STATISTICS PROJECT

EXPLORATORY DATA ANALYSIS

ON

AIR QUALITY INDEX OF INDIA

SUBMITTED TO

Praxis Business School

C23017	Manas Ranjan Pal
C23021	Prem Sambhaji Jadhav
C23030	Shubham Nath

INTRODUCTION

The air quality index (AQI) is a daily index for reporting air quality. Government agencies use an air quality index (AQI) to communicate to the public how polluted the air is now or how polluted it is expected to become. The AQI is calculated by averaging readings from an air quality sensor, which can rise due to vehicle traffic, forest fires, or any other source of air pollution. A low AQI indicates good air quality and low levels of pollution, whereas a high AQI indicates increased pollutant concentrations in the air, which is extremely harmful to human health.

As the AQI rises, so do the risks to public health, particularly for children, the elderly, and people with respiratory or cardiovascular issues. During these times, governments generally encourage people to limit their outdoor physical activity or even avoid going out altogether. Face masks, such as cloth masks, may also be advised.

The air quality index is made up of twelve pollutants (Benzene, Toluene, Xylene, PM10, PM2.5, NO2, NOx, NO, SO2, CO, O3, NH3) AQI levels and classifications: Good (0–50), Satisfactory (51–100), Polluted moderately (101–200), Poor (201–300), Very bad (301–400) Severe (401–500).

In this EDA, we will use the day-by-day AQI dataset, which contains data on the daily level of pollutants and AQI in approximately 12 Indian megacities from 2015 to 2020.

Identifying and analysing the most polluted cities in recent years. Understanding the impact of COVID-19-induced lockdowns on major city air quality: determining which cities saw the greatest improvement in air quality and which saw a spike in AQI levels despite a strict lockdown.

The data has been made publicly available by the Central Pollution Control Board: https://cpcb.nic.in/ which is the official portal of Government of India.

DATA ANALYSIS ON PYTHON

Loading the data into python notebook for checking null values and outliers.

In [1]:
```
import numpy as np
import pandas as pd

import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

In [2]:
```
data = pd.read_csv("India AQI.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI	AQI_Bucket
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00	NaN	NaN
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77	NaN	NaN
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25	NaN	NaN
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00	NaN	NaN
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78	NaN	NaN

In [4]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29531 entries, 0 to 29530
Data columns (total 16 columns):
 #   Column      Non-Null Count  Dtype  
---  ------      --------------  -----  
 0   City        29531 non-null  object 
 1   Date        29531 non-null  object 
 2   PM2.5       24933 non-null  float64
 3   PM10        18391 non-null  float64
 4   NO          25949 non-null  float64
 5   NO2         25946 non-null  float64
 6   NOx         25346 non-null  float64
 7   NH3         19203 non-null  float64
 8   CO          27472 non-null  float64
 9   SO2         25677 non-null  float64
 10  O3          25509 non-null  float64
 11  Benzene     23908 non-null  float64
 12  Toluene     21490 non-null  float64
 13  Xylene      11422 non-null  float64
 14  AQI         24850 non-null  float64
 15  AQI_Bucket  24850 non-null  object 
dtypes: float64(13), object(3)
memory usage: 3.6+ MB
```

Treating Outliers

```
In [5]: 1 data.describe()
```

Out[5]:

	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Tol
count	24933.000000	18391.000000	25949.000000	25946.000000	25346.000000	19203.000000	27472.000000	25677.000000	25509.000000	23908.000000	21490.00
mean	67.450578	118.127103	17.574730	28.560659	32.309123	23.483476	2.248598	14.531977	34.491430	3.280840	8.70
std	64.661449	90.605110	22.785846	24.474746	31.646011	25.684275	6.962884	18.133775	21.694928	15.811136	19.96
min	0.040000	0.010000	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.010000	0.000000	0.00
25%	28.820000	56.255000	5.630000	11.750000	12.820000	8.580000	0.510000	5.670000	18.860000	0.120000	0.60
50%	48.570000	95.680000	9.890000	21.690000	23.520000	15.850000	0.890000	9.160000	30.840000	1.070000	2.97
75%	80.590000	149.745000	19.950000	37.620000	40.127500	30.020000	1.450000	15.220000	45.570000	3.080000	9.15
max	949.990000	1000.000000	390.680000	362.210000	467.630000	352.890000	175.810000	193.860000	257.730000	455.030000	454.85

```
In [5]: 1 data.describe()
```

Out[5]:

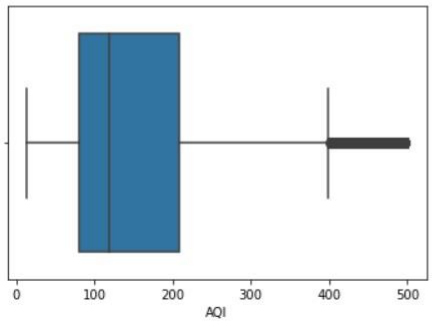
10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene	AQI
00	25949.000000	25946.000000	25346.000000	19203.000000	27472.000000	25677.000000	25509.000000	23908.000000	21490.000000	11422.000000	24850.000000
03	17.574730	28.560659	32.309123	23.483476	2.248598	14.531977	34.491430	3.280840	8.700972	3.070128	166.463581
10	22.785846	24.474746	31.646011	25.684275	6.962884	18.133775	21.694928	15.811136	19.969164	6.323247	140.696585
00	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.010000	0.000000	0.000000	0.000000	13.000000
00	5.630000	11.750000	12.820000	8.580000	0.510000	5.670000	18.860000	0.120000	0.600000	0.140000	81.000000
00	9.890000	21.690000	23.520000	15.850000	0.890000	9.160000	30.840000	1.070000	2.970000	0.980000	118.000000
00	19.950000	37.620000	40.127500	30.020000	1.450000	15.220000	45.570000	3.080000	9.150000	3.350000	208.000000
00	390.680000	362.210000	467.630000	352.890000	175.810000	193.860000	257.730000	455.030000	454.850000	170.370000	2049.000000

Here we can see the columns PM2.5, PM10 and AQI have outliers. The max values of these 3 show the data has values greater than 500 which is possibly data entry error from sensors.

We throttled the outliers to the max value possible for the pollutants, which is 500.

```
In [10]: 1 # To replace any value greater than 500 with 500
         2 data['AQI'] = np.where(data['AQI'] > 500, 500, data['AQI'])
         3 # To plot the boxplot after removing outliers
         4 sns.boxplot(data['AQI']);
```

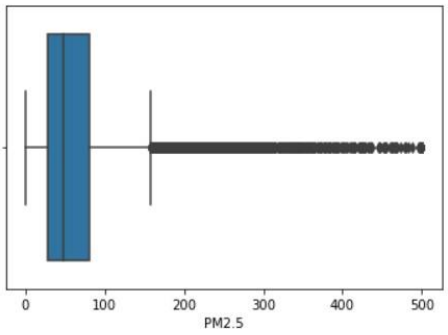
```
C:\Users\majim\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



In [11]:

```
# To replace any value greater than 500 with 500
data['PM2.5'] = np.where(data['PM2.5'] > 500, 500, data['PM2.5'])
# To plot the boxplot after removing outliers
sns.boxplot(data['PM2.5']);
```

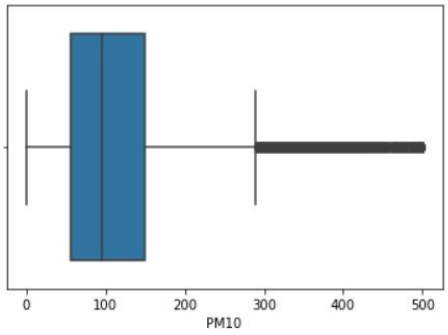
```
C:\Users\majim\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



In [12]:

```
# To replace any value greater than 500 with 500
data['PM10'] = np.where(data['PM10'] > 500, 500, data['PM10'])
# To plot the boxplot after removing outliers
sns.boxplot(data['PM10']);
```

```
C:\Users\majim\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword a
rg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit ke
yword will result in an error or misinterpretation.
  warnings.warn(
```



Treating missing values

In [13]:
```
data.isnull().sum()
```

Out[13]:
```
City             0
Date             0
PM2.5         4598
PM10         11140
NO            3582
NO2           3585
NOx           4185
NH3          10328
CO            2059
SO2           3854
O3            4022
Benzene       5623
Toluene       8041
Xylene       18109
AQI           4681
AQI_Bucket    4681
dtype: int64
```

Using median for imputing missing values of numerical variables and mode for categorical variables.

In [14]:
```
data['Date'] = pd.to_datetime(data['Date'])
```

In [15]:
```
df1 = data.copy()
df1['PM2.5']=df1['PM2.5'].fillna((df1['PM2.5'].median()))
df1['PM10']=df1['PM10'].fillna((df1['PM10'].median()))
df1['NO']=df1['NO'].fillna((df1['NO'].median()))
df1['NO2']=df1['NO2'].fillna((df1['NO2'].median()))
df1['NOx']=df1['NOx'].fillna((df1['NOx'].median()))
df1['NH3']=df1['NH3'].fillna((df1['NH3'].median()))
df1['CO']=df1['CO'].fillna((df1['CO'].median()))
df1['SO2']=df1['SO2'].fillna((df1['SO2'].median()))
df1['O3']=df1['O3'].fillna((df1['O3'].median()))
df1['Benzene']=df1['Benzene'].fillna((df1['Benzene'].median()))
df1['Toluene']=df1['Toluene'].fillna((df1['Toluene'].median()))
df1['Xylene']=df1['Xylene'].fillna((df1['Xylene'].median()))
df1['AQI']=df1['AQI'].fillna((df1['AQI'].median()))
df1['AQI_Bucket']=df1['AQI_Bucket'].fillna('Moderate')
```

In [16]:
```
df1.isnull().sum()
```

Out[16]:
```
City          0
Date          0
PM2.5         0
PM10          0
NO            0
NO2           0
NOx           0
NH3           0
CO            0
SO2           0
O3            0
Benzene       0
Toluene       0
Xylene        0
AQI           0
AQI_Bucket    0
dtype: int64
```

Looking at the categorical variables.

```
In [15]: df1['City'].nunique()
Out[15]: 26

In [16]: df1['City'].unique()
Out[16]: array(['Ahmedabad', 'Aizawl', 'Amaravati', 'Amritsar', 'Bengaluru',
                'Bhopal', 'Brajrajnagar', 'Chandigarh', 'Chennai', 'Coimbatore',
                'Delhi', 'Ernakulam', 'Gurugram', 'Guwahati', 'Hyderabad',
                'Jaipur', 'Jorapokhar', 'Kochi', 'Kolkata', 'Lucknow', 'Mumbai',
                'Patna', 'Shillong', 'Talcher', 'Thiruvananthapuram',
                'Visakhapatnam'], dtype=object)

In [17]: df1['AQI_Bucket'].nunique()
Out[17]: 6

In [18]: df1['AQI_Bucket'].unique()
Out[18]: array(['Moderate', 'Poor', 'Very Poor', 'Severe', 'Satisfactory', 'Good'],
               dtype=object)
```

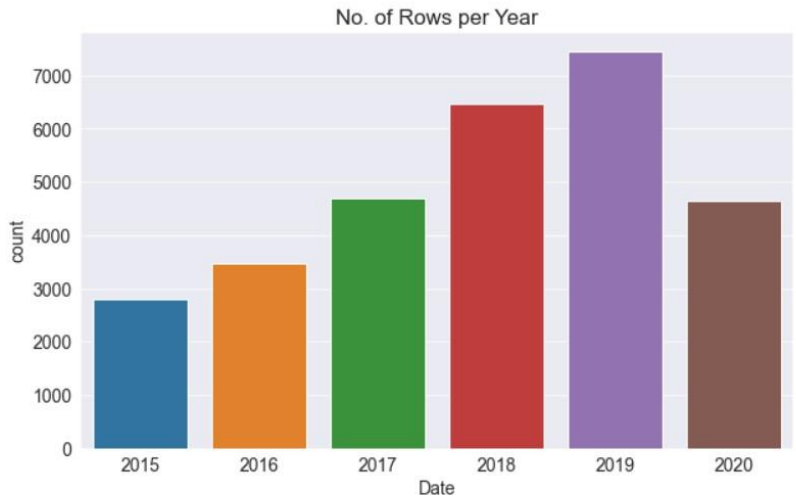
Checking range of dates for which we have the data.

```
In [19]: df1.Date.min()
Out[19]: Timestamp('2015-01-01 00:00:00')

In [20]: df1.Date.max()
Out[20]: Timestamp('2020-07-01 00:00:00')
```

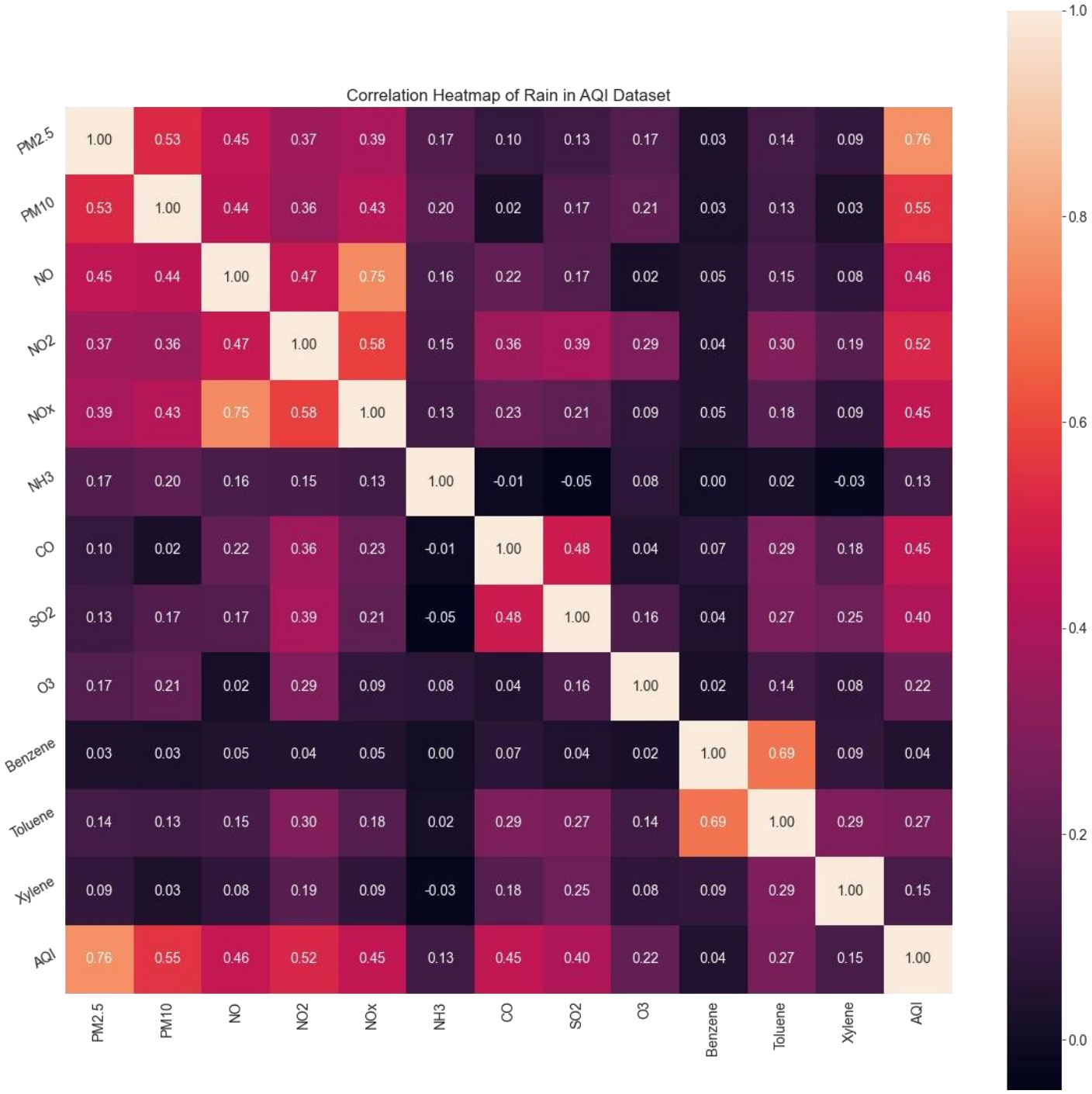
Checking number of rows in the dataset for each year.

```
In [24]: sns.set_style('darkgrid')
         matplotlib.rcParams['font.size'] = 14
         matplotlib.rcParams['figure.figsize'] = (10, 6)
         matplotlib.rcParams['figure.facecolor'] = '#00000000'
         plt.title('No. of Rows per Year')
         sns.countplot(x=df1.Date.dt.year);
```



Correlation heatmap of numerical variables:

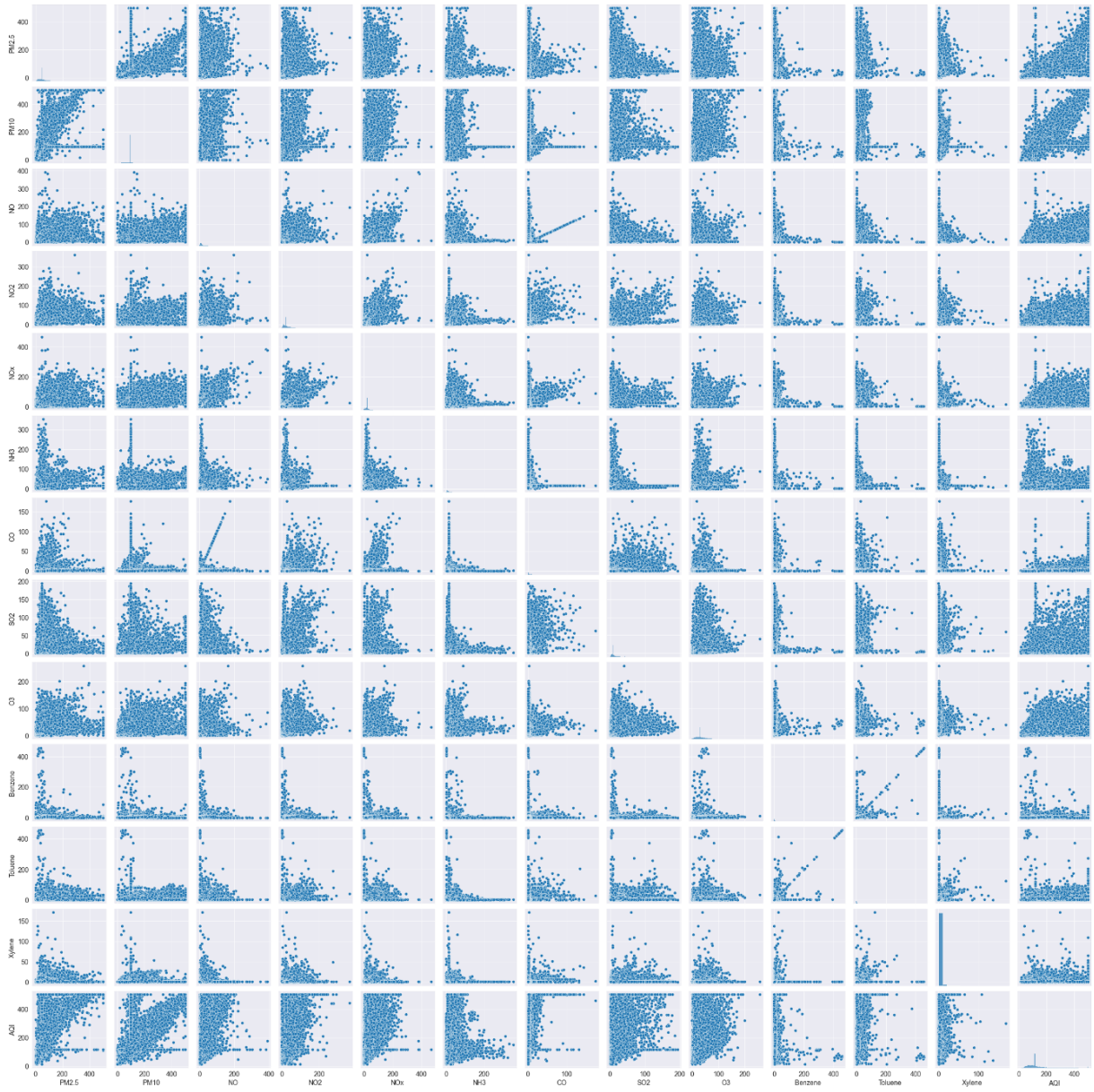
In [28]:
```
correlation = df1.corr()
plt.figure(figsize=(20,20))
plt.title('Correlation Heatmap of Rain in AQI Dataset')
ax = sns.heatmap(correlation, square=True, annot=True, fmt='.2f', linecolor='white')
ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
ax.set_yticklabels(ax.get_yticklabels(), rotation=30)
plt.show()
```



Scatter plot for each numeric variable with other numeric variables:

In [24]:
```
num = ['PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene', 'Toluene', 'Xylene', 'AQI']
```

In [27]:
```
sns.pairplot(df1[num], kind='scatter', diag_kind='hist', palette='Rainbow')
plt.show()
```



Now that we have preprocessed the data, exporting the dataframe to a csv file for further plotting of charts in Tableau.

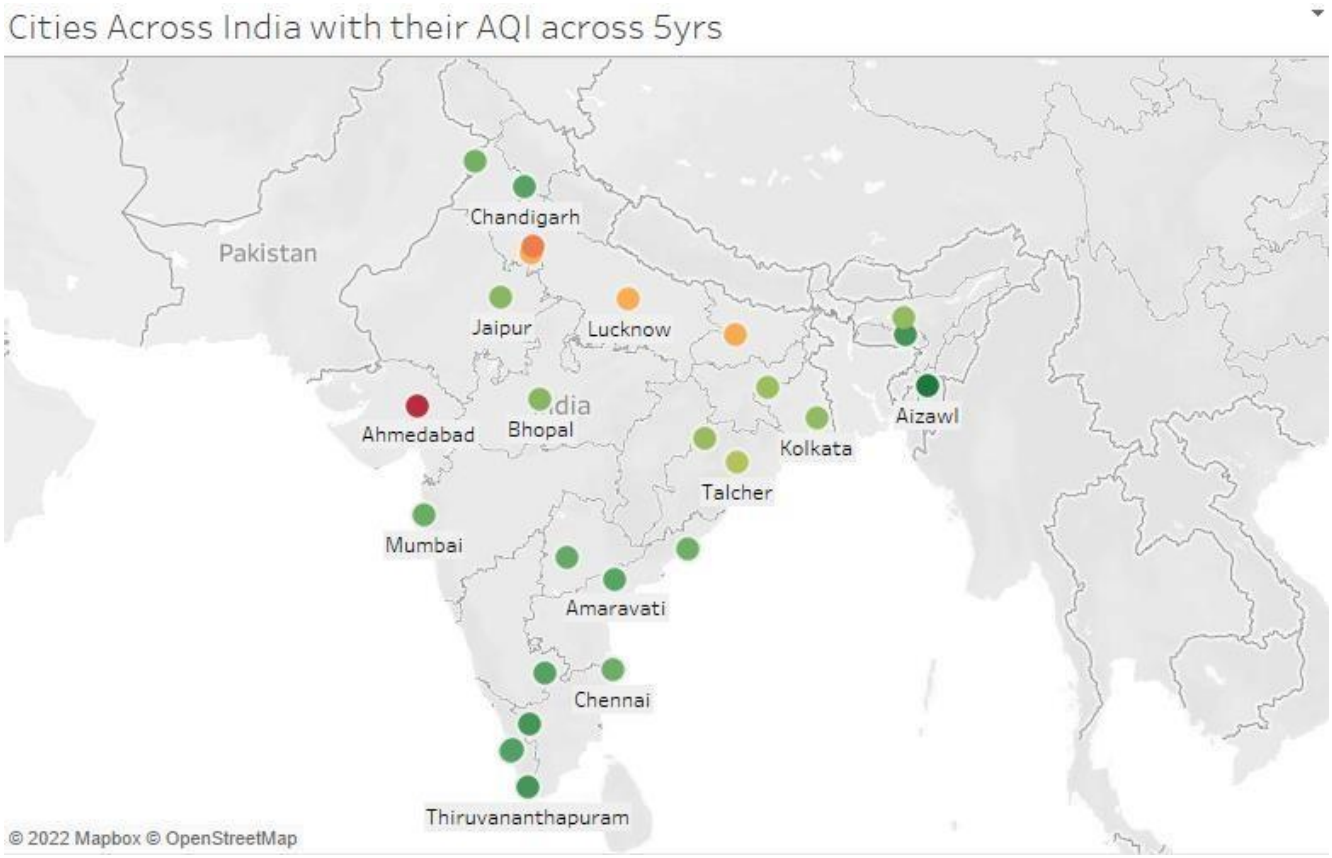
In [19]:
```
df1.to_csv("Preprocessed_AQI_Data.csv")
```

AIR QUALITY INDEX: PRE & POST COVID ANALYSIS ON TABLEAU

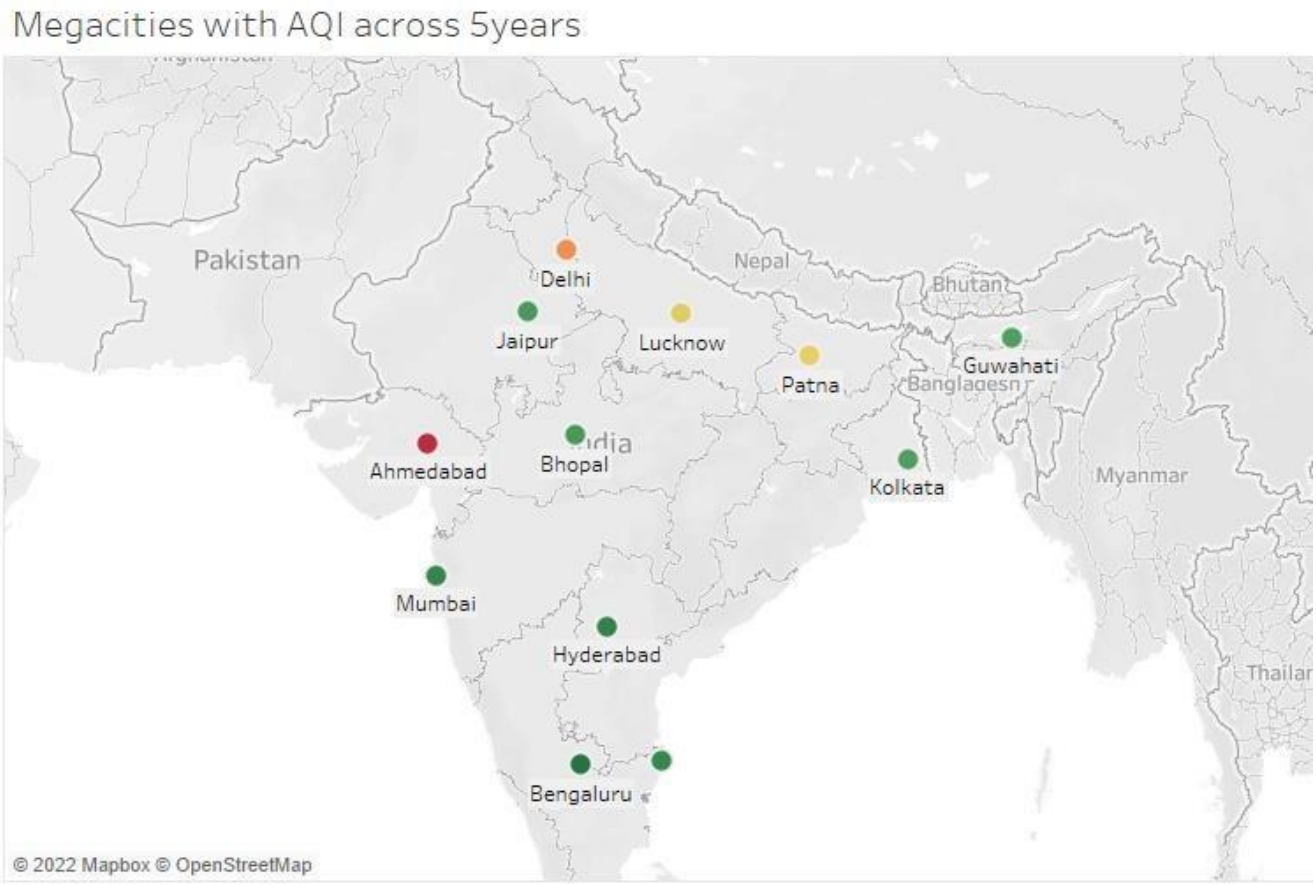
The air quality index tells you about how clean or polluted your air is. The main objective of AQI is to focus on the health effects, you may experience within a few hours or days after breathing polluted air.

In our case study we are analysing 2015-2020 data with the objective to throw some light on the health hazards which the polluted air creates.

The raw data has 26 cities shown by the geolocation map below:



We would also analyse the pre and post covid AQI to draw some insights on how covid affected the AQIs of 12 Megacities across India.



BTX

Benzene is a widely used industrial chemical. Benzene is found in crude oil and is a major part of gasoline. It's used to make plastics, resins, synthetic fibres, rubber lubricants, dyes, detergents, drugs and pesticides. Toluene is typically used in the production of paints, rubber, lacquers and adhesives. Xylenes is used as a solvent in the manufacturing of chemicals, agricultural sprays, adhesives and coatings, as an ingredient in aviation fuel.

Health Hazards: Benzene can cause bone marrow not to produce enough red blood cells, which can lead to anaemia. Also, it can damage the immune system by changing blood levels of antibodies and causing the loss of white blood cells. Exposure to Toluene and Xylenecan irritate the eyes, nose, skin, and throat. Xylene can also cause headaches, dizziness, confusion, loss of muscle coordination, and in high doses, death.

From the graph below we see that emission of BTX has increased in the post-covid period which leads us to the fact that Benzyne is an important industrial chemical to produce plastic from which we can derive the fact that due to increased production of PPE (Personal Protective

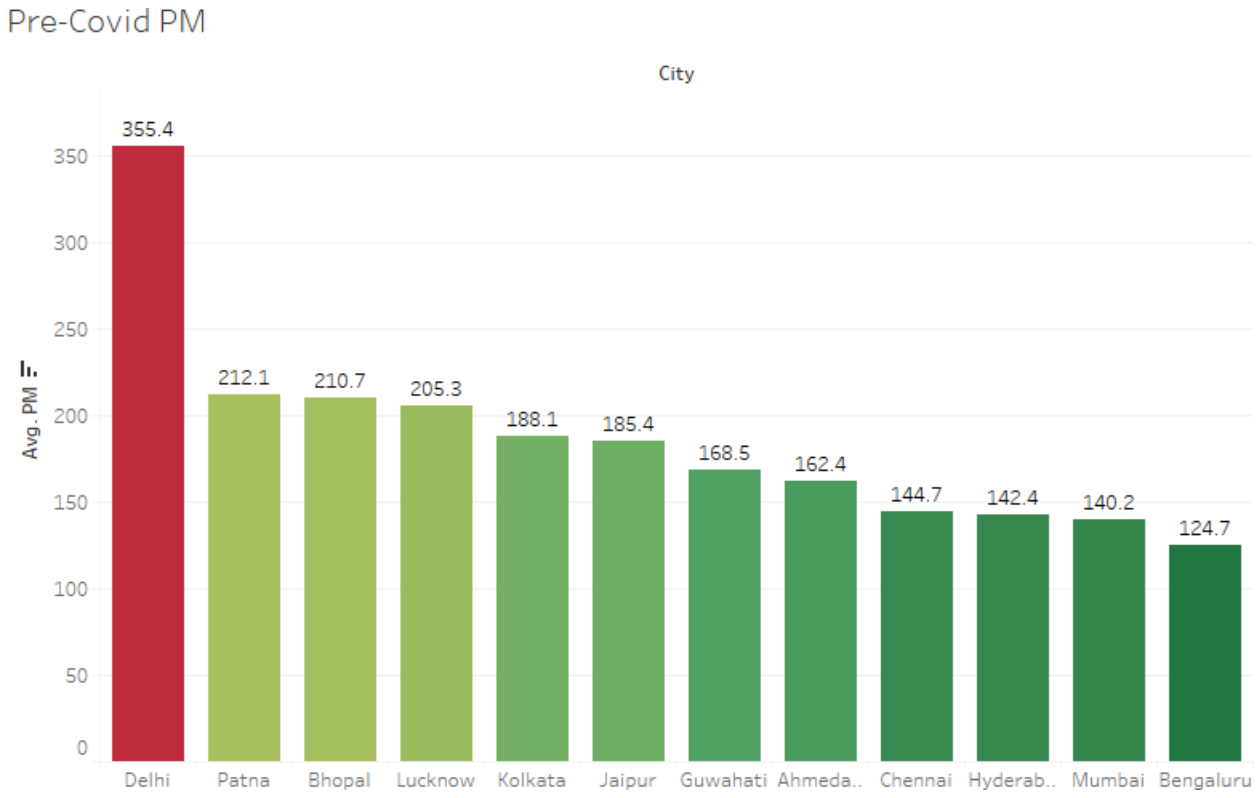
Equipment) kits, the Benzene emission rose. At the same time, Xylene is used in aviation fuel from which we can conclude that coming back of migrants to home or to their native places increased due to Covid.

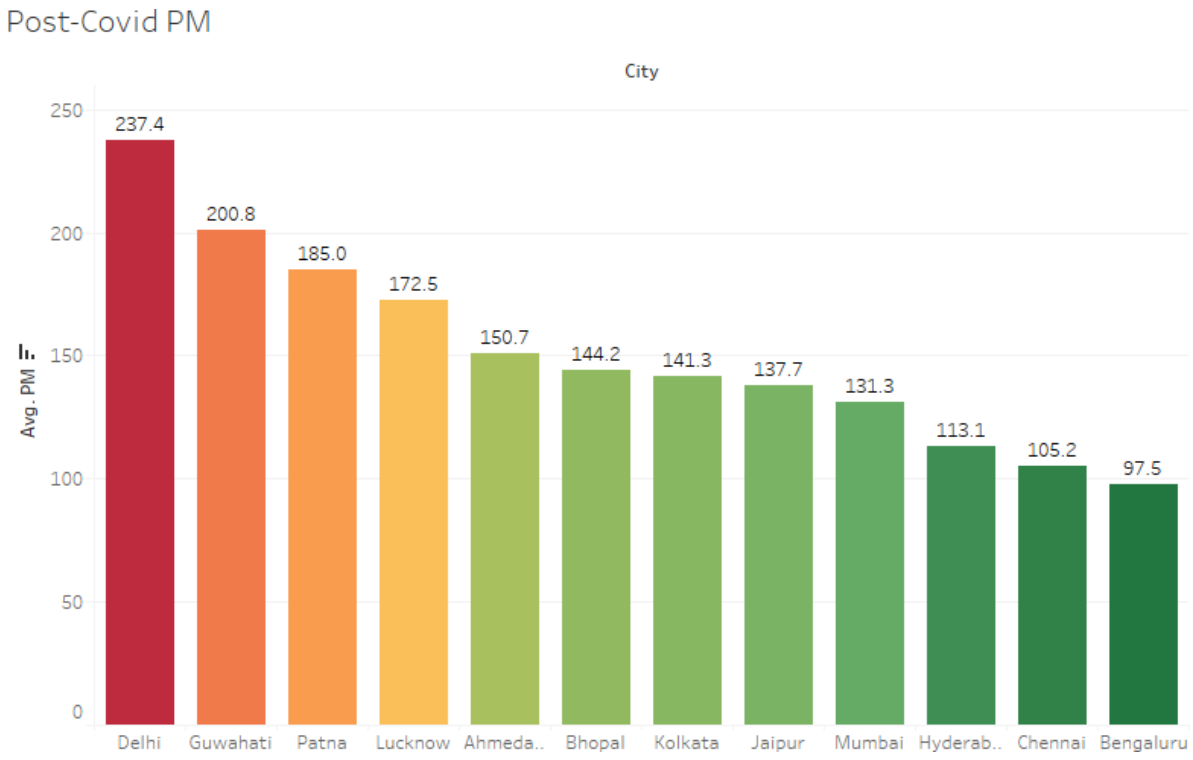
PM

PM10 and PM2.5 often derive from different emissions sources, and also have different chemical compositions. Emissions from combustion of gasoline, oil, diesel fuel or wood produce much of the PM2.5 pollutions found in outdoor air, as well as a significant proportion of PM10.

Exposure to fine particles can cause short-term health effects such as eye, nose, throat and lung irritation, coughing, sneezing, runny nose and shortness of breath. Further it also affects lung function and worsen medical conditions such as asthma and heart disease.

The amount of emission of PM 2.5 and PM 10 decreased in post covid area, which leads us to the fact that the air quality improved significantly in the megacities due to Lockdown.





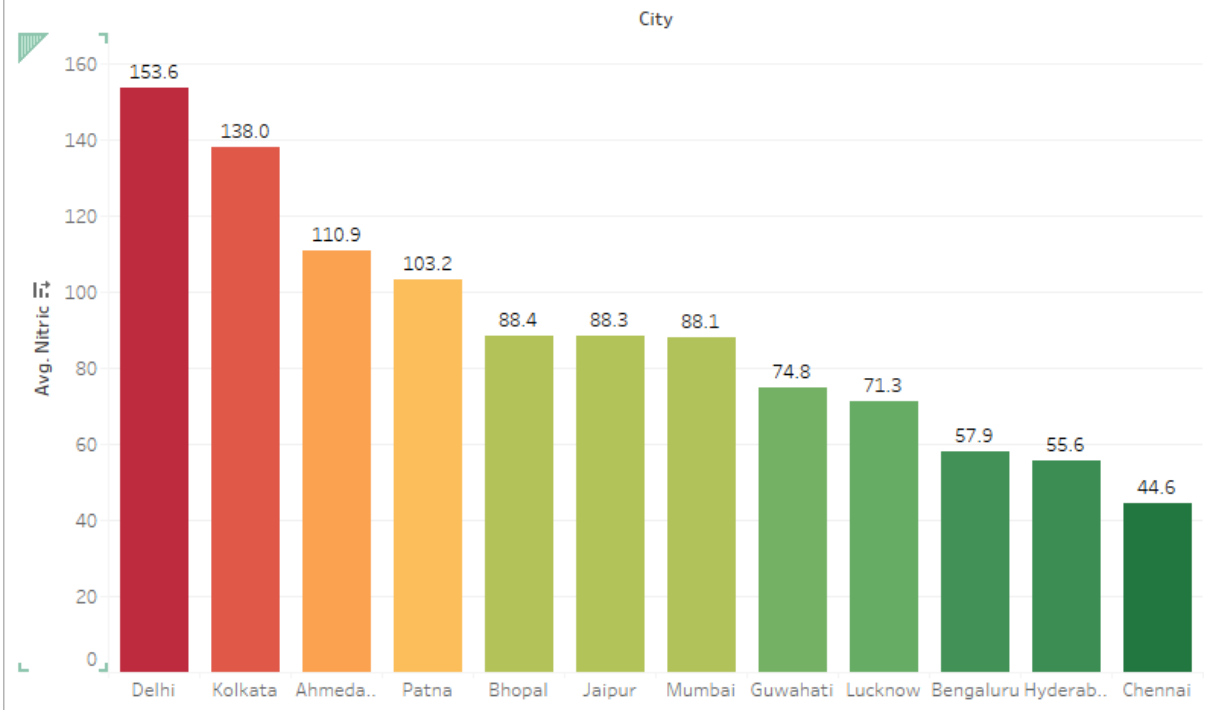
NITROGEN OXIDE

NOX often appears as a brownish gas. It is a strong oxidizing agent and plays a major role in the atmospheric reactions with volatile organic compounds (VOC) that produce ozone (smog) on hot summer days.

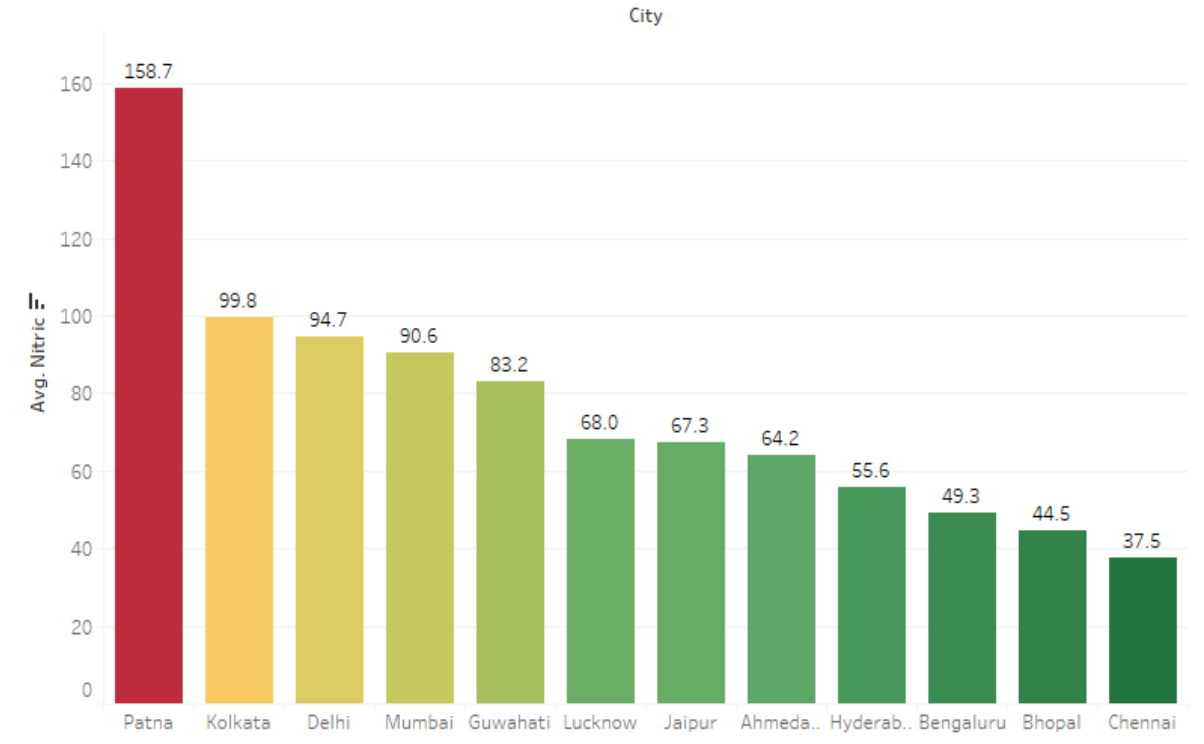
The nitrogen oxides family (NO, NO2, NOX) can react with ammonia, VOCs, and other compounds to form PM 2.5 pollution that easily penetrates into sensitive and deep parts of the lung causing respiratory diseases like emphysema and bronchitis. NOX also can aggravate a pre-existing heart disease, leading to premature death.

When we are comparing the pre-covid and post-covid graphs of the nitrogen family, we see a significant decrease in its emission due to the fact that it is a part of vehicular pollution content and again due to Lockdown, the vehicular emissions dropped.

Pre-Covid Nitric



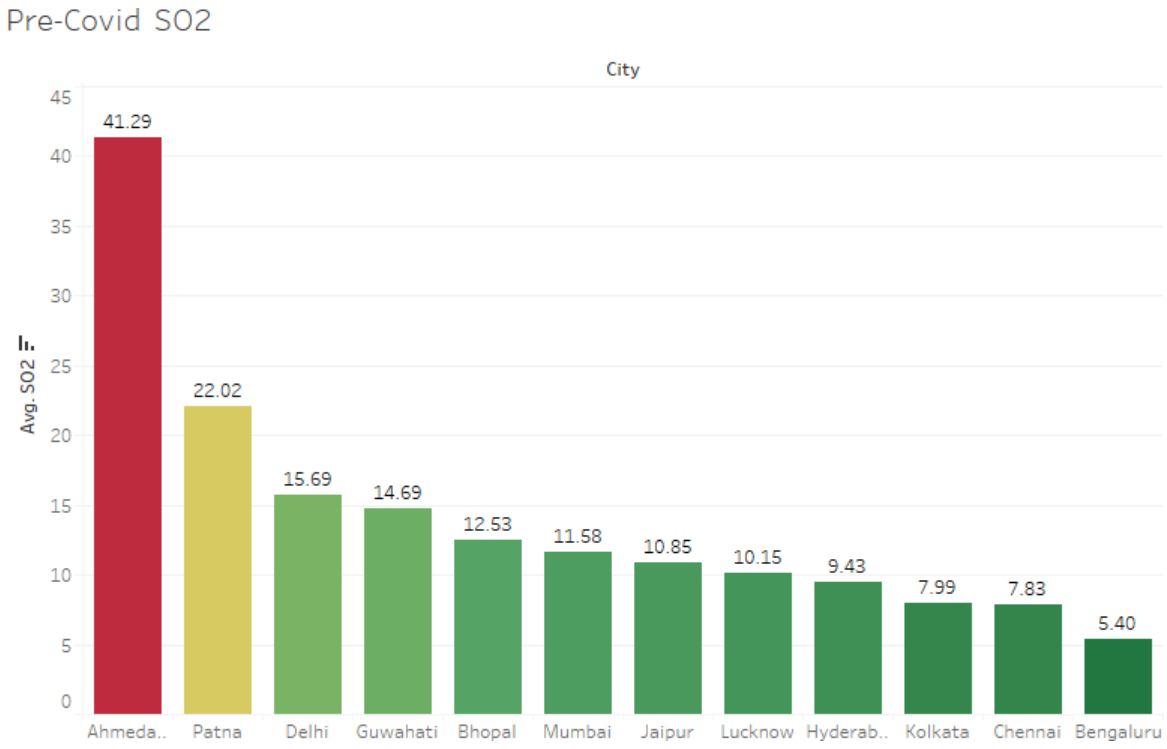
Post-Covid Nitric

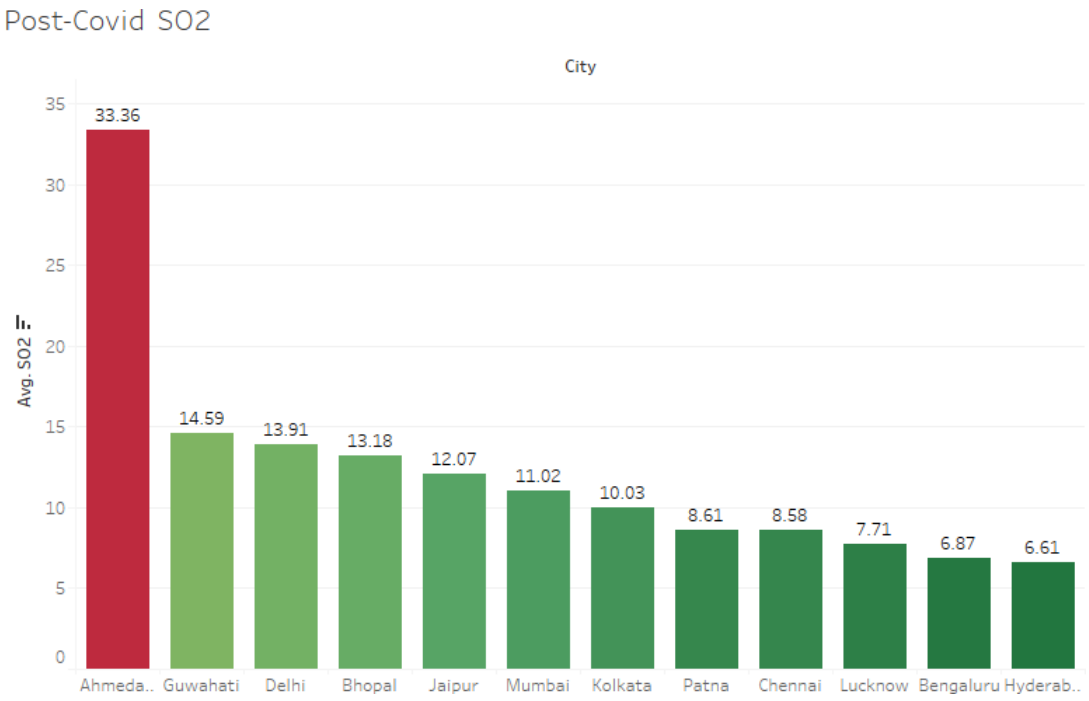


SO2

Sulphur dioxide is used in many industries. It’s used to manufacture sulfuric acid, paper, and food preservatives.

Exposure to sulphur dioxide may irritate the eyes, nose, and throat. Symptoms include nasal mucus, choking, cough, reflex bronchi constriction, and when liquid: frostbite Workers may be harmed from exposure to sulphur dioxide. The level of exposure depends upon the dose, duration, and work being done.



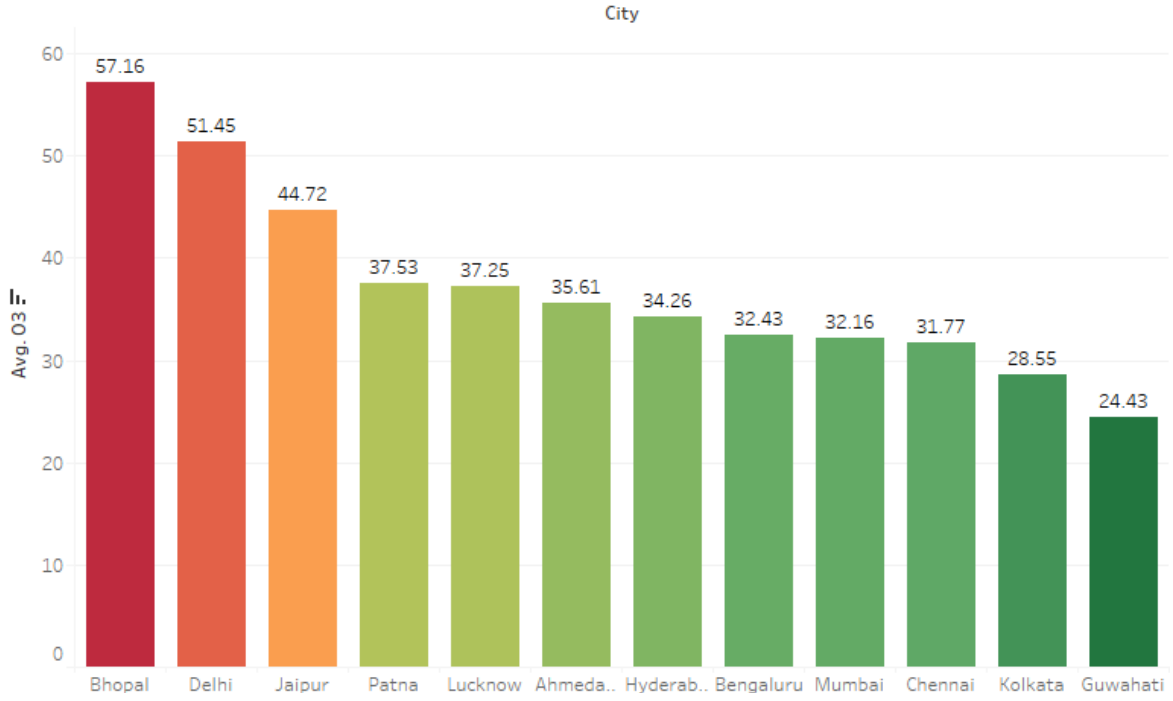


O3

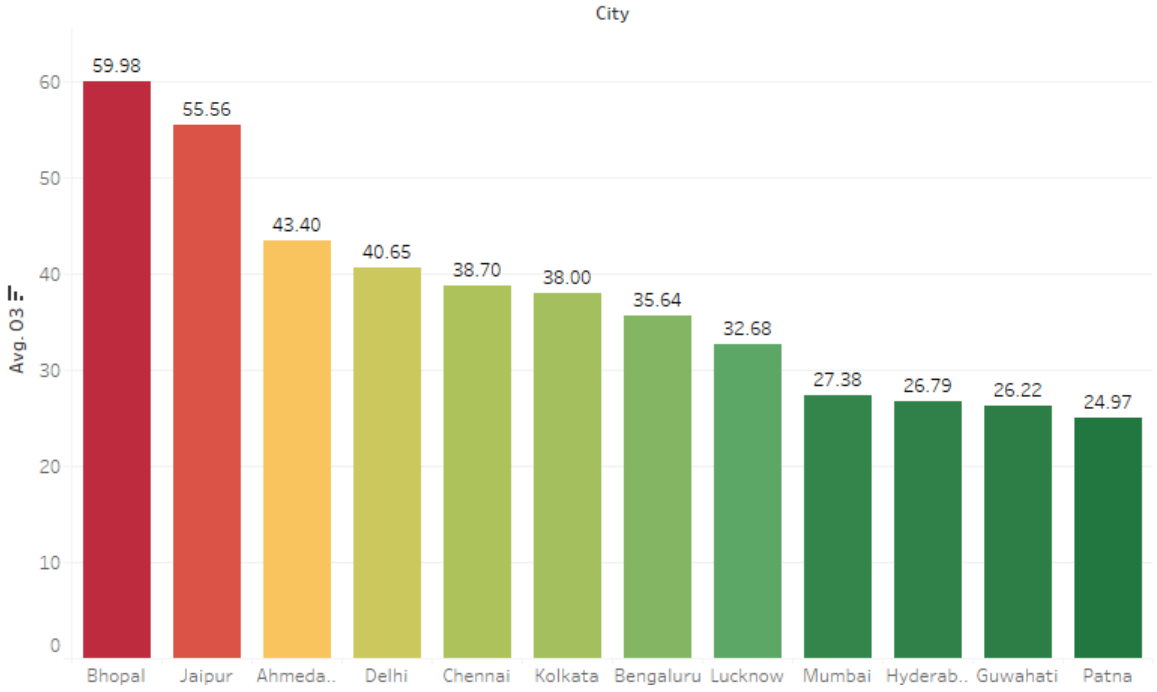
Ozone is used in many industries. It is used for purifying the air and drinking water, in industrial waste treatment, oils, bleaching, and waxes, and to make other chemicals.

Ozone (O_3) is a colourless to blue gas with a pungent odour. Exposure to ozone may cause headaches, coughing, dry throat, shortness of breath, a heavy feeling in the chest, and fluid in the lungs. Higher levels of exposure can lead to more severe symptoms. Chronic exposure may lead to asthma.

Pre-Covid O3



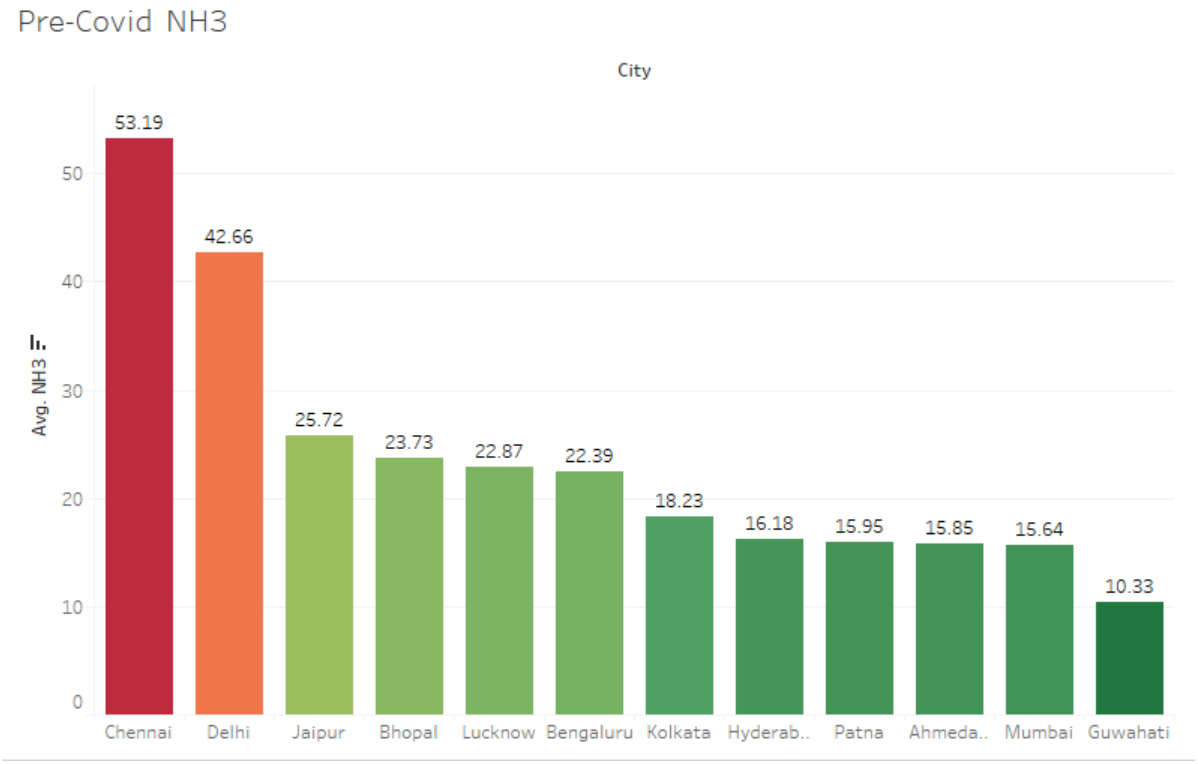
Post-Covid O3

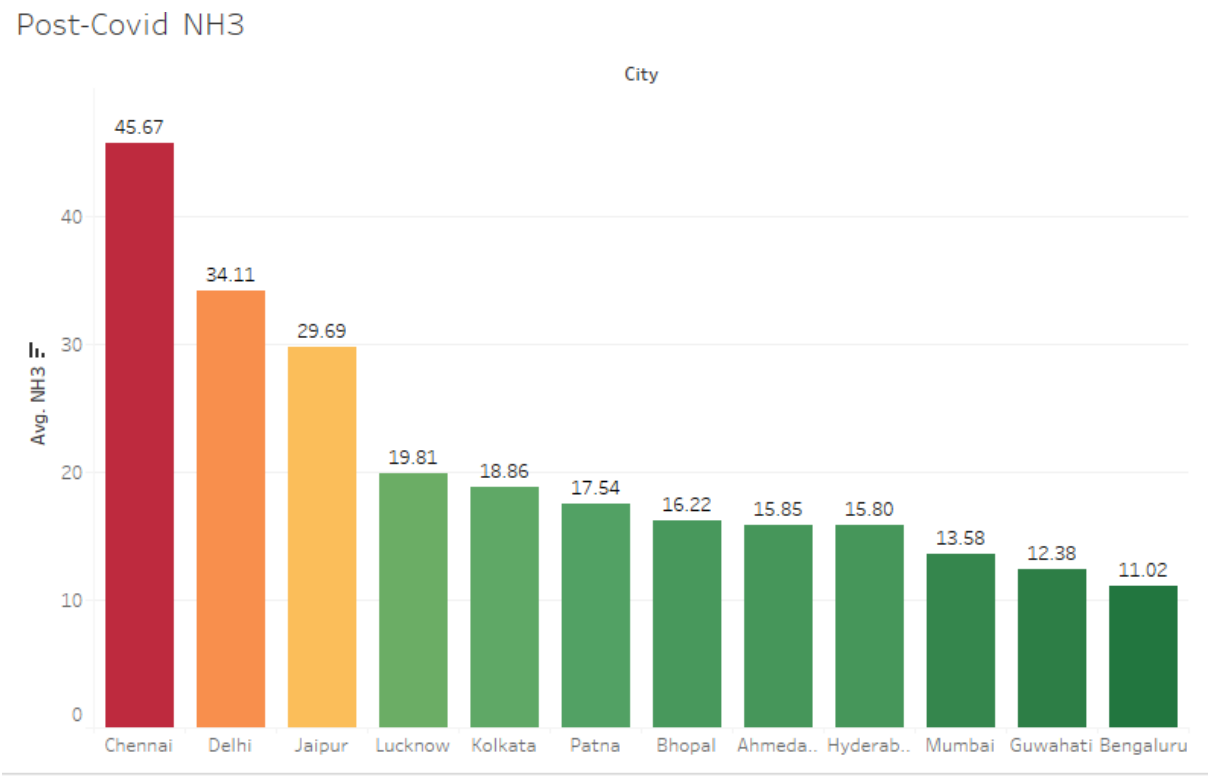


NH3

Ammonia is used in many industries like fertilizer, rubber, nitric acid, urea, plastics, Fibres, synthetic resin, solvents, and other chemicals. It is even used in mining and metallurgy, refrigerant in food processing, produce ice, and petroleum refining.

Its contact can severely irritate and burn the skin and eyes leading to eye damage. Exposure can irritate the eyes, nose and throat. Inhaling Ammonium Hydroxide can irritate the lungs. Higher exposures may cause a build-up of fluid in the lungs (pulmonary Edema), a medical emergency.



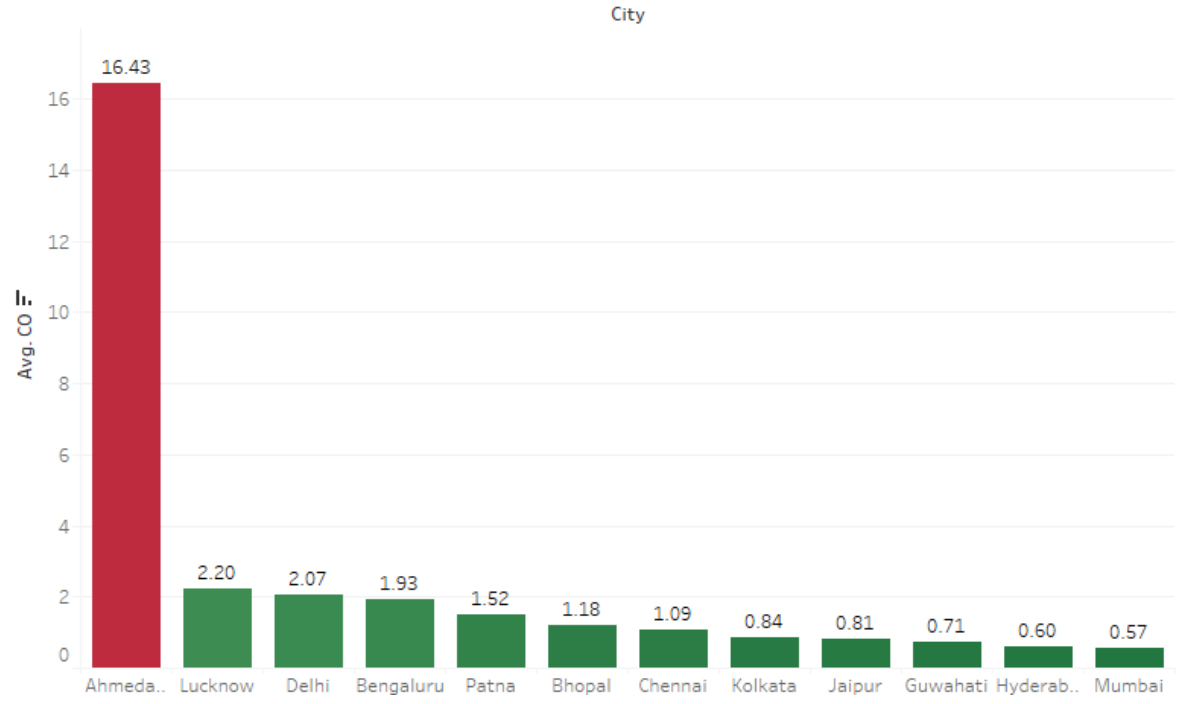


CO

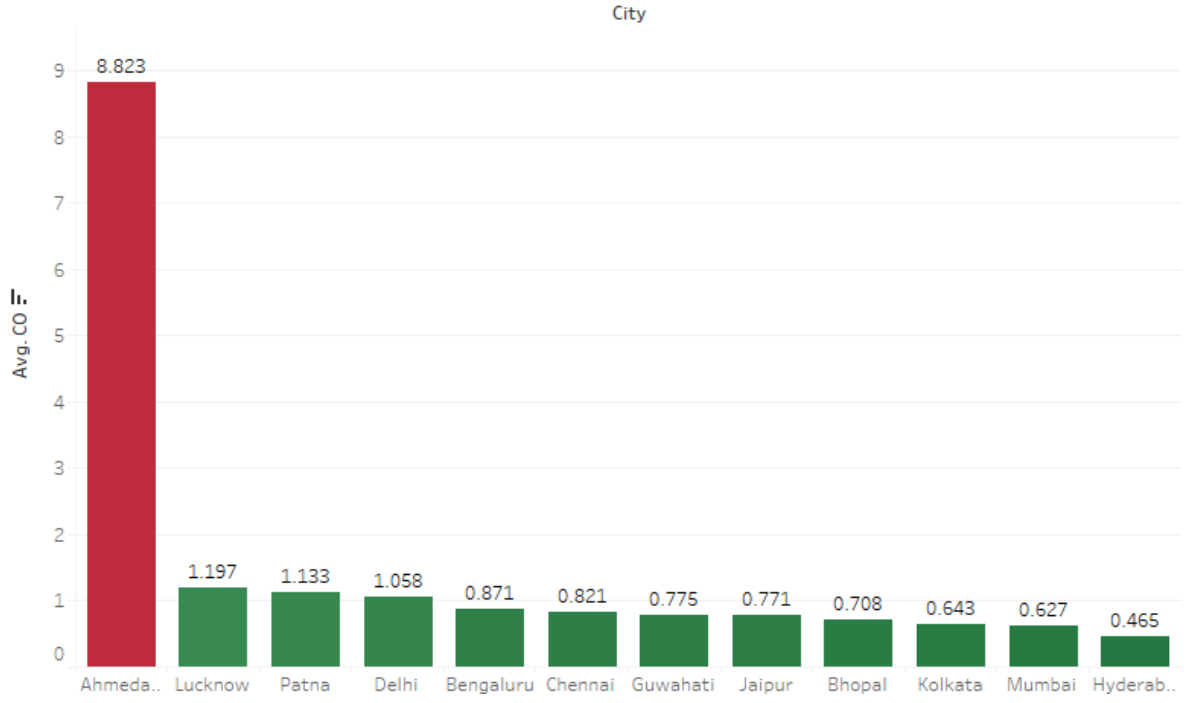
It is used in the purification of nickel. It is used in water gas shift reactions to produce hydrogen. It is used in meat colouring. It is used as a reducing agent.

Carbon monoxide is harmful when breathed because it displaces oxygen in the blood and deprives the heart, brain and other vital organs of oxygen. Large amounts of CO can overcome you in minutes without warning — causing you to lose consciousness and suffocate.

Pre-Covid CO



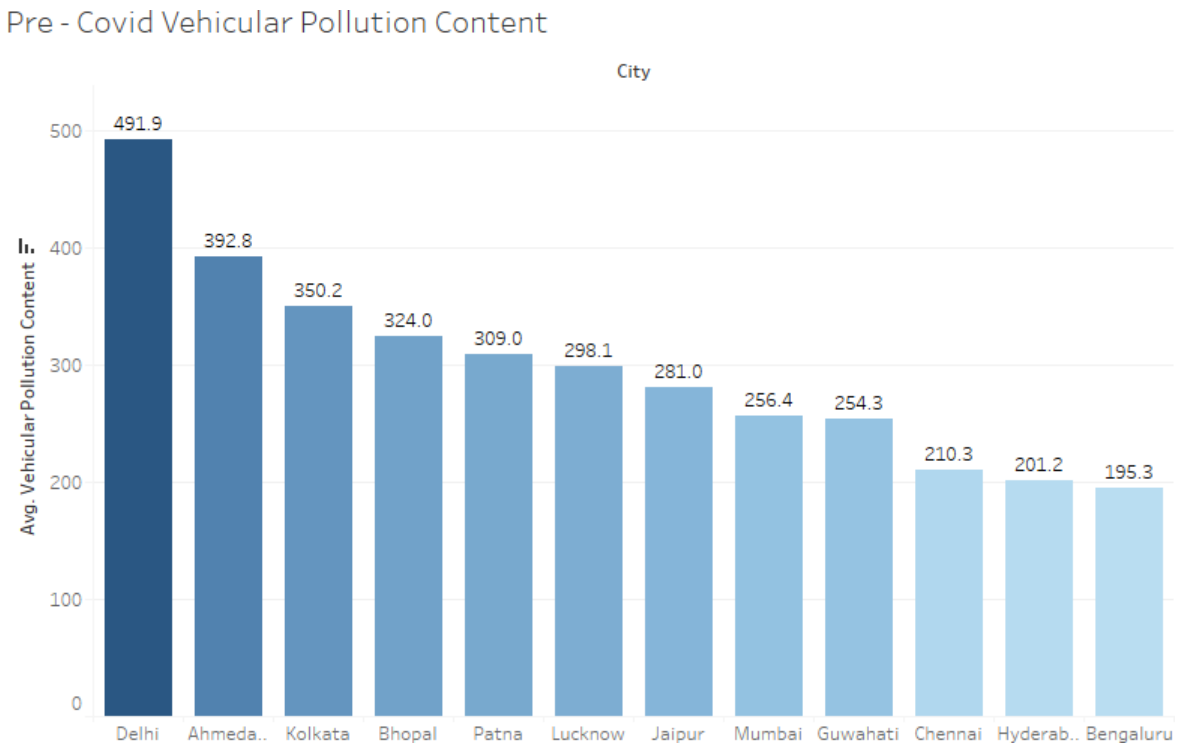
Post-Covid CO

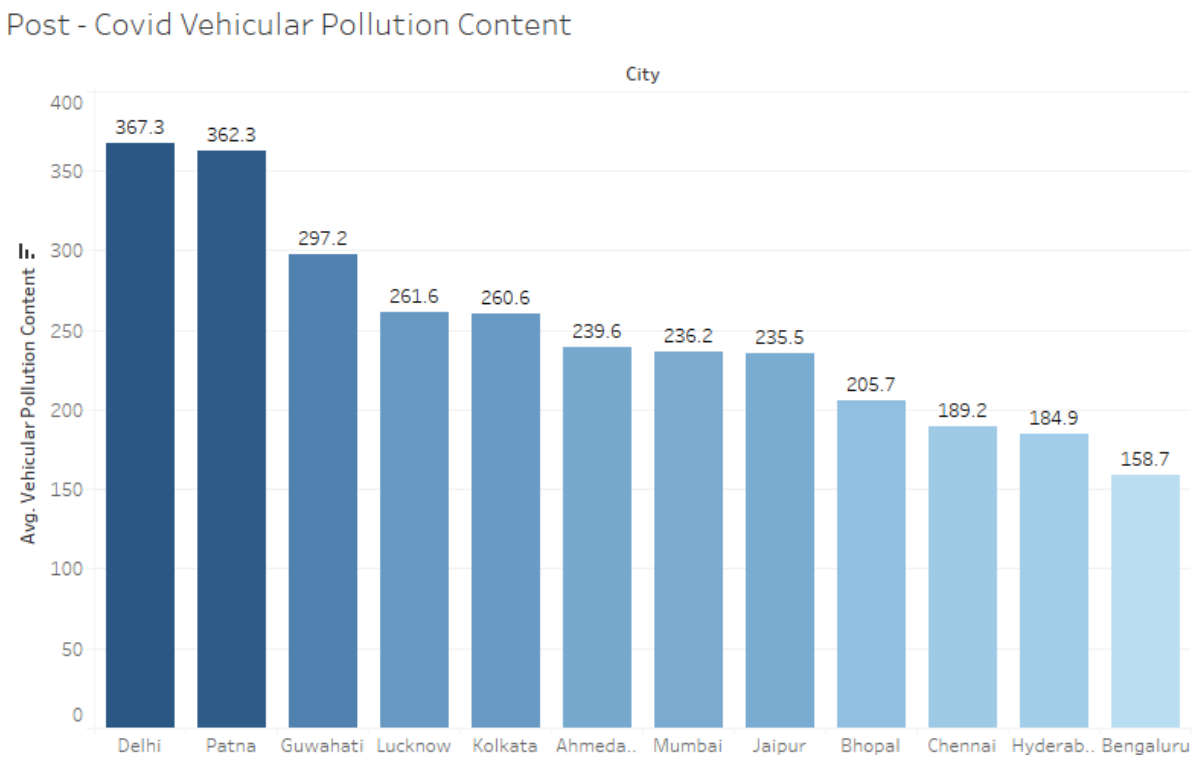


VEHICULAR POLLUTION CONTENT

The contents of Vehicular Pollution are – CO, NO, NO2, NOX, PM 2.5 and PM 10. Combining all these pollutants, we create the measure for Vehicular pollution content. The graph below shows this.

As we know that, in Delhi, poor quality air damages irreversibly the lungs of 2.2 million or 50 percent of all children. Dropped vehicular pollution in the post covid scenario leads us to the fact that the quality of air improved significantly which is a good sign for the people staying in these megacities.

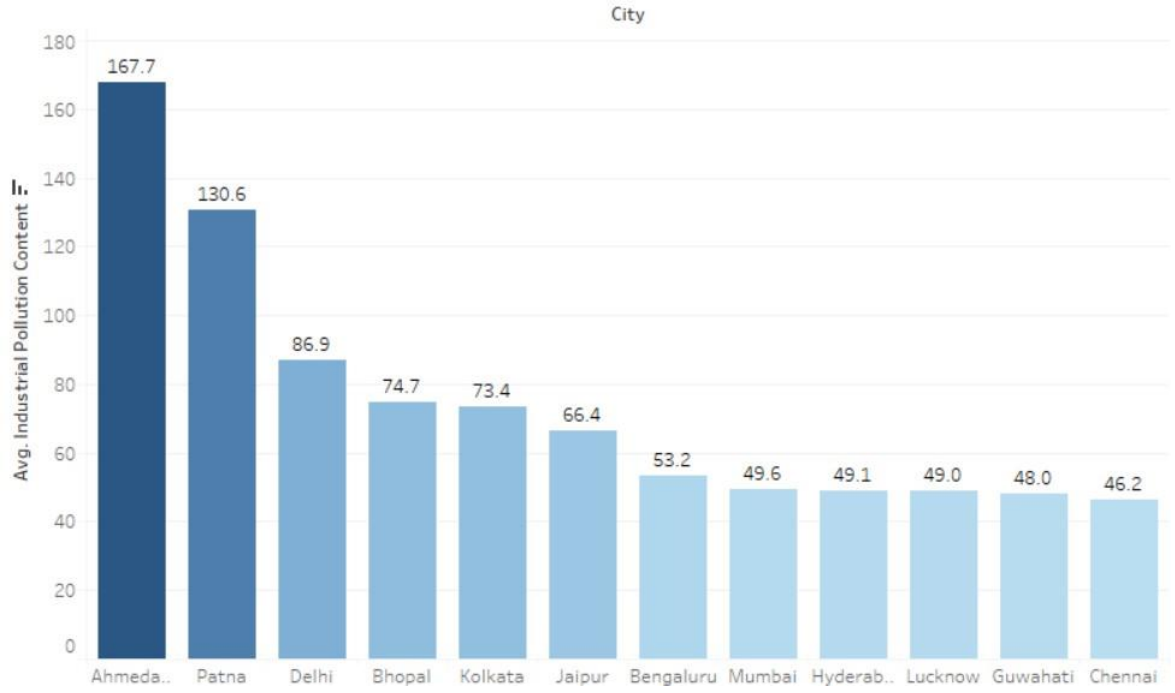




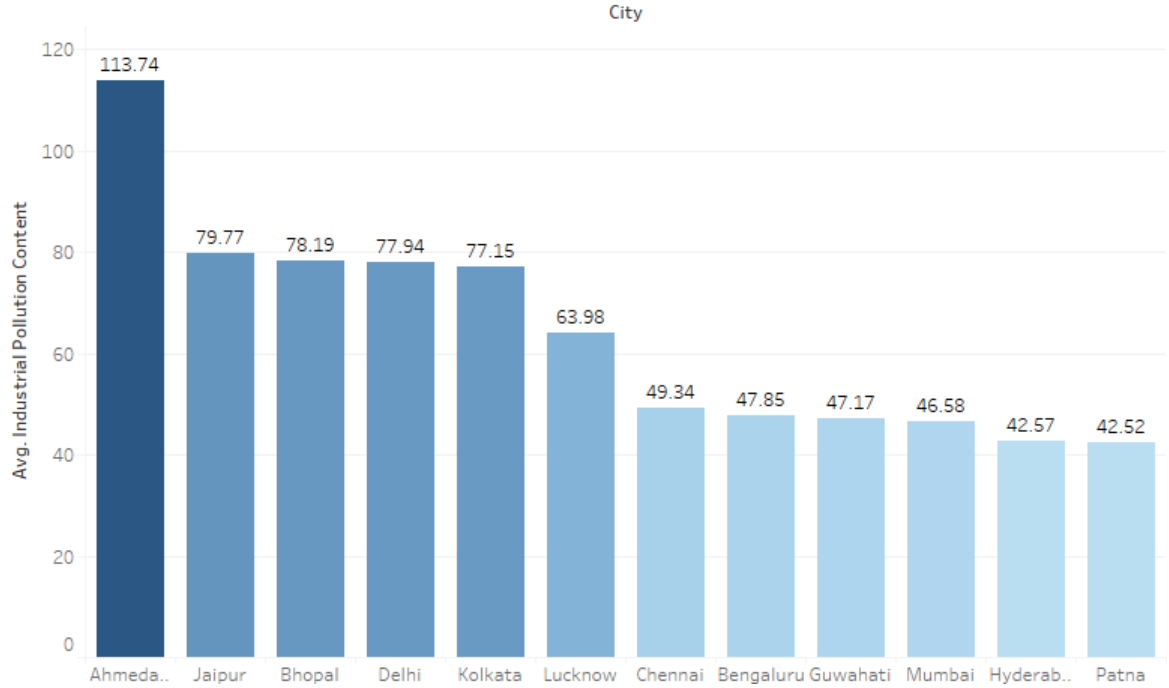
INDUSTRIAL POLLUTION CONTENT

The contents of Industrial Pollution are – Benzene, Toluene, Xylene, SO2 and O3. From the graphs below, we can derive the fact that for some cities where the concentration of factories is high like Bhopal, Lucknow, Kolkata, the industrial Pollution increased whereas in some of the cities it decreased because many industries had to shut as a result of Lockdown.

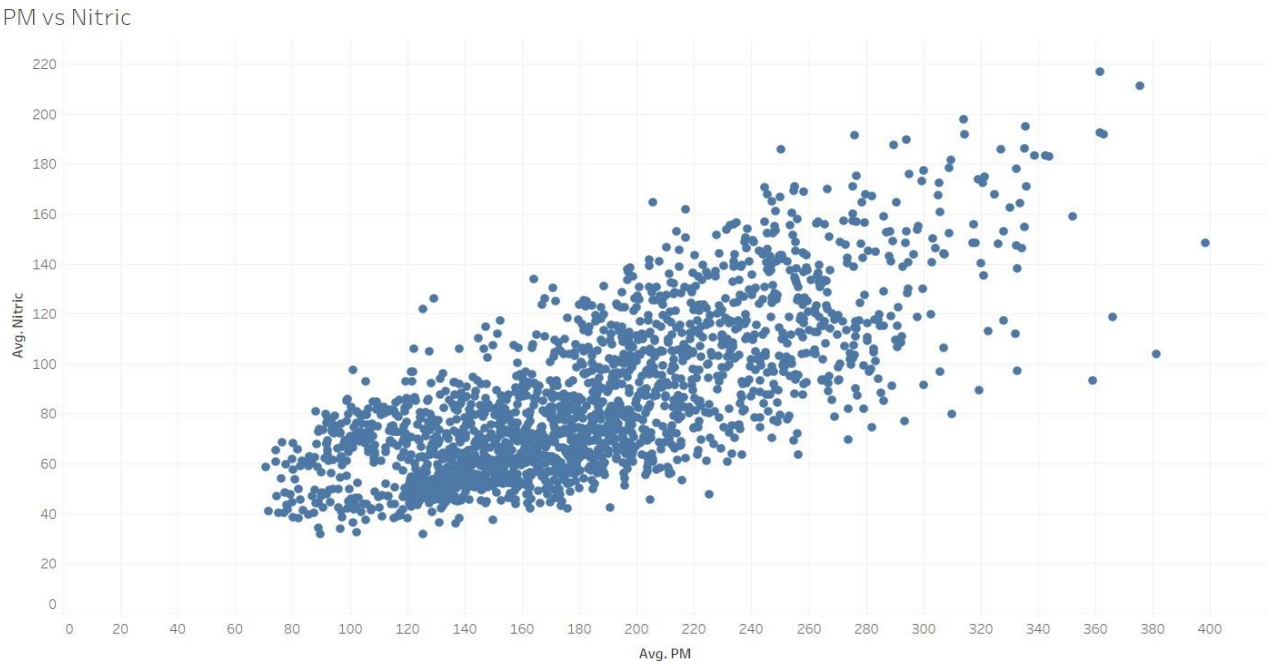
Pre - Covid Industrial Pollution Content



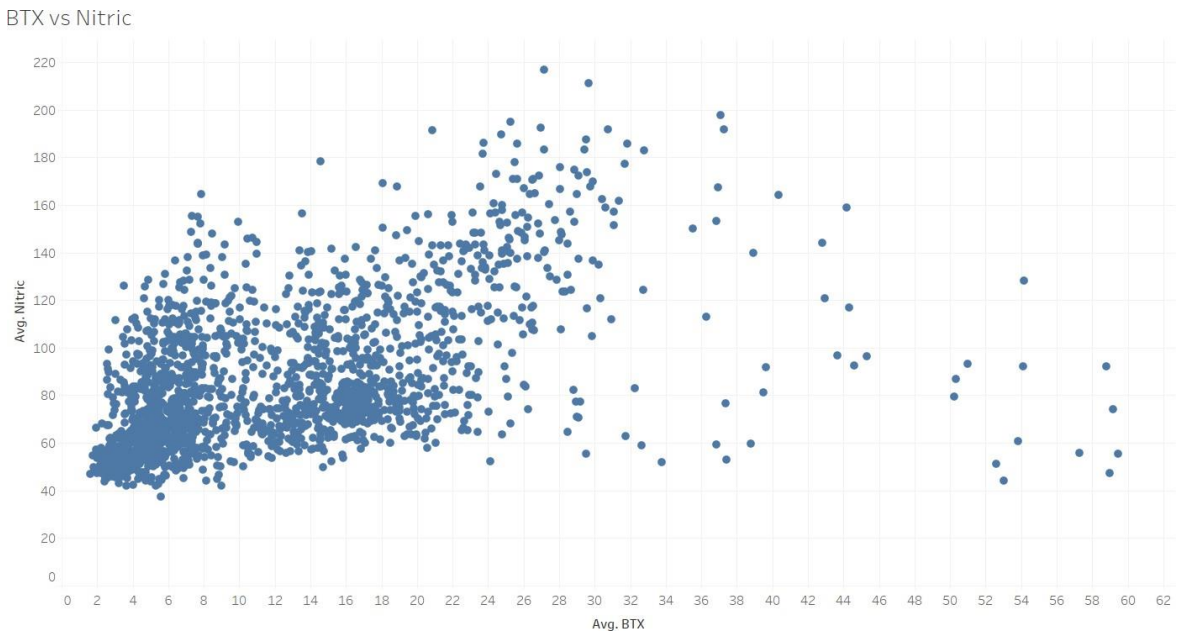
Post - Covid Industrial Pollution Content

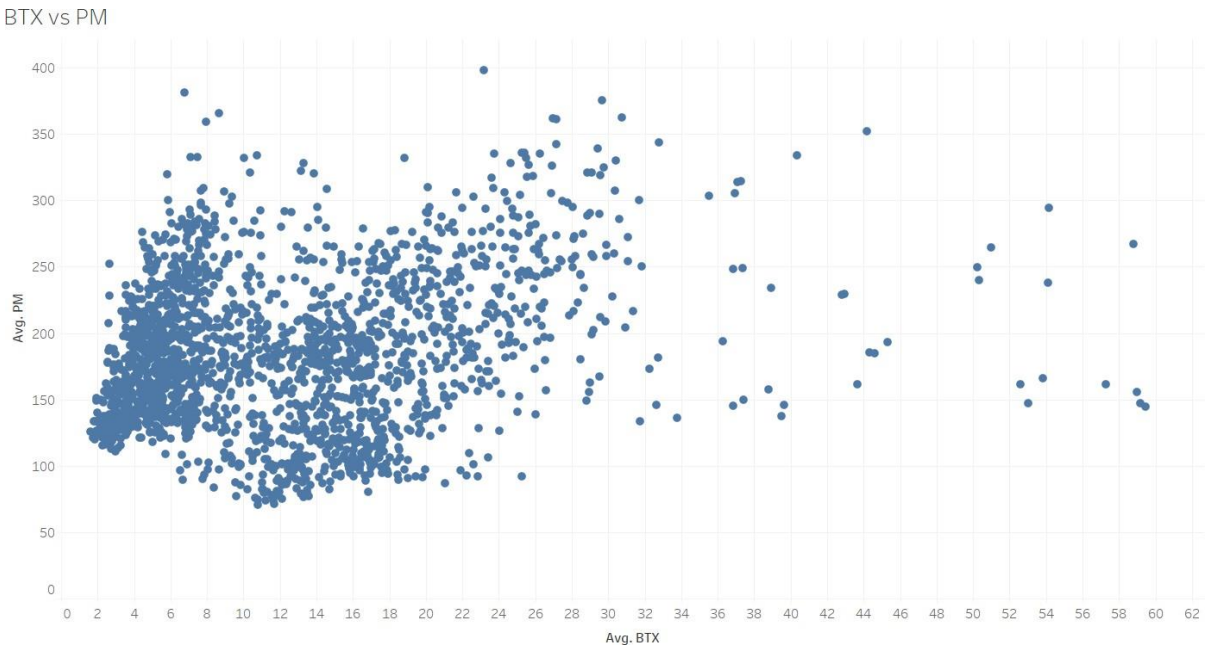


Here we are plotting a scatter plot between pollutant groups.



Here we can see a good correlation between Average of PM and Average of Nitric Oxides





CONCLUSION

Here we performed an exploratory data analysis on the dataset air quality index of India for the years 2015-20. By performing pre and post covid analysis we found that the air quality index has seen a decrease in the megacities which is a good sign for the environment.