

AIR QUALITY ANALYSIS AND PREDICTION IN TAMILNADU

Phase 1: Problem Definition and Design Thinking

Problem Definition: 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.

Design Thinking:

1. **Project Objectives:** Define objectives such as analyzing air quality trends, identifying pollution hotspots, and building a predictive model for RSPM/PM10 levels.
2. **Analysis Approach:** Plan the steps to load, preprocess, analyze, and visualize the air quality data.
3. **Visualization Selection:** Determine visualization techniques (e.g., line charts, heatmaps) to effectively represent air quality trends and pollution levels.

1.Project Objectives: Define objectives such as analyzing air quality trends, identifying pollution hotspots, and building a predictive model for RSPM/PM10 levels.

ANALYZE AIR QUALITY TRENDS:

Objective: To thoroughly examine historical air quality data from monitoring stations in Tamil Nadu to identify long-term trends and patterns in air quality.

Key Activities:

- Collect and preprocess historical air quality data.
- Perform time-series analysis to uncover trends, seasonality, and irregularities.

- Create visualizations, such as line charts and trend plots, to represent the air quality trends effectively.

IDENTIFY POLLUTION HOTSPOT:

Objective: To pinpoint geographic areas within Tamil Nadu that consistently experience high levels of air pollution.

Key Activities:

- Utilizing spatial data analysis techniques to identify pollution hotspots.
- Creating heatmaps, spatial distribution maps, or clustering analyses to visualize pollution concentration patterns.
- Analyze contributing factors to pollution in identified hotspots.

BUILD A PREDICTIVE MODEL FOR RSPM/PM10 LEVELS:

Objective: To develop a predictive model that estimates RSPM/PM10 levels based on the levels of SO₂ and NO₂, aiding in forecasting air quality.

Key Activities:

- Collecting and preprocessing relevant data, including RSPM/PM10, SO₂, and NO₂ levels.
- Splitting the data into training and testing sets.
- Choosing an appropriate machine learning algorithm (e.g., regression).
- Training and validating the model using historical data.
- Evaluate the model's performance and accuracy using appropriate metrics (e.g., MAE, MSE, R-squared).
- Deploying the predictive model for real-time or future air quality predictions.

These objectives will serve as a guideline for our project, ensuring that we cover each aspect of air quality analysis and prediction in Tamil Nadu .

2. Analysis Approach: Plan the steps to load, preprocess, analyze, and visualize the air quality data.

Data Collection:

- Identify and collect air quality data from monitoring stations in TamilNadu.
- Storing the data in a structured format, such as a database or CSV files, for easy access and analysis.

Data Preprocessing:

- Clean the data to handle missing values, outliers, and inconsistencies. This includes techniques like data interpolation.
- Standardize units and formats for consistency across different monitoring stations.
- Performing quality checks to ensure data integrity.

Spatial Analysis:

- For identifying pollution hotspots, use spatial data analysis techniques.
- Conduct clustering analysis to group monitoring stations with similar air quality characteristics.

Predictive Modeling:

- To build a predictive model, preparing the data by selecting relevant features (SO₂, NO₂ levels) and defining target variables (RSPM/PM₁₀ levels).
- Splitting the data into training and testing sets for model development and evaluation.
- Choosing an appropriate machine learning algorithm (e.g., linear regression, random forest).
- Train and validate the predictive model, and assess its performance using appropriate metrics.

Data Visualization:

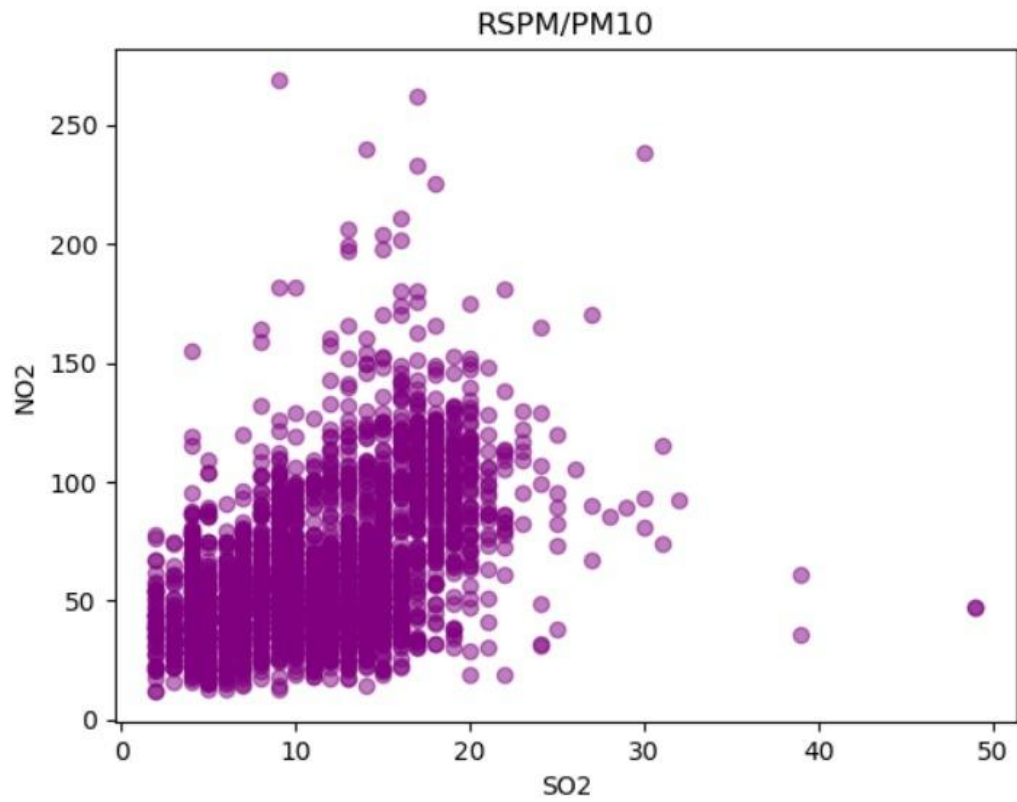
- Creating informative data visualizations to communicate your findings.
- Using geographical maps to visualize pollution hotspots.
- Generating time-series plots to showcase air quality trends over time.
- Designing interactive dashboards, if applicable, to allow users to explore the data dynamically.

Interpretation and Insights:

- Interpret the results obtained from the analysis and modeling efforts.
- Provide insights into air pollution trends, hotspot locations, and the accuracy of predictive model.
- Highlighting any actionable recommendations.

By following this structured approach, we can effectively load, preprocess, analyze, and visualize air quality data, leading to meaningful insights and actionable recommendations for managing air pollution in Tamil Nadu.

3. Visualization Selection: Determine visualization techniques (e.g., line charts, heatmaps) to effectively represent air quality trends and pollution levels.



```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

In [7]: df = pd.read_csv("cpcb_dly_aq_tamil_nadu-2014.csv")
df

Out[7]:
```

	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	01-02-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	01-07-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN
2	38	21-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN
3	38	23-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN
4	38	28-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	NaN
...
2874	773	12-03-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.0	102.0	NaN
2875	773	12-10-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.0	91.0	NaN
2876	773	17-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.0	100.0	NaN
2877	773	24-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.0	95.0	NaN
2878	773	31-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.0	94.0	NaN

2879 rows x 11 columns

```
In [8]: randomrows = df.sample(n=9)
print(randomrows)
```

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
985	180	11-11-2014	Tamil Nadu	Chennai	NEERI, CSIR Campus Chennai	National Environmental Engineering Research In...	Industrial Area	3.0	45.0	51.0	NaN
411	765	25-02-2014	Tamil Nadu	Chennai	Anna Nagar, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	18.0	21.0	85.0	NaN
1130	237	04-11-2014	Tamil Nadu	Coimbatore	SIDCO Office, Coimbatore	Tamilnadu State Pollution Control Board	Industrial Area	5.0	23.0	75.0	NaN
1209	238	02-06-2014	Tamil Nadu	Coimbatore	Collector's Office, Coimbatore	Tamilnadu State Pollution Control Board	Industrial Area	4.0	22.0	NaN	NaN
545	764	05-02-2014	Tamil Nadu	Chennai	Adyar, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	14.0	21.0	50.0	NaN
2069	763	31-10-2014	Tamil Nadu	Mettur	SIDCO Industrial Complex, Mettur	Tamilnadu State Pollution Control Board	Industrial Area	9.0	29.0	26.0	NaN
1897	762	24-02-2014	Tamil Nadu	Mettur	Raman Nagar, Mettur	Tamilnadu State Pollution Control Board	Industrial Area	7.0	19.0	22.0	NaN
2384	366	26-09-2014	Tamil Nadu	Thoothukudi	AVM Jewellery Building, Tuticorin	Tamilnadu State Pollution Control Board	Industrial Area	9.0	12.0	48.0	NaN
51	38	22-07-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	14.0	17.0	50.0	NaN

```
In [9]: print(df.tail(5))
```

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
2874	773	12-03-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.0	102.0	NaN
2875	773	12-10-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.0	91.0	NaN
2876	773	17-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.0	100.0	NaN
2877	773	24-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.0	95.0	NaN
2878	773	31-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.0	94.0	NaN

```
In [10]: nullnum = df.isnull()
print(nullnum)
```

	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	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False
...
2874	False	False	False	False	False	False	False	False	False	False	False
2875	False	False	False	False	False	False	False	False	False	False	False
2876	False	False	False	False	False	False	False	False	False	False	False
2877	False	False	False	False	False	False	False	False	False	False	False
2878	False	False	False	False	False	False	False	False	False	False	False
...
2874	False	False	False	False	False	False	False	False	False	False	False
2875	False	False	False	False	False	False	False	False	False	False	False
2876	False	False	False	False	False	False	False	False	False	False	False
2877	False	False	False	False	False	False	False	False	False	False	False
2878	False	False	False	False	False	False	False	False	False	False	False

[2879 rows x 11 columns]

```
In [11]: total_null_values = df.isnull().sum().sum()
print("Total number of null values:", total_null_values)
```

Total number of null values: 2907

```
In [12]: print("DataFrame Info:")
print(df.info())
print("\nDescriptive Statistics:")
print(df.describe())
```

DataFrame Info:

```
<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
```

Descriptive Statistics:

	Stn Code	SO2	NO2	RSPM/PM10	PM 2.5
count	2879.000000	2868.000000	2866.000000	2875.000000	0.0
mean	475.750261	11.503138	22.136776	62.494261	NaN
std	277.675577	5.051702	7.128694	31.368745	NaN
min	38.000000	2.000000	5.000000	12.000000	NaN
25%	238.000000	8.000000	17.000000	41.000000	NaN
50%	366.000000	12.000000	22.000000	55.000000	NaN
75%	764.000000	15.000000	25.000000	78.000000	NaN
max	773.000000	49.000000	71.000000	269.000000	NaN

```
In [13]: print("Data Types of Columns:")
print(df.dtypes)
```

Data Types of Columns:

Stn Code	int64
Sampling Date	object
State	object
City/Town/Village/Area	object
Location of Monitoring Station	object
Agency	object
Type of Location	object
SO2	float64
NO2	float64
RSPM/PM10	float64
PM 2.5	float64
dtype:	object

```
In [14]: duplicate_rows = df[df.duplicated()]
print("Duplicate Rows:")
print(duplicate_rows)
```

Duplicate Rows:

Empty DataFrame

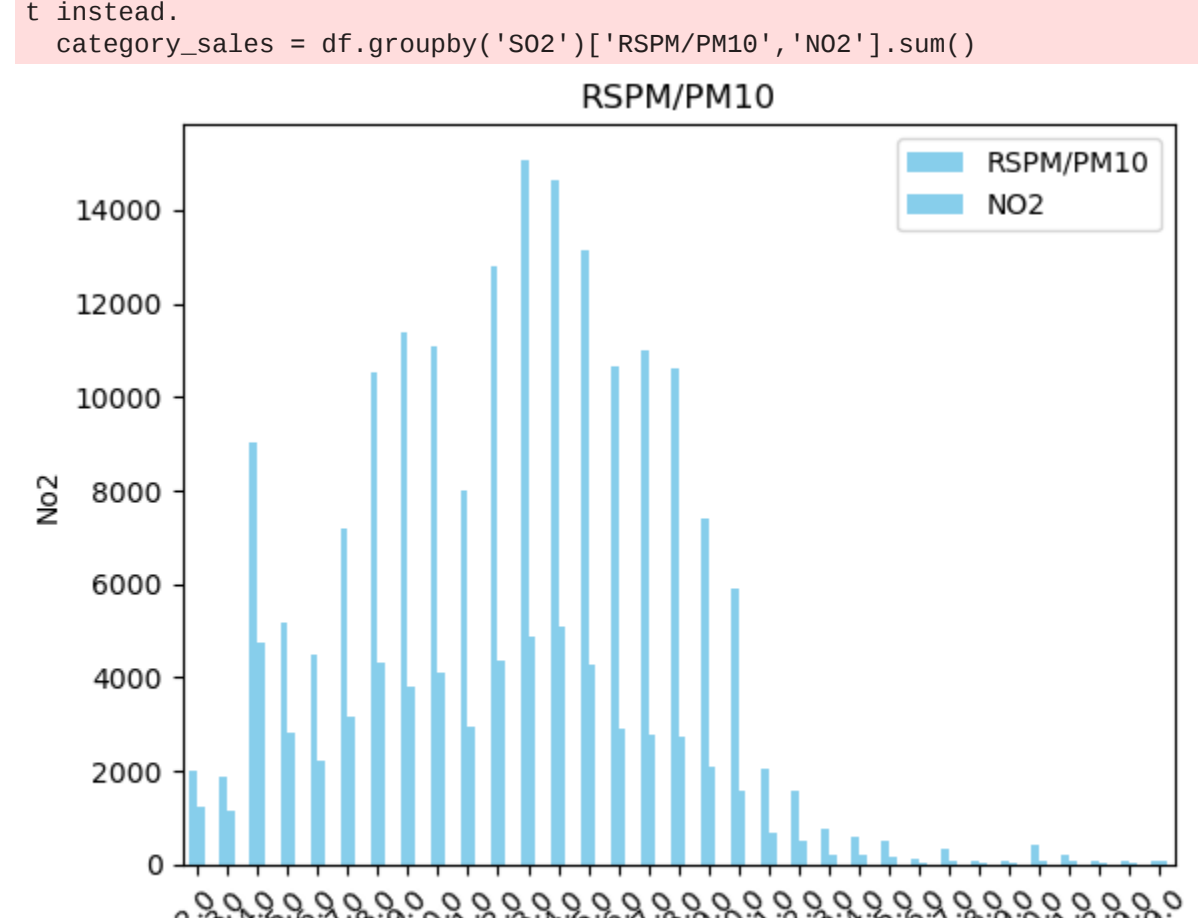
Columns: [Stn Code, Sampling Date, State, City/Town/Village/Area, Location of Monitoring Station, Agency, Type of Location, SO2, NO2, RSPM/PM10, PM 2.5]

Index: []

```
In [15]: # Bar Chart
category_sales = df.groupby('SO2')[['RSPM/PM10', 'NO2']].sum()
category_sales.plot(kind='bar', color='skyblue')
plt.title('RSPM/PM10')
plt.xlabel('SO2')
plt.ylabel('NO2')
plt.xticks(rotation=45)
plt.show()
```

C:\Users\savio\AppData\Local\Temp\ipykernel_12524\654104536.py:2: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.

category_sales = df.groupby('SO2')[['RSPM/PM10', 'NO2']].sum()



```
In [20]: #Line Chart
df_time = df.groupby('SO2')[['RSPM/PM10', 'NO2']].sum()

df_time['SO2'].plot(label='SO2', color='blue')
df_time['NO2'].plot(label='NO2', color='green')

plt.xlabel('SO2')
plt.ylabel('NO2')
plt.legend()
plt.show()
```

```
-----
KeyError                                Traceback (most recent call last)
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3802, in Index.get_loc(self, key, method, tolerance)
3801 try:
-> 3802     return self._engine.get_loc(casted_key)
3803 except KeyError as err:
File ~\anaconda3\Lib\site-packages\pandas\_libs\index.py:138, in pandas._libs.index.IndexEngine.get_loc()
File ~\anaconda3\Lib\site-packages\pandas\_libs\index.py:165, in pandas._libs.index.IndexEngine.get_loc()
File pandas._libs\hashtable_class_helper.pxi:5745, in pandas._libs.hashtable.PyObjectHashTable.get_item()
File pandas._libs\hashtable_class_helper.pxi:5753, in pandas._libs.hashtable.PyObjectHashTable.get_item()
KeyError: 'SO2'

The above exception was the direct cause of the following exception:

KeyError                                Traceback (most recent call last)
Cell In[20], line 5
      1 #Line Chart
      3 df_time = df.groupby('SO2')[['RSPM/PM10', 'NO2']].sum()
----> 5 df_time['NO2'].plot(label='NO2', color='green')
      6 df_time['NO2'].plot(label='NO2', color='green')
      8 plt.xlabel('SO2')
```

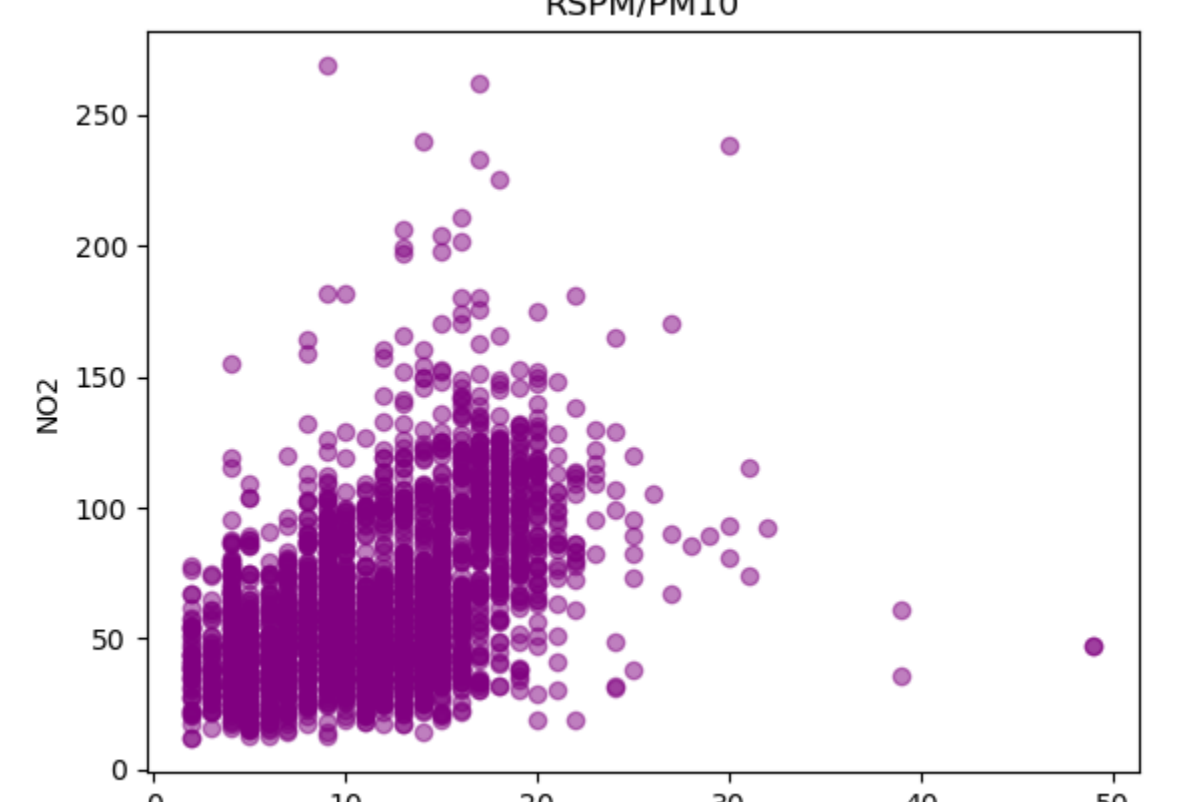
```
File ~\anaconda3\Lib\site-packages\pandas\core\frame.py:3807, in DataFrame._getitem_(self, key)
3805 if self.columns.nlevels > 1:
3806     return self._getitem_multilevel(key)
-> 3807 indexer = self.columns.get_loc(key)
3808 if is_integer(indexer):
3809     indexer = [indexer]
```

```
File ~\anaconda3\Lib\site-packages\pandas\core\indexes\base.py:3804, in Index.get_loc(self, key, method, tolerance)
3802     return self._engine.get_loc(casted_key)
3803 except KeyError as err:
-> 3804     raise KeyError(key) from err
3805 except TypeError:
3806     # If we have a listlike key, _check_indexing_error will raise
3807     # InvalidIndexError. Otherwise we fall through and re-raise ...
3808     # the TypeError.
3809     self._check_indexing_error(key)

KeyError: 'SO2'
```

```
In [22]: # Scatter Plot

plt.scatter(df['SO2'], df['RSPM/PM10'], alpha=0.5, color='purple')
plt.title('RSPM/PM10')
plt.xlabel('SO2')
plt.ylabel('NO2')
plt.show()
```



```
In [ ]:
```