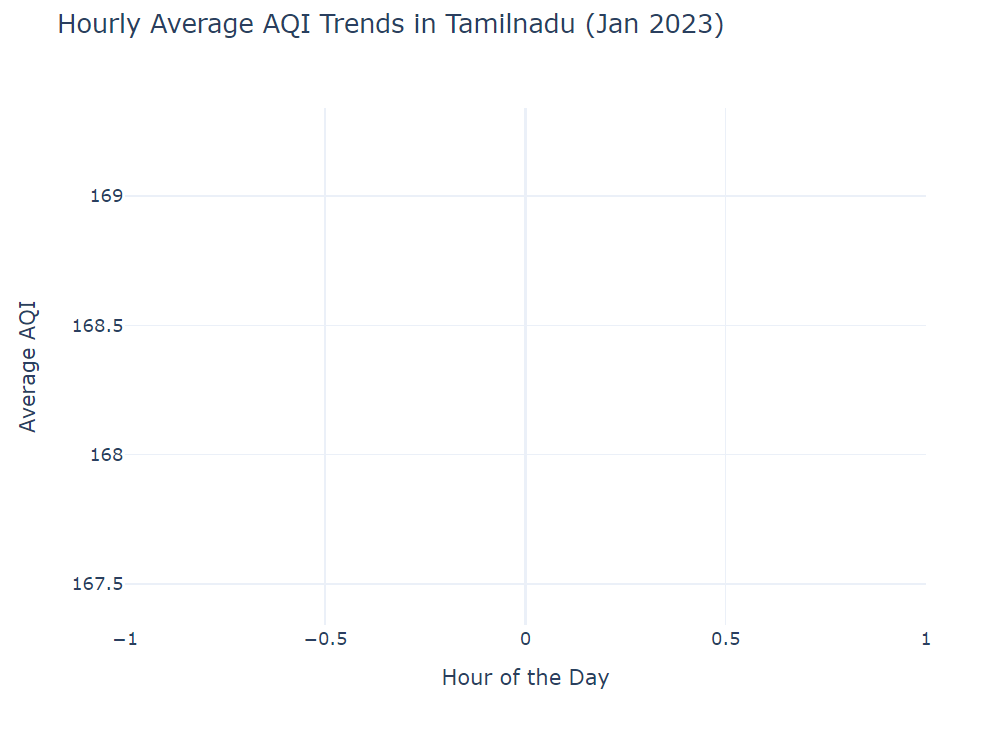
**Output:**

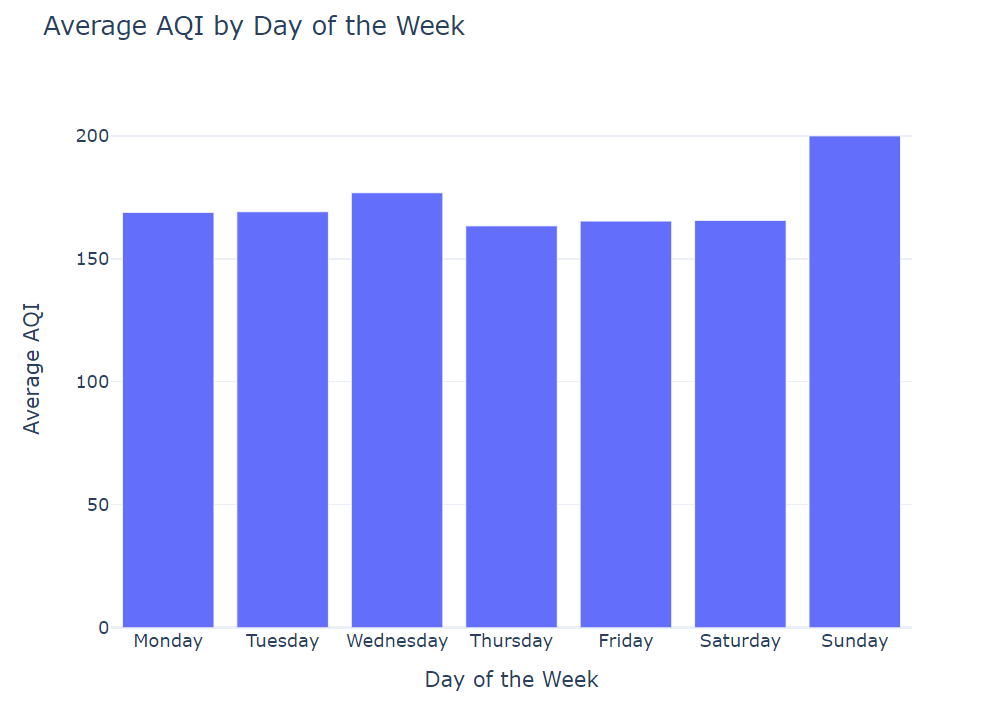
****

****

****

****

****

****

**Conclusion:**

In conclusion, the provided Python script offers a comprehensive analysis and visualization of air quality data for Tamil Nadu, India. By calculating the Air Quality Index (AQI) and categorizing it into distinct levels, the script provides valuable insights into the overall air quality status. Through a series of visualizations, including bar plots, histograms, pie charts, and correlation matrices, the script effectively communicates key patterns and trends in air quality over time.

The script's ability to showcase hourly and weekly trends in AQI values enables a deeper understanding of how air quality fluctuates throughout the day and across different days of the week. Additionally, the visualization of pollutant concentrations provides a clear representation of the relative contributions of different pollutants to overall air quality, aiding in identifying critical areas for pollution control and regulation.

With its capacity to present complex data in an accessible and intuitive manner, this script serves as a valuable tool for environmental agencies, policymakers, researchers, and analysts seeking to make data-driven decisions in the field of air quality management and research. By leveraging the insights provided by the script, stakeholders can devise effective strategies and policies to address air quality concerns and promote a healthier environment for the residents of Tamil Nadu.