**NAME : M.BHARATHI**

**COLLEGE CODE : 4101**

**REG NO : 410121104010**

**DATA ANALYTICS WITH COGNOS**

**PROJECT TITLE : Air Quality Analysis**

**PHASE 5: Project Documentation and Submission.**

**AIM :**

To document the Air Quality Analysis project and prepare it for submission.



**DOCUMENTATION :**

**STEP 1:**

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

☆ Our goal is to analyze the quality of the air from the given data set . Collecting or downloading the data from the given dataset.

***Dataset***

***Link:*** [***https://tn.data.gov.in/resource/location-wise- daily-ambient-air-quality-tamil-nadu-year-2014***](https://tn.data.gov.in/resource/location-wise-%20daily-ambient-air-quality-tamil-nadu-year-2014)

● Analyze the data .

● And then from the given data set our aim is to find the most polluted city and to build a predictive model.

**Machine Learning :**

As we need to make predictions , consider building machine learning models.

Split the data into training and testing sets, train models, and evaluate their performance.

**STEP 2:**

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

**Data Collection:**

Downloading the dataset in the CSV format using the above mentioned link.

**Data Loading:**

Using libraries like Pandas in Python to load the data into a structured format like a DataFrame.

csv\_file\_path = 'weather.csv'

df = pd.read\_csv(csv\_file\_path)

**Data Preprocessing:**

*Handle missing data*: Decide on a strategy (e.g., imputation or removal) for dealing with missing values.

*Data cleaning*: Correct any inconsistencies or errors in the data.

# df = df.dropna()

# df = df.drop\_duplicates()

**Data visualization:** Create plots and charts (histograms, scatter plots, box plots) to explore data distributions and relationships between variables.

Use libraries like Matplotlib, Seaborn, or Plotly in Python for visualization.

**STEP 3:**

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

Different visualization methods can highlight various aspects of the data, so it's important to choose the right ones based on your specific goals and the characteristics of your data. Here are some visualization techniques you can consider:

***Line Charts***: Line charts are excellent for displaying time-series data, such as air quality trends over a specific period.

***Bar Charts***: Bar charts can be used to compare air quality measurements across different locations or categories. For example, you can create a bar chart to compare pollution levels in different cities or display pollutant concentrations for various pollutants side by side.

***Heatmaps***: Heatmaps are useful for visualizing spatial data.

***Pie Charts***: Each slice of the pie can represent a different pollutant, and the size of the slice corresponds to its contribution to the total pollution.

Hence here I prefer using bar charts to visualize the pollution rate in different cities based on RSPM and PM10 levels.

**STEP 4:**

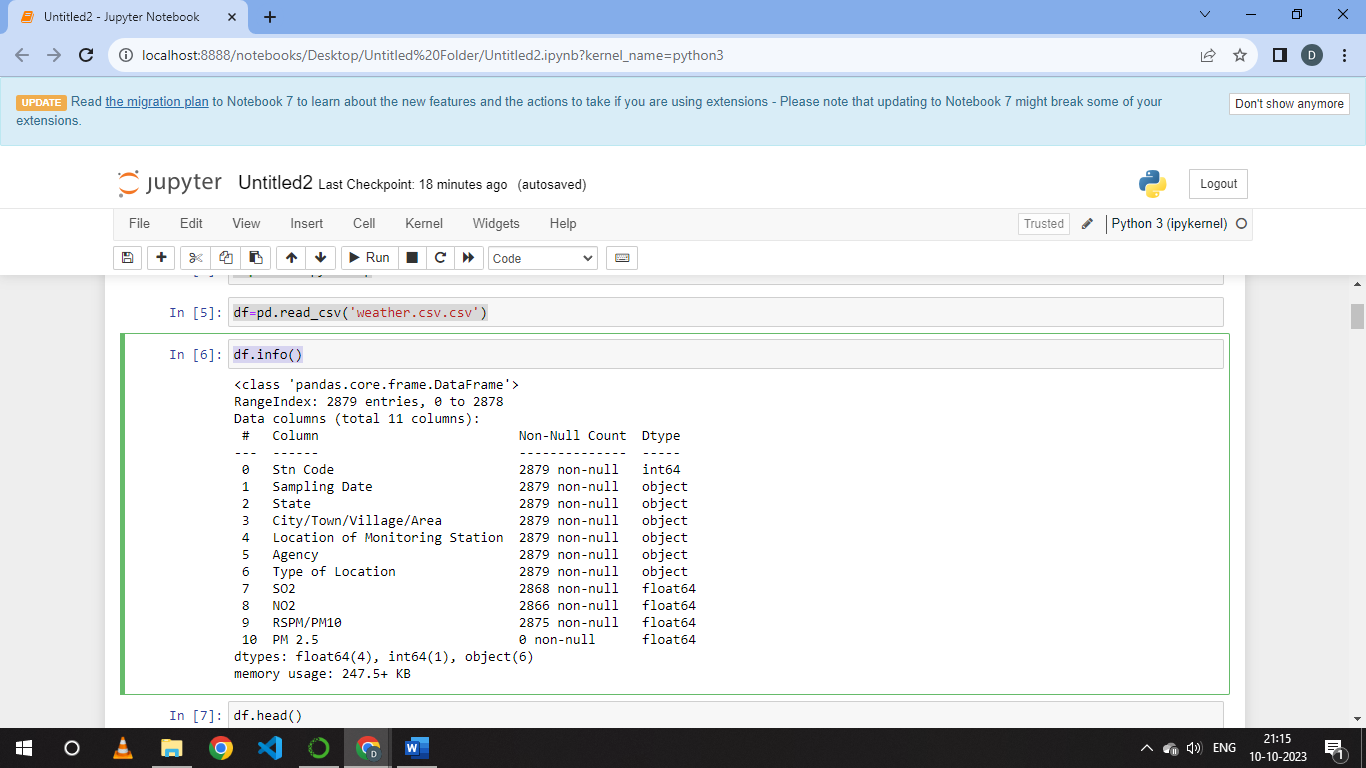
**THE CODE FOR VISUALIZATION :**

import pandas as pd

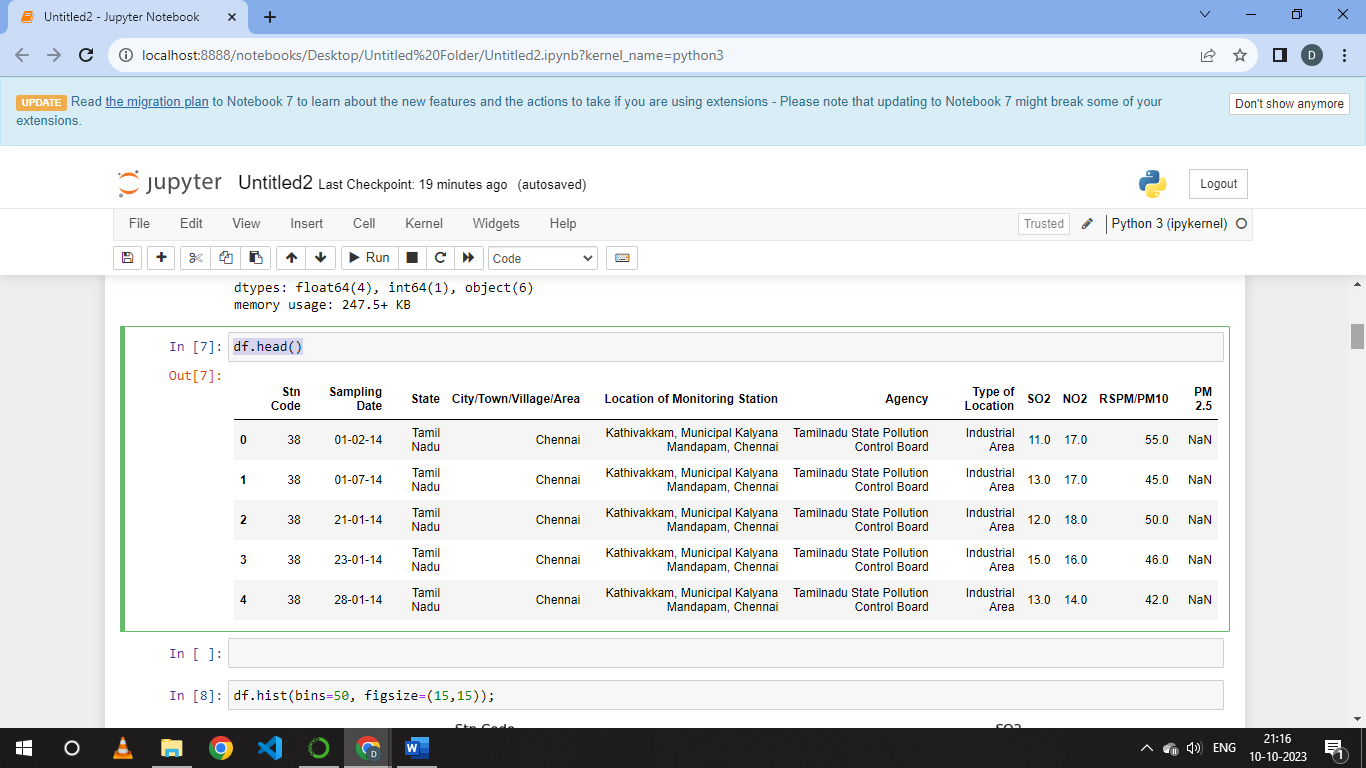
import numpy as np

df=pd.read\_csv('weather.csv')

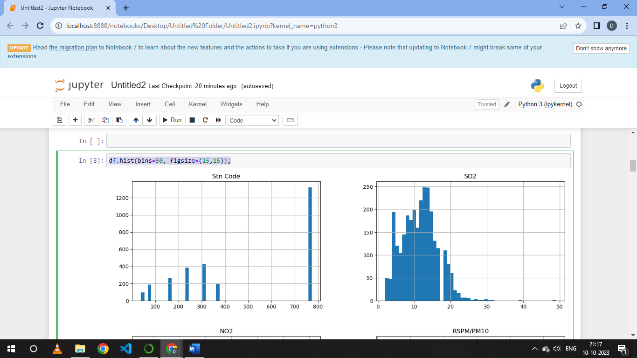
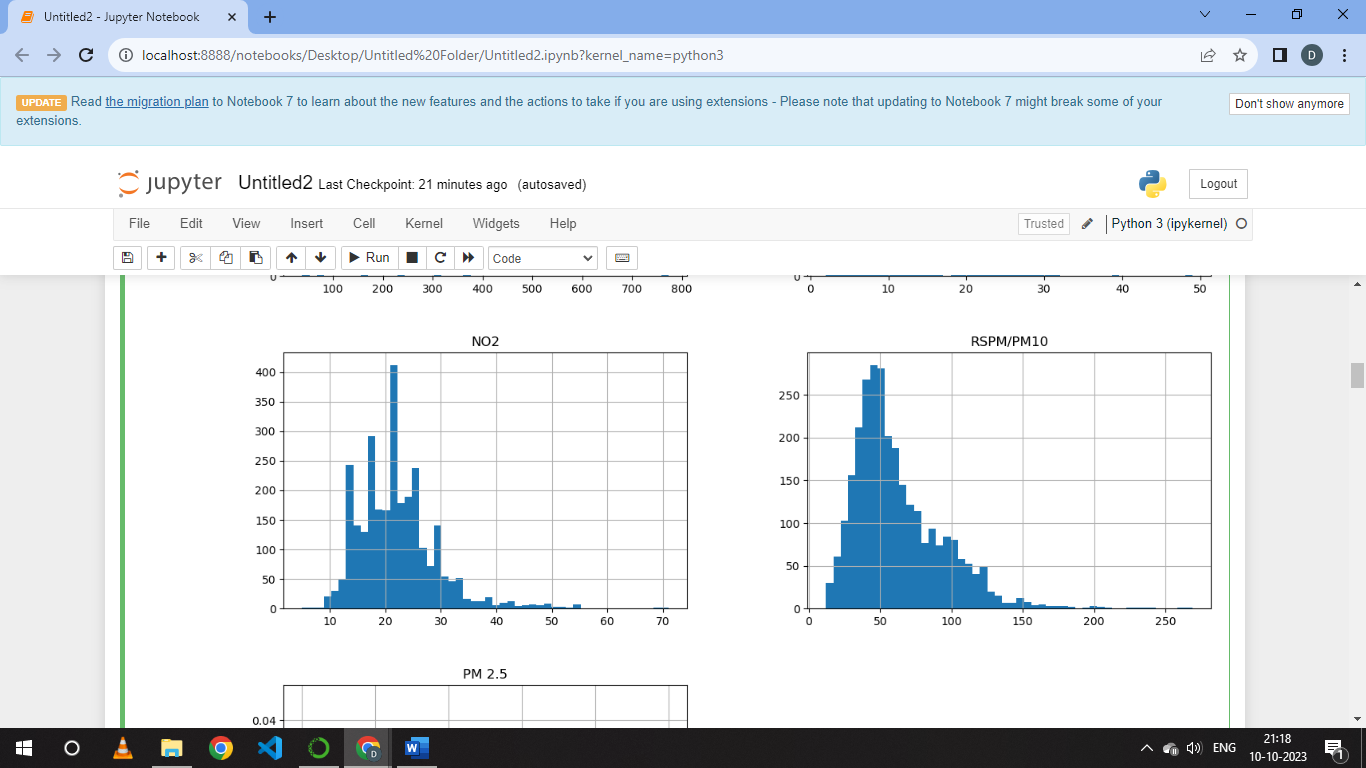
df.info()



df.head()



df.hist(bins=50, figsize=(15,15))

from sklearn.compose import ColumnTransformer

from sklearn.pipeline import Pipeline

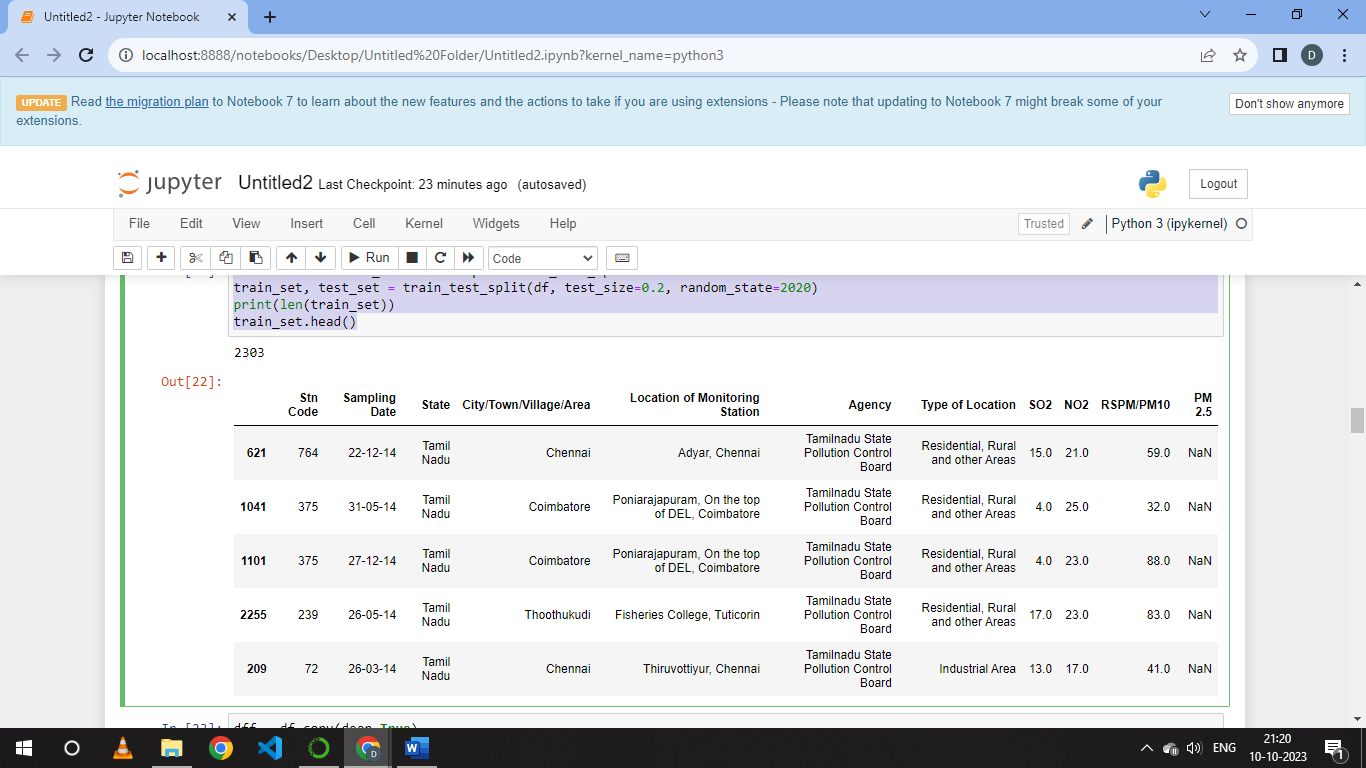
from sklearn.preprocessing import StandardScaler

from sklearn.model\_selection import train\_test\_split

train\_set, test\_set = train\_test\_split(df, test\_size=0.2, random\_state=2020)

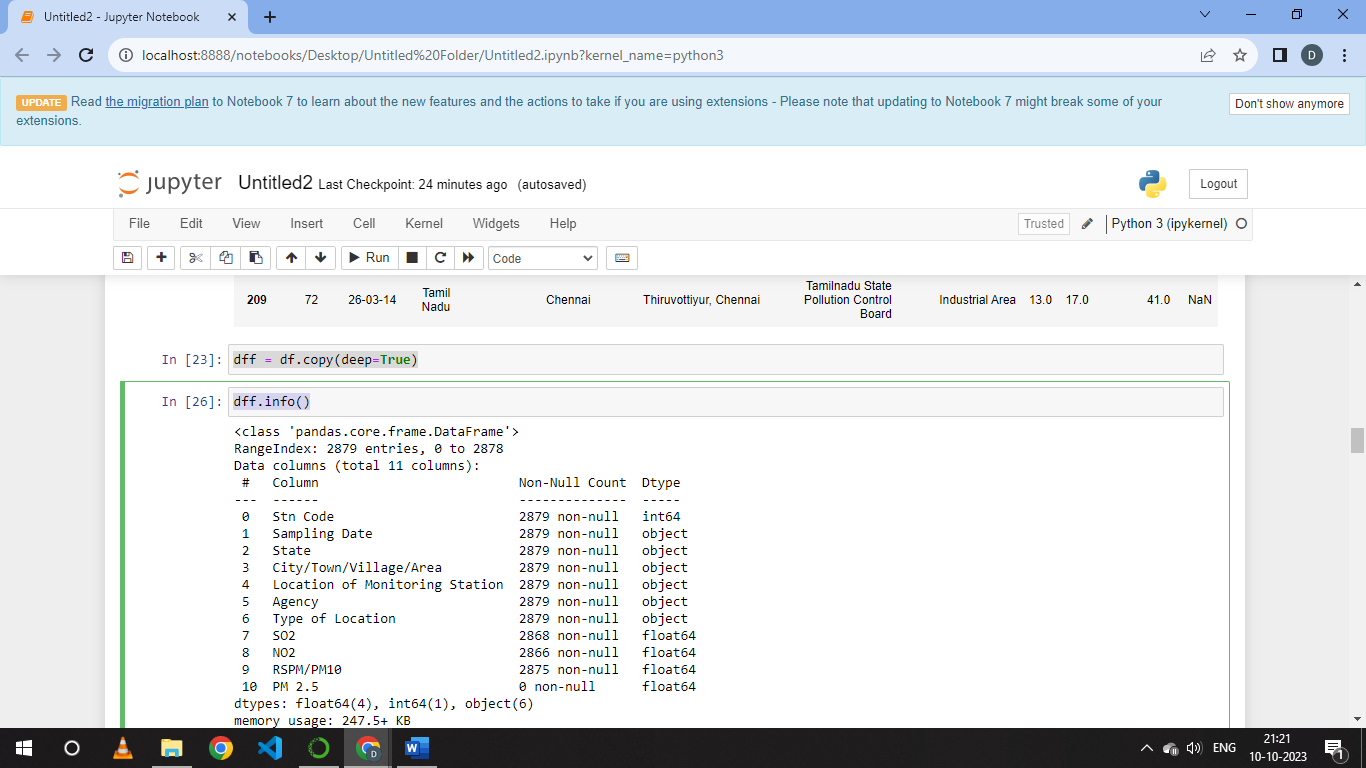
print(len(train\_set))

train\_set.head()



dff = df.copy(deep=True)

dff.info()

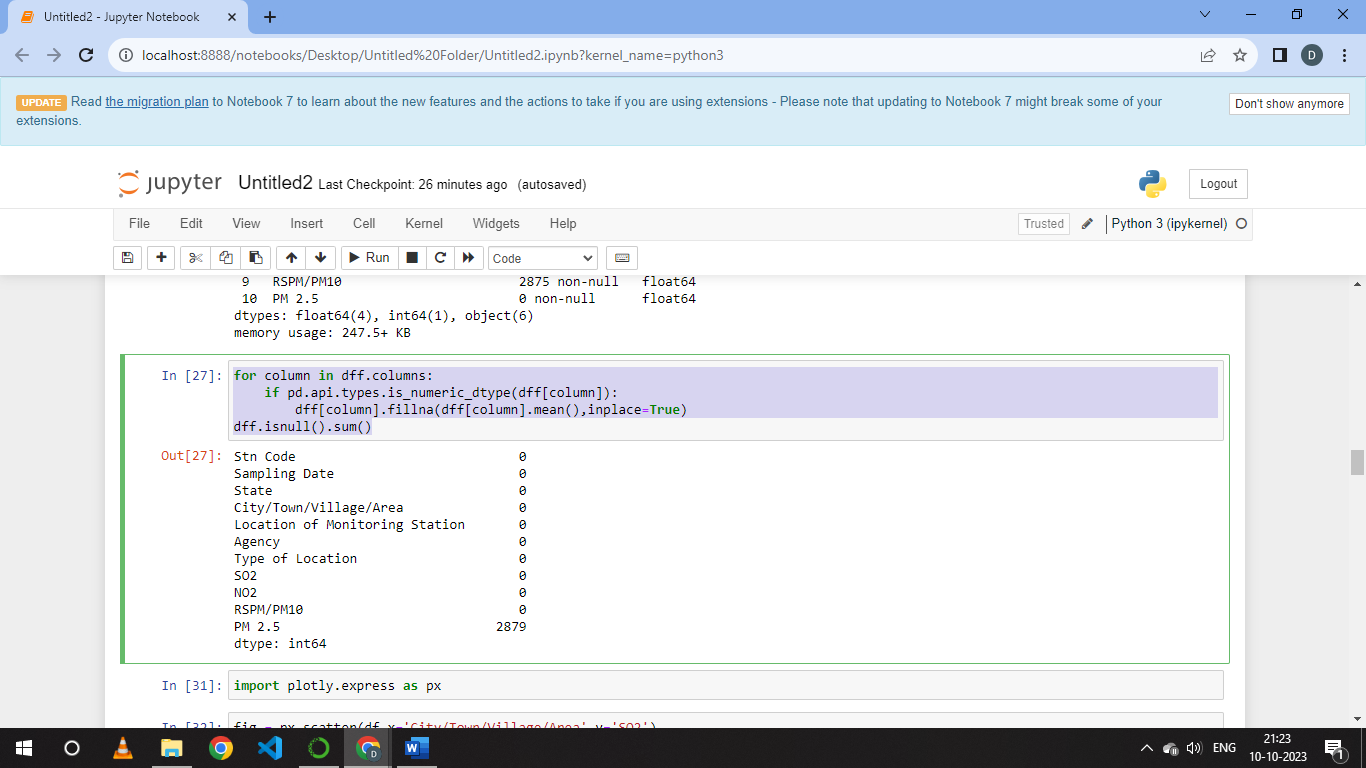


for column in dff.columns:

if pd.api.types.is\_numeric\_dtype(dff[column]):

dff[column].fillna(dff[column].mean(),inplace=True)

dff.isnull().sum()



import plotly.express as px

fig = px.scatter(df,x='City/Town/Village/Area',y='SO2')

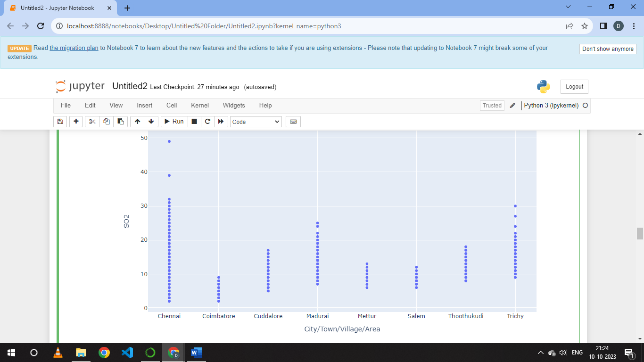
fig1 = px.scatter(df,x='City/Town/Village/Area',y='NO2')

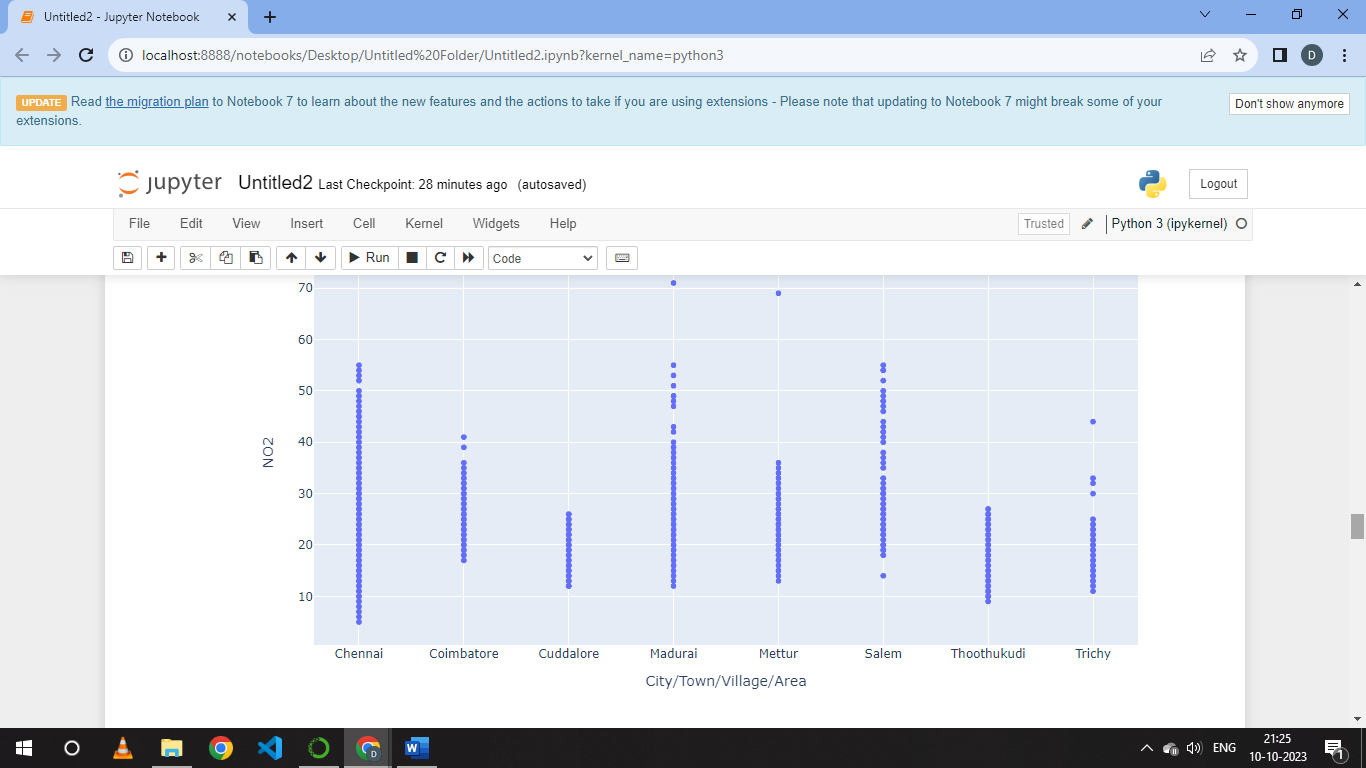
fig2 = px.scatter(df,x='City/Town/Village/Area',y='RSPM/PM10')

fig.show()

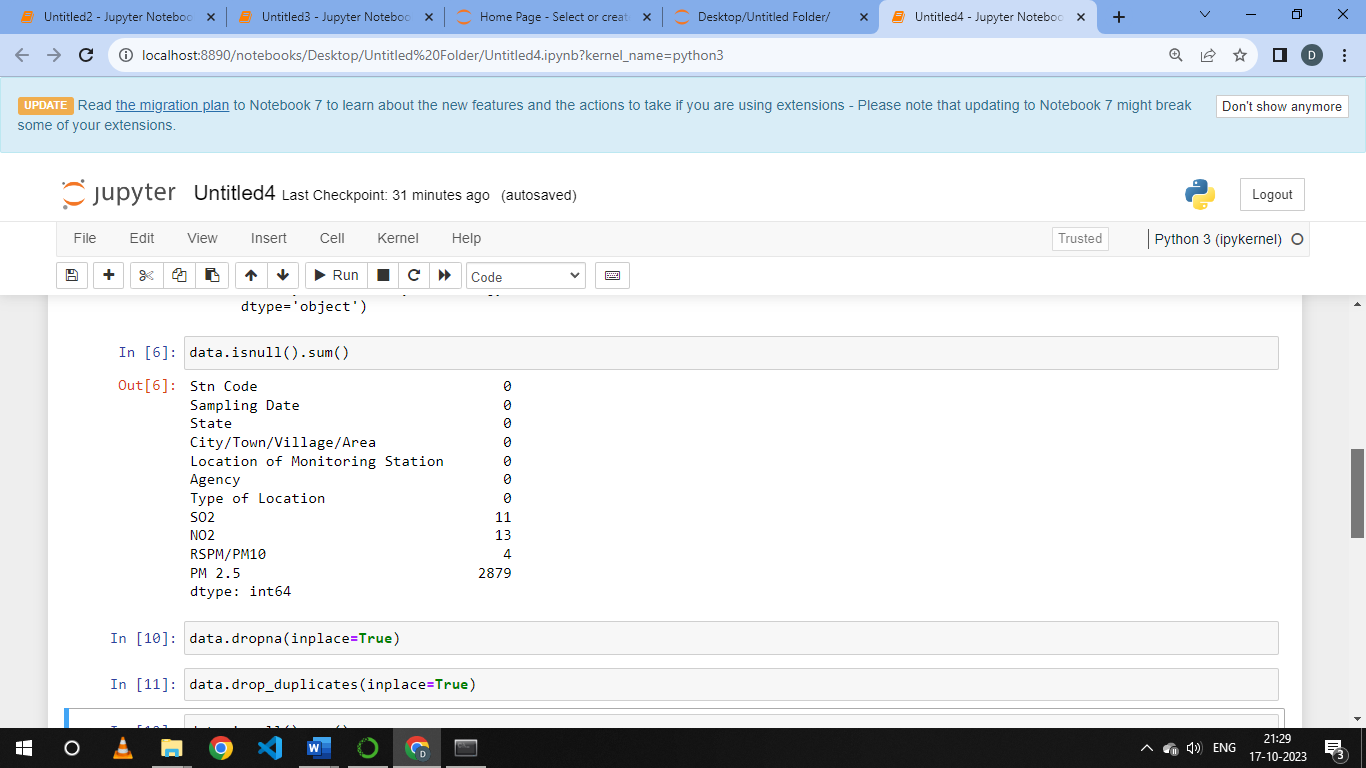
fig1.show()

fig2.show()

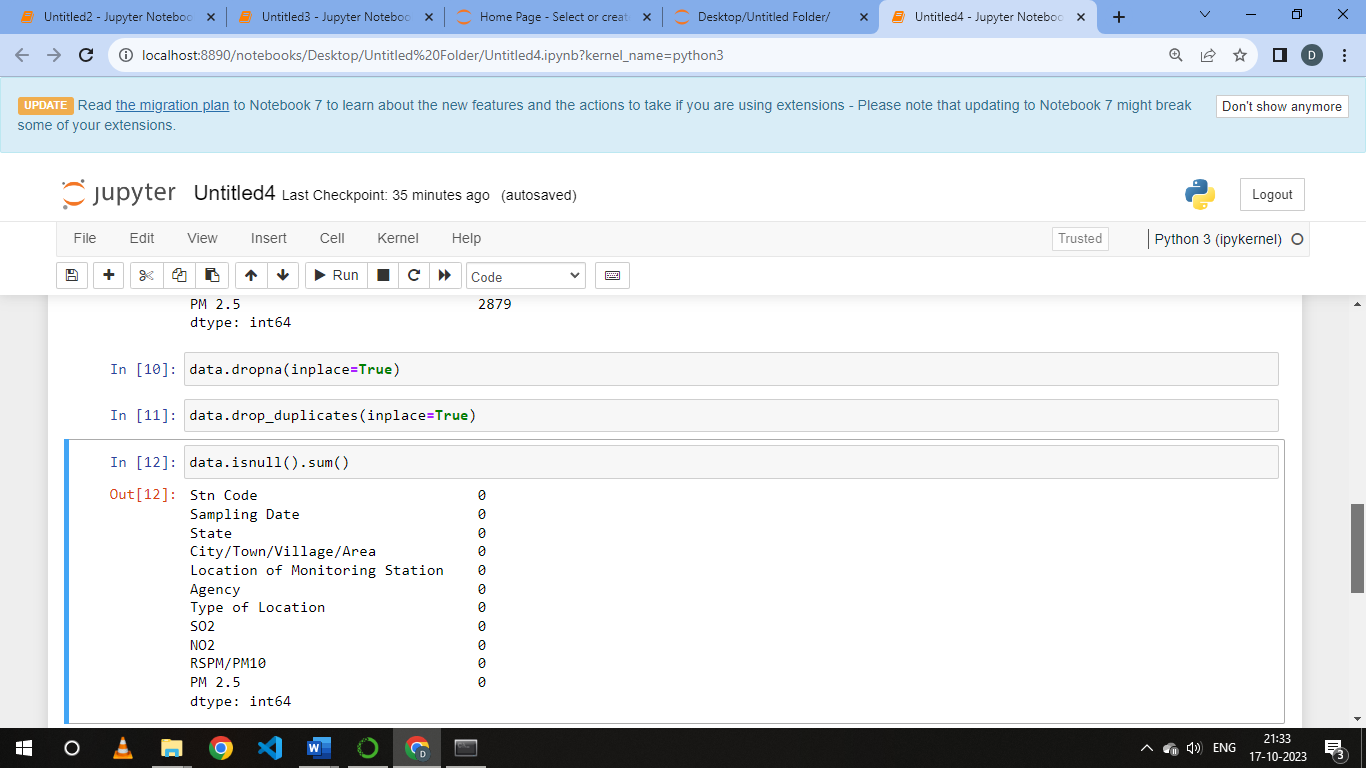




Data.dropna(inplace=True)



Data.drop\_duplicates(inplace=True)



def cal\_NOi(NO2):

ni=0

if (NO2<=40):

ni= NO2\*(50/40)

elif (NO2>40 and NO2<=80):

ni= 50+(NO2-40)\*(50/40)

elif (NO2>80 and NO2<=380):

ni= 100+(NO2-80)\*(100/300)

elif (NO2>380 and NO2<=800):

ni= 200+(NO2-380)\*(100/420)

elif (NO2>800 and NO2<=1600):

ni= 300+(NO2-800)\*(100/800)

elif (NO2>1600):

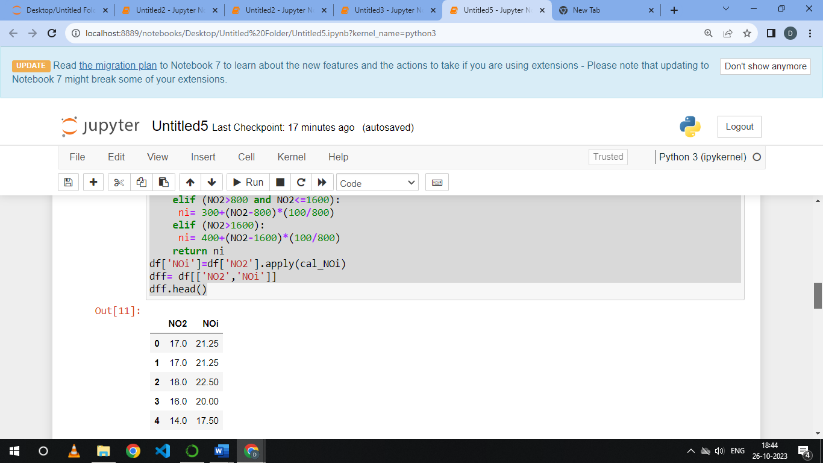
ni= 400+(NO2-1600)\*(100/800)

return ni

df['NOi']=df['NO2'].apply(cal\_NOi)

dff= df[['NO2','NOi']]

dff.head()



**STEP 5:**

To analyze air pollution trends and pollution levels in Tamil Nadu follow the given below steps:

1. Data Collection
2. Data Visualization
3. Statistical Analysis
4. Identifying Sources
5. Comparisons
6. Public Health Impact.

By following these steps, it can provide valuable insights into air pollution trends and pollution levels in Tamil Nadu, helping to inform decision making and efforts to improve air quality in the region.

**CONCLUSION :**

Overall, this analysis underscores the importance of immediate action to address air pollution in Tamil Nadu. It provides a foundation for evidence-based policy making and initiatives aimed at improving air quality, protecting public health and creating a more sustainable environment for the region’s residents.