AIR QUALITY ASSESSMENT-TAMILNADU

TEAM MEMBER

962221106074: S.MOHAMED SHAMEEM

PHASE-5

DOCUMENTATION AND SUBMISSION:

INTRODUCTION:

Data analysis is a critical component of understanding and addressing air quality issues in Tamil Nadu. It involves the systematic examination of collected data to extract meaningful insights and patterns. In the context of air quality assessment, data analysis plays a pivotal role in evaluating the current state of the atmosphere, identifying sources of pollution, and formulating effective mitigation strategies.This analysis encompasses various parameters, including concentrations of pollutants, meteorological conditions, and their correlations. The objective is to provide a comprehensive overview of air quality trends, seasonal variations, and areas of concern within the state. Additionally, it helps in assessing compliance with regulatory standards and evaluating the effectiveness of existing pollution control measures.Through advanced analytical techniques, stakeholders can gain valuable insights into the complex interactions between pollutants, meteorology, and human activities. This knowledge forms the basis for evidence-based decision-making and targeted interventions to improve air quality and protect public health in Tamil Nadu. In the following sections, we will delve into specific aspects of data analysis, including pollutant concentration trends, source apportionment, and correlation with meteorological parameters.

OBJECTIVE:

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 visualized techniques, and creating a Predictive model using Python and relevant libraries. Air Quality Analysis Objectives.

DATASET:

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

DESIGN THINKING:

1) Air Quality Trends:

This objective involves studying historical data on air quality parameters, such as pollutant levels, meteorological conditions, and emission sources, to understand how air quality has changed over time. This analysis helps identify long-term trends, seasonal variations, and potential contributing factors.

2) Identifying Pollution Hotspots:

This objective focuses on pinpointing specific geographic areas or locations where air pollution levels consistently exceed acceptable limits. By identifying pollution hotspots, authorities can prioritize targeted interventions and regulatory measures to reduce pollution in these areas.

3)Building a Predictive Model for RSPM/PM10 Levels:

This objective involves developing a statistical or machine learning model that can forecast levels of Respirable Suspended Particulate Matter (RSPM) or Particulate Matter with a diameter of 10 micrometers or less (PM10). This model typically uses historical data, meteorological information, and other relevant variables to make predictions, aiding in proactive pollution management and public health planning. These objectives collectively contribute to better air quality management and the protection of public environment.

ANALYSIS APPROACH

1.Load Data:

✓Obtain the air quality data from reliable sources, which may include government agencies, environmental organizations, or research institutions.

✓The data may be available in various formats like CSV, Excel, or specialized formats like JSON orXML.

2. Preprocess Data:

✓Data Cleaning : Handle missing values: Replace or interpolate missing data points if possible, or consider removing incomplete records. Check for outliers and anomalies: Identify and address any data points that deviate significantly from the rest of the dataset.

✓Data Transformation: Convert data types: Ensure that variables are in the correct data type (e.g., numerical, categorical, date).Normalize or standardize data if necessary to bring it to a consistent scale. ✓Feature Engineering: Create new features that might be useful for analysis (e.g., derived variables, aggregates).Encode categorical variables using techniques like one-hot encoding.

3.Data Analysis:

✓Descriptive Statistics: Calculate basic statistics like mean, median, standard deviation, and percentiles to understand the distribution of the data.

✓Time Series Analysis (if applicable): Explore temporal patterns, trends, and seasonality using techniques like moving averages, decomposition, or autocorrelation.

✓Correlation Analysis: Identify relationships between different variables, especially pollutants, meteorological conditions, and geographical features.

✓Spatial Analysis (if applicable): Use GIS tools or libraries to analyze spatial patterns and relationships. 4.Build Predictive Models (Optional):

✓If you plan to build predictive models, split the data into training and testing sets.

✓Select an appropriate modeling technique (e.g., regression, time series forecasting, machine learning algorithms) and train the model.

ALGORITHM:

1. **Data Collection:**
   Gather air quality data from various sources like sensors, weather stations, or government agencies. This data typically includes parameters like PM2.5, PM10, NO2, CO, etc.
2. **Data Preprocessing:**
   Clean and prepare the data for analysis. This involves tasks like handling missing values, outlier detection, and normalization.
3. **Feature Engineering:**
   Extract relevant features from the data. These could be time of day, weather conditions, geographical coordinates, etc., that may impact air quality.
4. **Exploratory Data Analysis (EDA):**
   Analyze and visualize the data to gain insights. This can help you understand the relationships between different variables**.**
5. **Model Selection:**
   Choose a machine learning algorithm suitable for your task. For air quality assessment, regression models like Linear Regression, Decision Trees, Random Forest, or more advanced models like Neural Networks could be considered**.**
6. **Model Training:**
   Split the data into training and testing sets. Train the model on the training data and validate it on the testing data to ensure it generalizes well.
7. **Model Evaluation:**
   Use appropriate metrics (e.g., Mean Absolute Error, Root Mean Squared Error) to evaluate the performance of your model.
8. **Model Tuning:**
   Adjust hyperparameters or try different algorithms to improve performance**.**
9. **Deployment:**
   Once satisfied with the model's performance, deploy it in a suitable environment. This could be an app, a web service, or an integrated system.
10. **Monitoring and Maintenance:**
    Continuously monitor the model's performance and update it as needed. Air quality conditions can change over time.

LIBRARIES USED FOR PREPROCESSING:

NumPy: **NumPy stands for Numerical Python.It is a fundamental package for numerical computations in Python.It provides support for arrays and matrices, as well as a large collection of high-level mathematical functions to operate on these data structures.NumPy is widely used in scientific and engineering applications for tasks involving numerical operations.**

Pandas: **Pandas is a data manipulation and analysis library. It provides data structures like Series (1-dimensional) and DataFrame (2-dimensional), which are highly efficient and designed for working with structured data.Pandas allows for easy data ingestion, cleaning, transformation, and analysis.**

Seaborn: **Seaborn is a statistical data visualization library based on Matplotlib.It provides a high-level interface for creating informative and attractive statistical graphics.Seaborn simplifies the process of creating complex visualizations and is especially useful for exploring relationships between variables in datasets.**

```
In [1]: #import required libreries

In [2]: import numpy as np

In [3]: import seaborn as sns

In [4]: import pandas as pd
```

DATASET COLLECTED: RECENT DATA(5TH AUGUST 2023)

FILE HOME INSERT PAGE LAYOUT FORMULAS DATA REVIEW VIEW

PROTECTED VIEW Be careful—files from the Internet can contain viruses. Unless you need to edit, it's safer to stay in Protected View. Enable Editing

I13 Satisfactory

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	S.No	District (Location)		S02	N02	co	PM2.5	PMIO	AQI Index	AQI	Prominent				
2										Value	Pollutant				
3	1	Ariyalur		11	14	0.4	16	37	Good	37	PMIO				
4	2	Chengalpattu (Vand		13	18	0.8	20	96	Satisfacto	96	PMIO				
5	3		Kodungaiy	3	16	0.7	17	68	Satisfacto	76	pM10				
6	4		Koyambe	4	13	0.4	28	67	Satisfacto	62	pMIO				
7	5	Chennai	perungud	3	23	0.5	17	89	Satisfacto	89	PMIO				
8	6		Royapura	3	24	0.6	19	72	Satisfacto	72	PMIO				
9	7		Kuruchi-S	6	12	0.3	21	38	Good	38	PMIO				
10		Coimbatore													
11	8		PSG Colla	4	9	0.2	10	33	Good	33	PMI O				
12	9		Semmend	6	12	0.3	20	29	Good	29	PMIO				
13	10	Cuddalore	SIPCOT	17	13	0.5	34	43	Satisfacto	43	PMIO				
14	11	Dindigul						ND							
15	12	Hosur		6	3	o.i	24	45		45	PMIO				
16	13	Kanchipuram		1	2	o.i	24	49	Good	49	pMIO				
17	14	Karur		16	19	0.6	29	43	Good	43	ptv410				
18	15	Madurai		2	4	0.5	20	41	Good	41	PMIO				
19	16	Nagapattinam		17	19	0.5	24	15		24	PM2.5				
20	17	Namakkal						ND							
21	18	Ooty		13	16	0.3	12	30	Good	30	PMIO				
22	19	Perundurai		9	14	0.5	23	38	Good	38	PMIO				
23	20	Pudukkottai		24	26	0.9	21	49	Good	49	pMIO				
24	21	Ramanathapuram		7	3	0.4	11	51	Satisfacto	51	PMIO				
25	22	Ranipet, SIPCOT		20	21	0.3	11	35	Good	35	PMIO				
26	23	Salem		12	16	0.8	22	39	Good	39	PMIO				

LOADING OF DATA IN PYTHON:

```
In [1]: #import required libraries
```

```
In [2]: import pandas as pd
```

```
In [3]: import numpy as np
```

```
In [4]: import seaborn as sns
```

```
In [11]: data = pd.read_excel("C:\\Users\\pc\Pictures\\New folder\\airquality.xlsx")
```

```
In [12]: data.head()
```

Out[12]:

	S.No	District (Location)	Unnamed: 2	S02	N02	co	PM2.5	PMIO	AQI Index	AQI	Prominent
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Value	Pollutant
1	1.0	Ariyalur	NaN	11.0	14	0.4	16	37	Good	37	PMIO
2	2.0	Chengalpattu (Vandalur)	NaN	13.0	18	0.8	20	96	Satisfactory	96	PMIO
3	3.0	NaN	Kodungaiyur	3.0	16	0.7	17	68	Satisfactory	76	pM10

```
In [13]: pd.read_excel("C:\\Users\\pc\Pictures\\New folder\\airquality.xlsx")
```

Out[13]:

	S.No	District (Location)	Unnamed: 2	S02	N02	co	PM2.5	PMIO	AQI Index	AQI	Prominent
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Value	Pollutant
1	1.0	Ariyalur	NaN	11.0	14	0.4	16	37	Good	37	PMIO
2	2.0	Chengalpattu (Vandalur)	NaN	13.0	18	0.8	20	96	Satisfactory	96	PMIO
3	3.0	NaN	Kodungaiyur	3.0	16	0.7	17	68	Satisfactory	76	pM10
4	4.0	NaN	Koyambedu	4.0	13	0.4	28	67	Satisfactory	62	pMIO
5	5.0	Chennai	perungudi	3.0	23	0.5	17	89	Satisfactory	89	PMIO
6	6.0	NaN	Royapuram	3.0	24	0.6	19	72	Satisfactory	72	PMIO
7	7.0	NaN	Kuruchi-SIDCO	6.0	12	0.3	21	38	Good	38	PMIO
8	NaN	Coimbatore	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9	8.0	NaN	PSG Collage	4.0	9	0.2	10	33	Good	33	PMI O
10	9.0	NaN	Semmendalam	6.0	12	0.3	20	29	Good	29	PMIO
11	10.0	Cuddalore	SIPCOT	17.0	13	0.5	34	43	Satisfactory	43	PMIO
12	11.0	Dindigul	NaN	NaN	NaN	NaN	NaN	ND	NaN	NaN	NaN

PREPROCESSING OF DATA:

```
In [59]: data2
```

Out[59]:

	S.No	city	Unnamed: 2	S02	N02	co	PM2.5	PMIO	AQI Index	AQI	Prominent
0	NaN	22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	26	Pollutant
1	1.0	0	NaN	11.0	14	0.4	16	37	Good	8	PMIO
2	2.0	23	NaN	13.0	18	0.8	20	96	Satisfactory	23	PMIO
3	3.0	22	Kodungaiyur	3.0	16	0.7	17	68	Satisfactory	20	pM10
4	4.0	22	Koyambedu	4.0	13	0.4	28	67	Satisfactory	17	pMIO
5	5.0	14	perungudi	3.0	23	0.5	17	89	Satisfactory	22	PMIO
6	6.0	22	Royapuram	3.0	24	0.6	19	72	Satisfactory	18	PMIO
7	7.0	22	Kuruchi-SIDCO	6.0	12	0.3	21	38	Good	9	PMIO
8	NaN	25	NaN	NaN	NaN	NaN	NaN	NaN	NaN	24	NaN
9	8.0	22	PSG Collage	4.0	9	0.2	10	33	Good	6	PMI O
10	9.0	22	Semmendalam	6.0	12	0.3	20	29	Good	3	PMIO

```
In [56]: dist=(data2['city'])
         distset=set(dist)
         dd=list(distset)
         dict0fwords = {dd[i] :i for i in range(0,len(dd))}
         data2['city']=data2['city'].map(dict0fwords)
```

```
In [57]: dist=(data2['AQI'])
         distset=set(dist)
         dd=list(distset)
         dict0fwords = {dd[i] :i for i in range(0,len(dd))}
         data2['AQI']=data2['AQI'].map(dict0fwords)
```

```
In [58]: data2["AQI"]=data2["AQI"].fillna(data2["AQI"].mean())
```

```
In [59]: data2
```

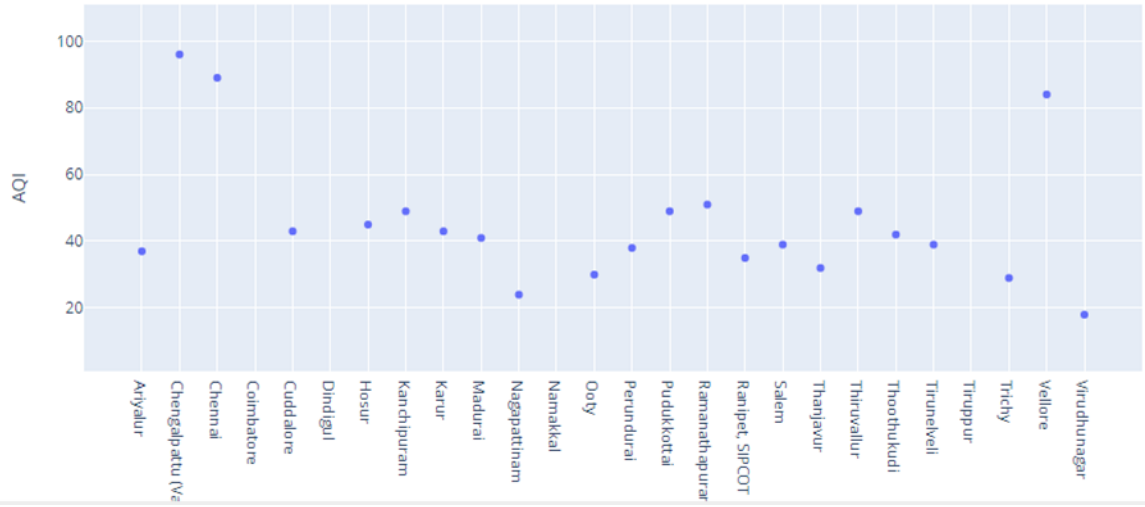
In [59]:
```
data2
```

Out[59]:

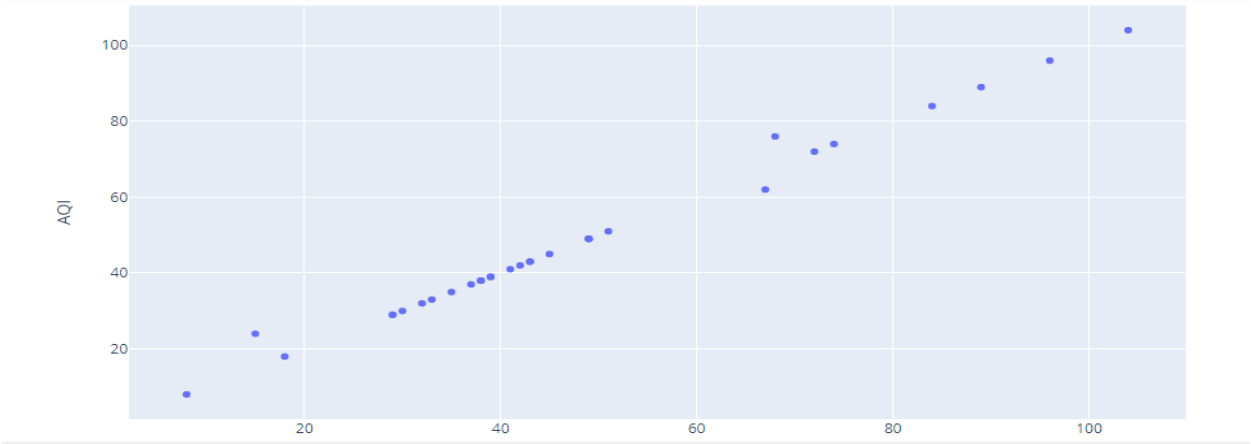
	S.No	city	Unnamed: 2	S02	N02	co	PM2.5	PMIO	AQI Index	AQI	Prominent
0	NaN	22	NaN	NaN	NaN	NaN	NaN	NaN	NaN	26	Pollutant
1	1.0	0	NaN	11.0	14	0.4	16	37	Good	8	PMIO
2	2.0	23	NaN	13.0	18	0.8	20	96	Satisfactory	23	PMIO
3	3.0	22	Kodungaiyur	3.0	16	0.7	17	68	Satisfactory	20	pM10
4	4.0	22	Koyambedu	4.0	13	0.4	28	67	Satisfactory	17	pMIO
5	5.0	14	perungudi	3.0	23	0.5	17	89	Satisfactory	22	PMIO
6	6.0	22	Royapuram	3.0	24	0.6	19	72	Satisfactory	18	PMIO
7	7.0	22	Kuruchi-SIDCO	6.0	12	0.3	21	38	Good	9	PMIO
8	NaN	25	NaN	NaN	NaN	NaN	NaN	NaN	NaN	24	NaN
9	8.0	22	PSG Collage	4.0	9	0.2	10	33	Good	6	PMI O
10	9.0	22	Semmendalam	6.0	12	0.3	20	29	Good	3	PMIO

VISUALIZATION WITH IMPORTS:

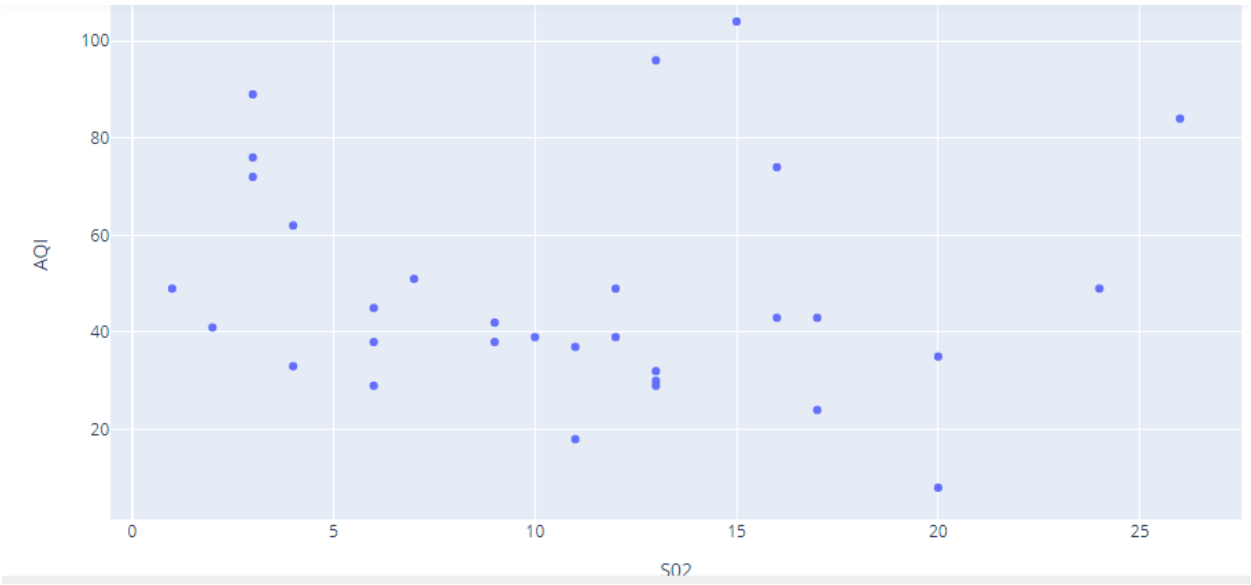
In [62]:
```
import plotly.express as px

#plotting the bubble chart
fig=px.scatter(data, x="city" , y="AQI")

#showing the plot
fig.show()
```



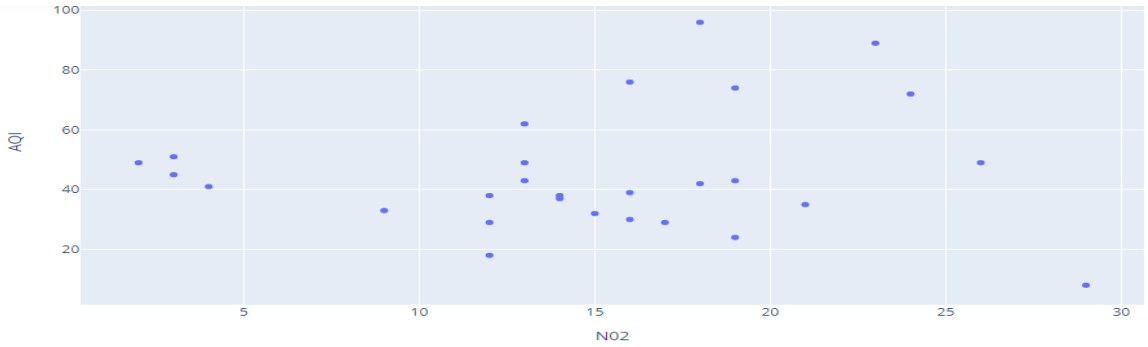
In [66]:
```
import plotly.express as px

#plotting the bubble chart
fig2=px.scatter(data, x="PMIO",y="AQI")

#showing the plot
fig2.show()
```



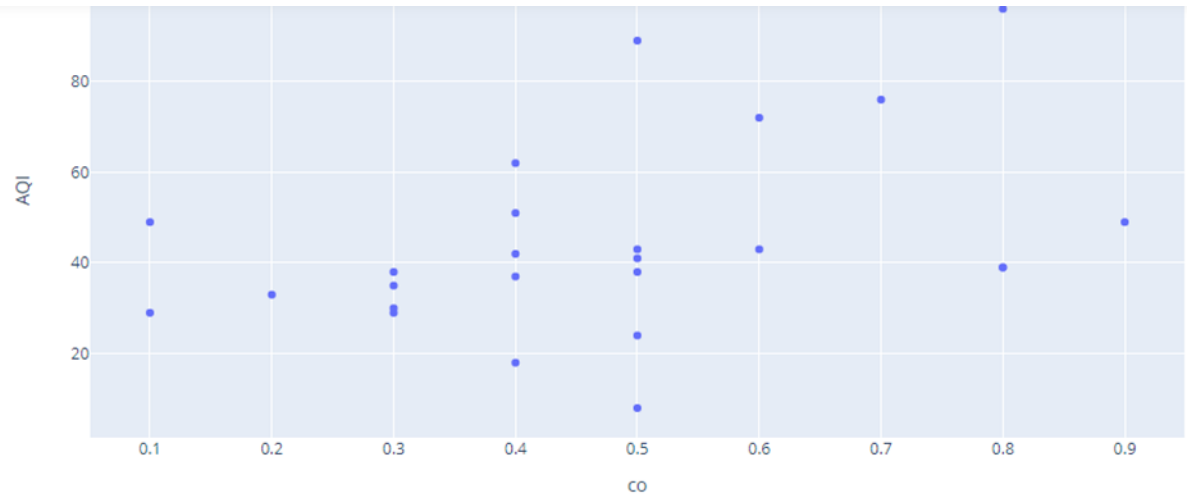
In [68]:
```
import plotly.express as px

#plotting the bubble chart
fig3=px.scatter(data, x="S02",y="AQI")

#showing the plot
fig3.show()
```



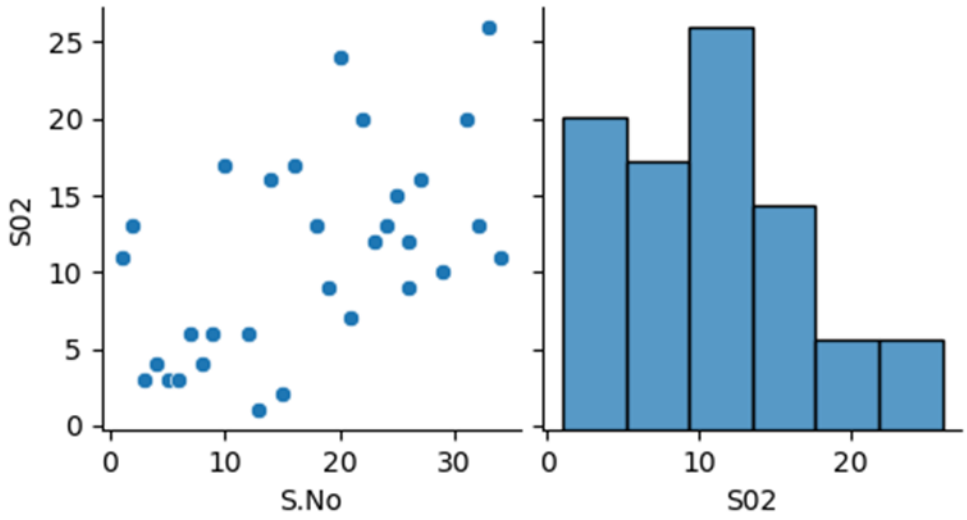
In [69]:
```
import plotly.express as px

#plotting the bubble chart
fig4=px.scatter(data, x="N02",y="AQI")

#showing the plot
fig4.show()
```

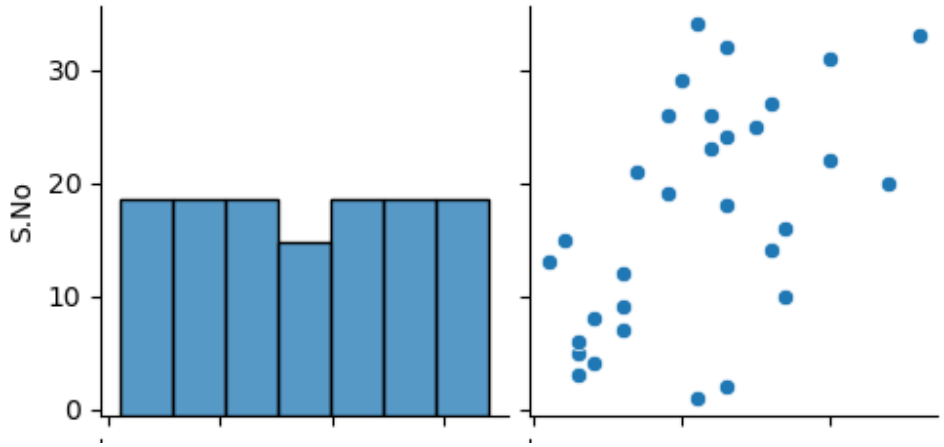


In [71]:
```
import plotly.express as px

#plotting the bubble chart
fig5=px.scatter(data, x="co",y="AQI")

#showing the plot
fig5.show()
```



In [73]:
```
sns.pairplot(data)
```





CONCLUSION:

In our project, the air quality analysis using Python data analysis tools has provided valuable insights into the pollution levels in the specified region. The data revealed patterns and trends in pollutant concentrations over time, allowing for a better understanding of the factors influencing air quality. This analysis also identified potential sources of pollution and highlighted areas that may require targeted intervention. Overall, leveraging Python for air quality analysis proves to be an effective approach for environmental monitoring and policy-making efforts to mitigate air pollution.