

# Air Quality Index In India

This is the project on AQI in India of last few year(2015-2020).The dataset consists of AQIs of some of the cities. I got this dataset for this project from Kaggle(<https://www.kaggle.com/rohanrao/air-quality-data-in-india>) and was recommended by jovian itself in thier forum. I tried to extract a lot of information from the dataset using Pandas, matplotlib, seaborn etc; about the deviation of AQI in some years,how it was deviating in various months, how it is different in various Cities, which city got the highest AQI during these years, etc

I really want to thank [Data Analysis with Python: Zero to Pandas \(zerotopandas.com\)](https://zerotopandas.com) which helped me in learning a lot about Data Analysis, I have learned various operations which can be done by importing various modules and libraries so that information can be easily extraxted out of the dataset.

## Downloading the Dataset

I have downloaded the dataset from the Kaggle(<https://www.kaggle.com/rohanrao/air-quality-data-in-india>) regarding the AQI index in some cities in India. It is the CSV file.

In [1]:

```
1 !pip install jovian opendatasets --upgrade --quiet
```

Let's begin by downloading the data, and listing the files within the dataset.

In [2]:

```
1 dataset_url = 'https://www.kaggle.com/rohanrao/air-quality-data-in-india'
```

In [4]:

```
1 import opendatasets as od
2 od.download(dataset_url)
```

Please provide your Kaggle credentials to download this dataset. Learn more:

<http://bit.ly/kaggle-creds> (<http://bit.ly/kaggle-creds>)

Your Kaggle username: vedangisharma

Your Kaggle Key: .....

8%|█| 6.00M/72.9M [00:00<00:01, 61.3MB/s]

Downloading air-quality-data-in-india.zip to ./air-quality-data-in-india

100%|██████████| 72.9M/72.9M [00:01<00:00, 64.8MB/s]

The dataset has been downloaded and extracted.

In [5]:

```
1 data_dir = './air-quality-data-in-india'
```

In [6]:

```
1 import os
2 os.listdir(data_dir)
```

Out[6]:

```
['city_hour.csv',
 'city_day.csv',
 'station_day.csv',
 'stations.csv',
 'station_hour.csv']
```

Let us save and upload our work to Jovian before continuing.

In [7]:

```
1 project_name = "aqi-in-india"
```

In [8]:

```
1 !pip install jovian --upgrade -q
```

In [9]:

```
1 import jovian
```

In [10]:

```
1 jovian.commit(project=project_name)
```

<IPython.core.display.Javascript object>

[jovian] Attempting to save notebook..

[jovian] Updating notebook "vedangi001/aqi-in-india" on <https://jovian.ai> (<https://jovian.ai>)

[jovian] Uploading notebook..

[jovian] Committed successfully! <https://jovian.ai/vedangi001/aqi-in-india>  
(<https://jovian.ai/vedangi001/aqi-in-india>)

Out[10]:

```
'https://jovian.ai/vedangi001/aqi-in-india'
```

## Data Preparation and Cleaning

the dataset consists the data of various cities in each day for past few years. Some rows have a lot of NAN values, so we need to remove them, as we can not gather a lot of information from those rows.

In [11]:

```
1 import pandas as pd
```

In [12]:

```
1 dataset=pd.read_csv('air-quality-data-in-india/city_day.csv')
```

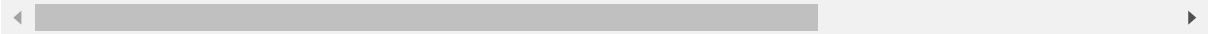
In [13]:

```
1 dataset
```

Out[13]:

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	B
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	
1	Ahmedabad	2015-01-02	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	
2	Ahmedabad	2015-01-03	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	
3	Ahmedabad	2015-01-04	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	
4	Ahmedabad	2015-01-05	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	
...	...	...	...	...	...	...	...	...	...	...	...	
29526	Visakhapatnam	2020-06-27	15.02	50.94	7.68	25.06	19.54	12.47	0.47	8.55	23.30	
29527	Visakhapatnam	2020-06-28	24.38	74.09	3.42	26.06	16.53	11.99	0.52	12.72	30.14	
29528	Visakhapatnam	2020-06-29	22.91	65.73	3.45	29.53	18.33	10.71	0.48	8.42	30.96	
29529	Visakhapatnam	2020-06-30	16.64	49.97	4.05	29.26	18.80	10.03	0.52	9.84	28.30	
29530	Visakhapatnam	2020-07-01	15.00	66.00	0.40	26.85	14.05	5.20	0.59	2.10	17.05	

29531 rows × 16 columns



In [14]:



```
1 dataset.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
```

In [15]:



```
1 dataset.describe()
```

Out[15]:

	PM2.5	PM10	NO	NO2	NOx	NH3
<b>count</b>	24933.000000	18391.000000	25949.000000	25946.000000	25346.000000	19203.000000
<b>mean</b>	67.450578	118.127103	17.574730	28.560659	32.309123	23.483476
<b>std</b>	64.661449	90.605110	22.785846	24.474746	31.646011	25.684275
<b>min</b>	0.040000	0.010000	0.020000	0.010000	0.000000	0.010000
<b>25%</b>	28.820000	56.255000	5.630000	11.750000	12.820000	8.580000
<b>50%</b>	48.570000	95.680000	9.890000	21.690000	23.520000	15.850000
<b>75%</b>	80.590000	149.745000	19.950000	37.620000	40.127500	30.020000
<b>max</b>	949.990000	1000.000000	390.680000	362.210000	467.630000	352.890000

In [16]:



```
1 dataset.AQI[dataset.AQI==2049]
```

Out[16]:

```
1145    2049.0
Name: AQI, dtype: float64
```

In [17]:

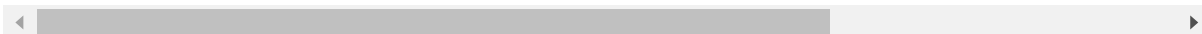


```
1 df=dataset[dataset.AQI_Bucket=='Severe']
2 df
```

Out[17]:

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benz
30	Ahmedabad	2015-01-31	94.52	NaN	24.39	32.66	52.61	NaN	24.39	67.39	111.33	
31	Ahmedabad	2015-02-01	135.99	NaN	43.48	42.08	84.57	NaN	43.48	75.23	102.70	
32	Ahmedabad	2015-02-02	178.33	NaN	54.56	35.31	72.80	NaN	54.56	55.04	107.38	
33	Ahmedabad	2015-02-03	139.70	NaN	30.61	28.40	56.73	NaN	30.61	33.79	73.60	
43	Ahmedabad	2015-02-13	103.36	NaN	18.18	39.56	57.33	NaN	18.18	80.43	40.11	1
...	...	...	...	...	...	...	...	...	...	...	...	
26454	Talcher	2019-02-15	68.38	336.82	35.57	NaN	29.15	7.66	2.52	16.17	7.50	
26461	Talcher	2019-02-22	116.52	768.16	27.89	NaN	24.94	7.23	4.20	16.71	3.71	
26463	Talcher	2019-02-24	71.07	369.02	32.85	NaN	27.66	7.52	2.96	6.63	8.00	
26719	Talcher	2019-11-07	48.32	512.41	41.82	1.21	41.62	8.49	2.11	36.43	13.67	
26720	Talcher	2019-11-08	57.45	493.34	39.71	1.90	40.38	8.35	1.68	36.34	13.75	

1338 rows × 16 columns



The dataset consist of so many rows having different AQIs but this project is mainly for the cities having higher AQI or where conditions is severe. So, the other AQI rows where condition is not so Severe can be removed

In [18]:



```
1 columns=['City','Date','PM2.5','NO','NO2','NOx','CO','SO2','O3','AQI']
```

In [19]:



```
1 dfs=df[columns].copy()
2 dfs=dfs.dropna()
3 dfs
```

Out[19]:

	City	Date	PM2.5	NO	NO2	NOx	CO	SO2	O3	AQI
30	Ahmedabad	2015-01-31	94.52	24.39	32.66	52.61	24.39	67.39	111.33	514.0
31	Ahmedabad	2015-02-01	135.99	43.48	42.08	84.57	43.48	75.23	102.70	782.0
32	Ahmedabad	2015-02-02	178.33	54.56	35.31	72.80	54.56	55.04	107.38	914.0
33	Ahmedabad	2015-02-03	139.70	30.61	28.40	56.73	30.61	33.79	73.60	660.0
43	Ahmedabad	2015-02-13	103.36	18.18	39.56	57.33	18.18	80.43	40.11	510.0
...	...	...	...	...	...	...	...	...	...	...
26379	Talcher	2018-12-02	100.93	25.19	0.02	23.45	1.86	18.29	8.62	434.0
26429	Talcher	2019-01-21	99.88	23.55	0.03	22.54	2.34	31.05	10.45	456.0
26445	Talcher	2019-02-06	88.48	31.82	3.79	27.05	2.81	19.88	8.99	439.0
26719	Talcher	2019-11-07	48.32	41.82	1.21	41.62	2.11	36.43	13.67	401.0
26720	Talcher	2019-11-08	57.45	39.71	1.90	40.38	1.68	36.34	13.75	516.0

1146 rows × 10 columns

In [20]:



```
1 dfs['Date']=pd.to_datetime(dfs.Date)
2 dfs['year'] = pd.DatetimeIndex(dfs.Date).year
3 dfs['month'] = pd.DatetimeIndex(dfs.Date).month
4 dfs['day'] = pd.DatetimeIndex(dfs.Date).day
5 dfs['weekday'] = pd.DatetimeIndex(dfs.Date).weekday
```

In [21]:

```
1 dfs
```

Out[21]:

	City	Date	PM2.5	NO	NO2	NOx	CO	SO2	O3	AQI	year	mont
30	Ahmedabad	2015-01-31	94.52	24.39	32.66	52.61	24.39	67.39	111.33	514.0	2015	
31	Ahmedabad	2015-02-01	135.99	43.48	42.08	84.57	43.48	75.23	102.70	782.0	2015	
32	Ahmedabad	2015-02-02	178.33	54.56	35.31	72.80	54.56	55.04	107.38	914.0	2015	
33	Ahmedabad	2015-02-03	139.70	30.61	28.40	56.73	30.61	33.79	73.60	660.0	2015	
43	Ahmedabad	2015-02-13	103.36	18.18	39.56	57.33	18.18	80.43	40.11	510.0	2015	
...	...	...	...	...	...	...	...	...	...	...	...	.
26379	Talcher	2018-12-02	100.93	25.19	0.02	23.45	1.86	18.29	8.62	434.0	2018	1
26429	Talcher	2019-01-21	99.88	23.55	0.03	22.54	2.34	31.05	10.45	456.0	2019	
26445	Talcher	2019-02-06	88.48	31.82	3.79	27.05	2.81	19.88	8.99	439.0	2019	
26719	Talcher	2019-11-07	48.32	41.82	1.21	41.62	2.11	36.43	13.67	401.0	2019	1
26720	Talcher	2019-11-08	57.45	39.71	1.90	40.38	1.68	36.34	13.75	516.0	2019	1

1146 rows × 14 columns



In [22]:

```
1 import jovian
```

In [23]:

```
1 jovian.commit()
```

<IPython.core.display.Javascript object>

[jovian] Attempting to save notebook..  
[jovian] Updating notebook "vedangi001/aqi-in-india" on <https://jovian.ai> (<https://jovian.ai>)  
[jovian] Uploading notebook..  
[jovian] Committed successfully! <https://jovian.ai/vedangi001/aqi-in-india>  
(<https://jovian.ai/vedangi001/aqi-in-india>)

Out[23]:

'https://jovian.ai/vedangi001/aqi-in-india'

## Exploratory Analysis and Visualization

**TODO** In this section, I mainly tried to show the information using visual representations such as graphs and plots. Which can help to know out of last few years, when the AQI was highest, the trend during various months, etc.

Let's begin by importing matplotlib.pyplot and seaborn .

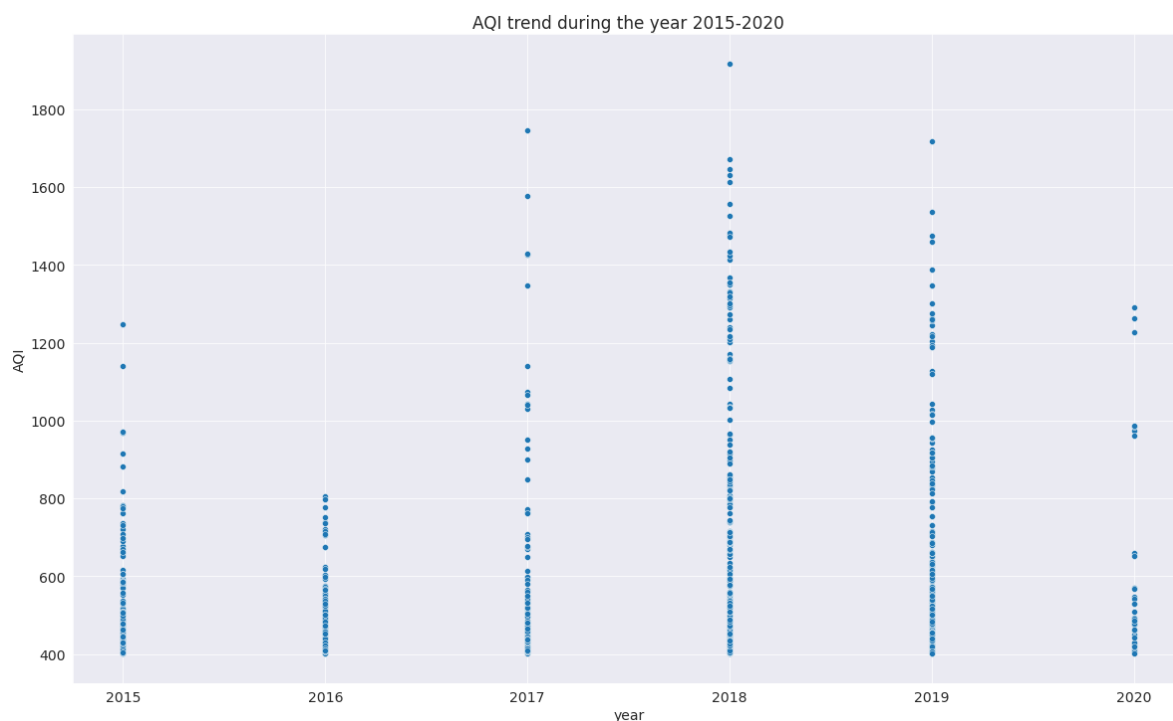
In [24]:

```
1 import seaborn as sns
2 import matplotlib
3 import matplotlib.pyplot as plt
4 %matplotlib inline
5
6 sns.set_style('darkgrid')
7 matplotlib.rcParams['font.size'] = 14
8 matplotlib.rcParams['figure.figsize'] = (20,12)
9 matplotlib.rcParams['figure.facecolor'] = '#00000000'
```

The following graph shows in which year the AQI was highest

In [25]:

```
1 plt.title('AQI trend during the year 2015-2020')
2 sns.scatterplot(x=dfs.year, y=dfs.AQI);
```



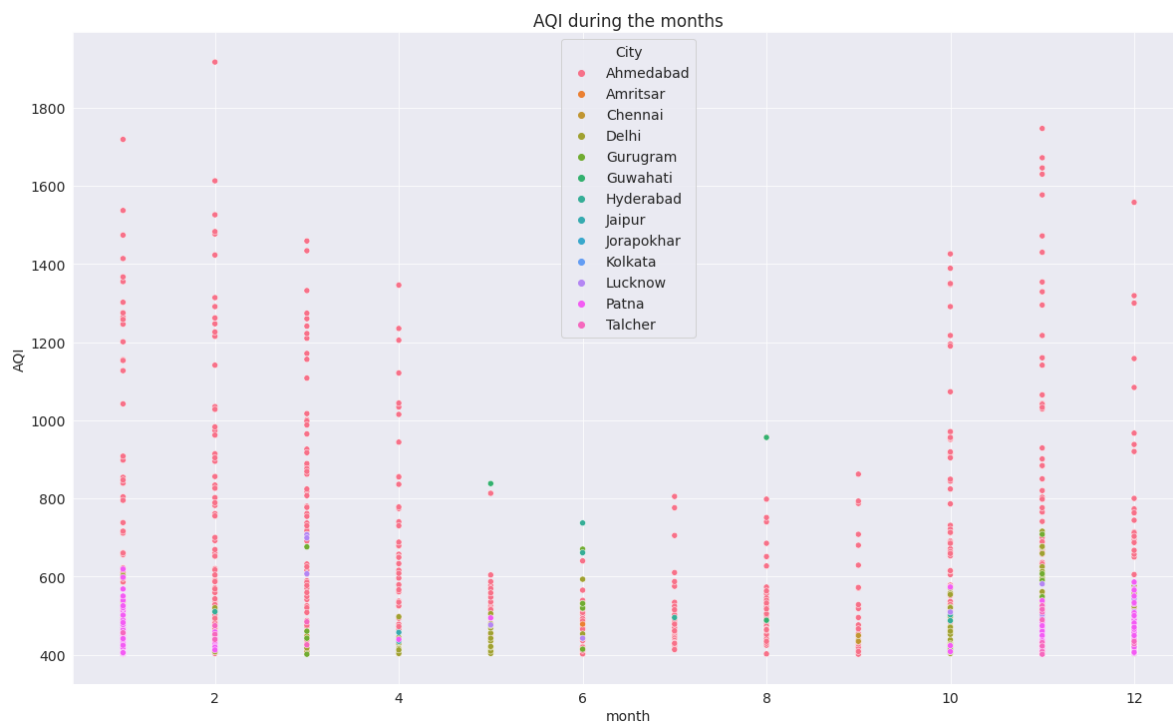
The plot tells that the AQI decreases from 2015 to 2016 and then again started increasing, the AQI was highest during 2018 and its value in some cities was around 2000, we can hypothesize out of this that the industrialisation, factory work maybe very high in this year, it again decreases in 2019 and even more less in 2020. The lower AQI in 2020 was because of lockdown, as factory works, industrial work, vehicles smoke emissions was very less.



The following plot shows the trend of AQI during various months

In [26]:

```
1 sns.scatterplot(x=dfs.month, y=dfs.AQI, hue=df.City);  
2 plt.title('AQI during the months');
```



The above graph shows that the AQI is less during the summers and monsoon, where as the highest AQI was recorded during the winters, this may be due to the fact that the air is more humid during the summers and monsoon and pollutants might have settled down keeping the air clear.

The next plot shows how was the AQI variation in different cities during the year 2018 the year 2018 was selected because that is the year when the air quality was very bad

In [27]:

```
1 year_2018=dfs[dfs.year==2018]
```

In [28]:

```
1 year_2018
```

Out[28]:

	City	Date	PM2.5	NO	NO2	NOx	CO	SO2	O3	AQI	year	mo
1099	Ahmedabad	2018-01-04	81.10	19.42	109.07	67.15	19.42	93.30	65.69	532.0	2018	
1100	Ahmedabad	2018-01-05	73.61	25.96	173.08	103.80	25.96	93.86	36.51	534.0	2018	
1101	Ahmedabad	2018-01-06	75.79	31.55	146.80	95.16	31.55	113.82	41.95	738.0	2018	
1102	Ahmedabad	2018-01-07	77.87	23.19	137.62	84.44	23.19	107.58	50.46	622.0	2018	
1103	Ahmedabad	2018-01-08	77.72	8.91	135.66	72.82	8.91	101.93	48.73	517.0	2018	
...	...	...	...	...	...	...	...	...	...	...	...	...
26374	Talcher	2018-11-27	110.22	21.49	0.32	21.42	1.92	26.37	8.15	525.0	2018	
26375	Talcher	2018-11-28	110.75	51.52	0.25	37.78	2.96	26.82	6.40	489.0	2018	
26377	Talcher	2018-11-30	100.66	23.27	0.10	22.40	2.08	28.86	8.24	509.0	2018	
26378	Talcher	2018-12-01	88.36	24.03	0.09	22.83	1.84	25.13	8.19	428.0	2018	
26379	Talcher	2018-12-02	100.93	25.19	0.02	23.45	1.86	18.29	8.62	434.0	2018	

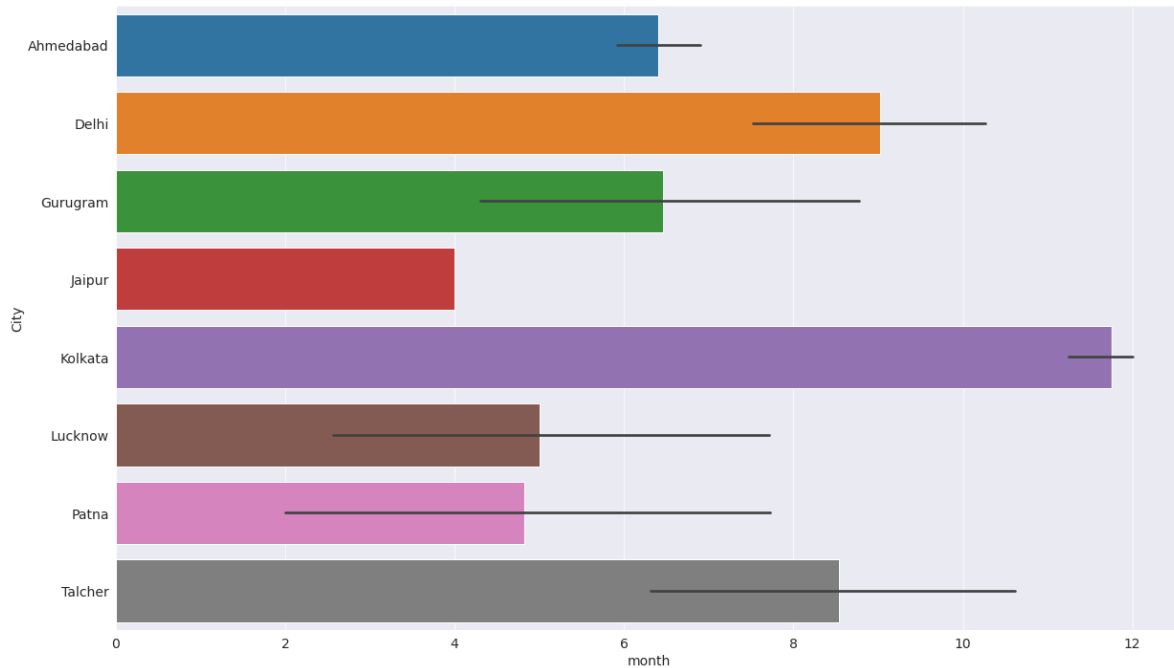
315 rows × 14 columns



In [29]:



```
1 sns.barplot(x=year_2018.month, y=year_2018.City, data=year_2018.AQI);  
2
```



The above plot shows the average was very higher in Kolkata where as the higher deviation was seen in Patna, Lucknow and Gurugram

**The following path comparably tells the AQI trend in the cities showing higher deviation**

In [30]:

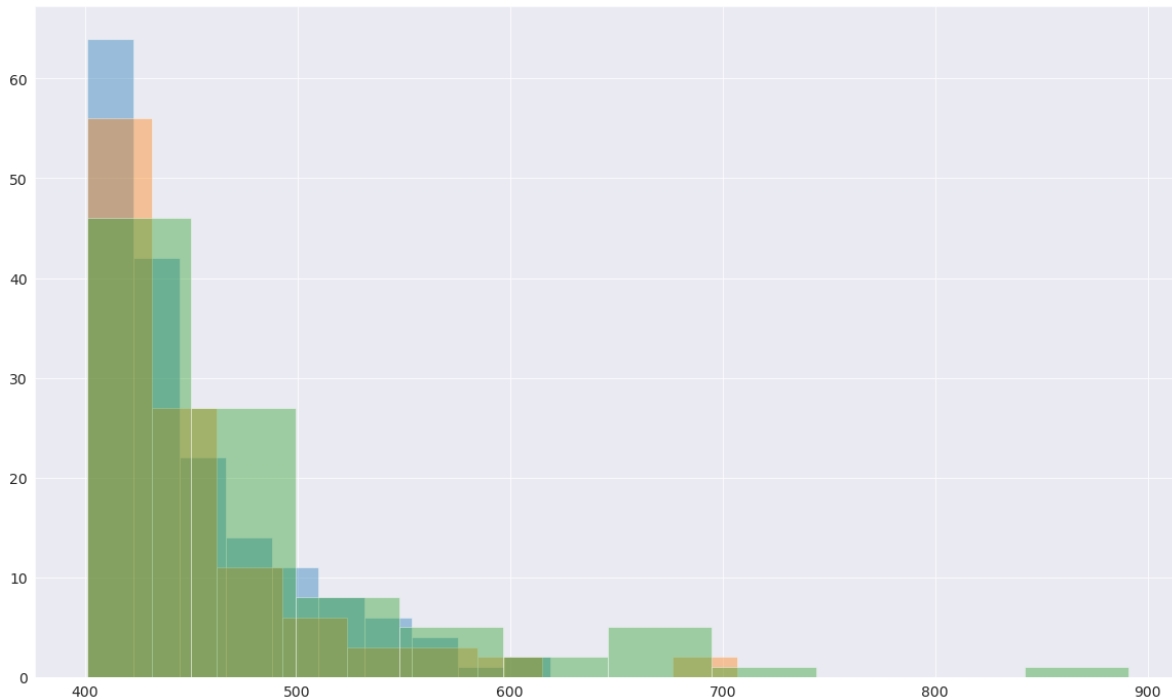


```
1 pt_df=df[df.City=='Patna']  
2 lc_df=df[df.City=='Lucknow']  
3 gr_df=df[df.City=='Gurugram']
```

In [31]:



```
1 plt.hist(pt_df.AQI, alpha=0.4);  
2 plt.hist(lc_df.AQI, alpha=0.4);  
3 plt.hist(gr_df.AQI, alpha=0.4);
```



Let us save and upload our work to Jovian before continuing

In [32]:



```
1 import jovian
```

In [33]:



```
1 jovian.commit()
```

<IPython.core.display.Javascript object>

```
[jovian] Attempting to save notebook..  
[jovian] Updating notebook "vedangi001/aqi-in-india" on https://jovian.ai (https://jovian.ai)  
[jovian] Uploading notebook..  
[jovian] Committed successfully! https://jovian.ai/vedangi001/aqi-in-india  
(https://jovian.ai/vedangi001/aqi-in-india)
```

Out[33]:

```
'https://jovian.ai/vedangi001/aqi-in-india'
```

## Asking and Answering Questions

In this section I have asked some common question which can be arise just by the looking data in its visual format till the time and tried to answer them.

**Q1: What is the variation of AQI during the week in Delhi?**

In [34]:



```
1 city=year_2018[year_2018.City=='Delhi']
```

In [35]:

```
1 city
```

Out[35]:

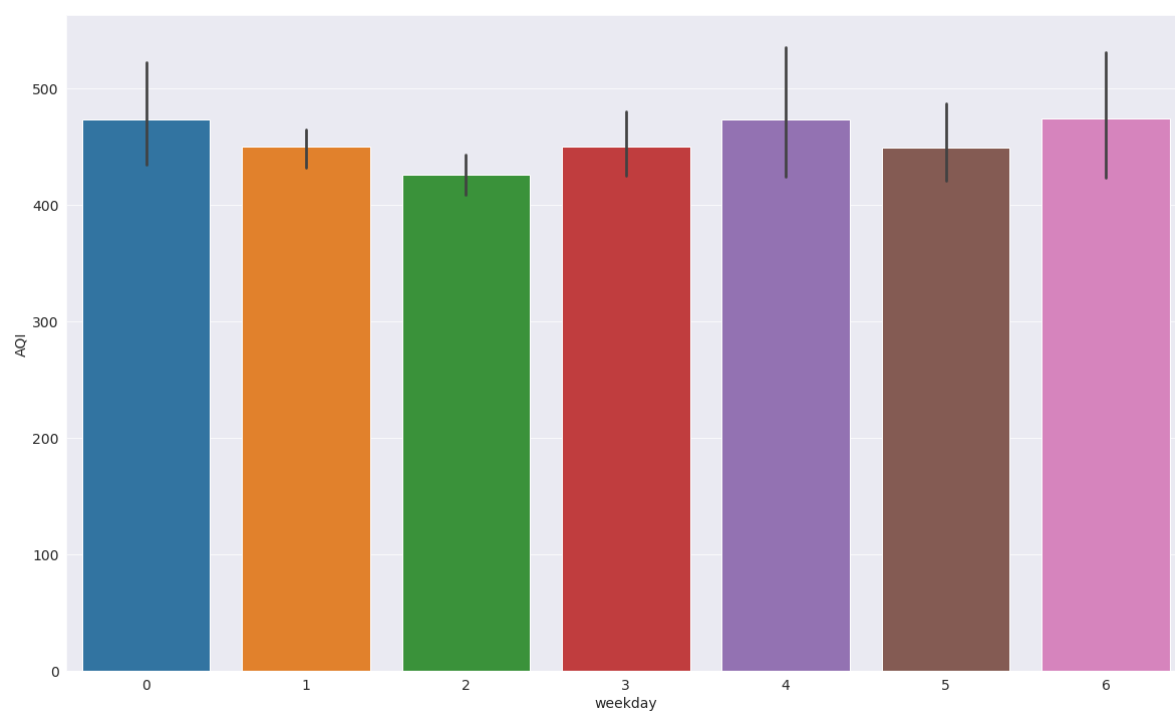
	City	Date	PM2.5	NO	NO2	NOx	CO	SO2	O3	AQI	year	month	c
11325	Delhi	2018-01-01	303.41	62.76	63.92	11.79	1.80	18.58	68.75	462.0	2018	1	
11326	Delhi	2018-01-02	277.48	31.26	53.73	4.26	1.52	12.43	68.68	439.0	2018	1	
11327	Delhi	2018-01-03	217.01	37.98	52.24	2.87	1.29	10.28	54.94	405.0	2018	1	
11330	Delhi	2018-01-06	252.57	65.27	59.12	8.53	1.59	18.15	39.65	403.0	2018	1	
11342	Delhi	2018-01-18	276.25	48.32	55.08	85.75	2.04	21.02	72.89	411.0	2018	1	
11343	Delhi	2018-01-19	253.70	41.26	49.45	81.11	1.82	22.27	73.60	451.0	2018	1	
11435	Delhi	2018-04-21	119.24	9.64	30.99	35.73	0.89	12.08	61.05	444.0	2018	4	
11488	Delhi	2018-06-13	261.77	11.58	25.75	30.62	0.77	6.91	27.45	453.0	2018	6	
11489	Delhi	2018-06-14	225.49	11.75	25.37	29.01	0.72	7.22	43.38	522.0	2018	6	
11490	Delhi	2018-06-15	153.28	12.05	24.61	29.31	0.70	7.31	40.15	593.0	2018	6	
11491	Delhi	2018-06-16	127.32	10.15	25.00	29.76	0.68	7.29	45.76	531.0	2018	6	
11625	Delhi	2018-10-28	205.00	81.11	57.66	100.04	2.45	15.62	52.78	403.0	2018	10	
11627	Delhi	2018-10-30	263.39	98.50	70.46	119.07	2.90	17.62	50.55	456.0	2018	10	
11628	Delhi	2018-10-31	225.17	99.90	68.02	122.87	2.91	18.11	47.05	422.0	2018	10	
11629	Delhi	2018-11-01	236.67	106.68	71.63	133.14	2.97	20.12	45.98	433.0	2018	11	
11633	Delhi	2018-11-05	361.98	26.22	48.25	48.80	1.88	14.63	45.00	424.0	2018	11	
11634	Delhi	2018-11-06	177.88	23.48	43.78	44.84	1.29	15.37	45.01	411.0	2018	11	
11636	Delhi	2018-11-08	420.30	89.10	64.57	106.68	2.45	26.52	38.35	452.0	2018	11	
11637	Delhi	2018-11-09	300.91	89.45	68.25	109.68	2.65	17.41	47.94	487.0	2018	11	
11638	Delhi	2018-11-10	288.86	99.32	79.50	127.11	3.01	18.01	42.55	430.0	2018	11	
11639	Delhi	2018-11-11	256.68	91.92	70.55	113.81	2.89	16.20	51.68	454.0	2018	11	
11640	Delhi	2018-11-12	271.70	75.93	71.88	101.50	2.85	17.40	53.37	437.0	2018	11	

	City	Date	PM2.5	NO	NO2	NOx	CO	SO2	O3	AQI	year	month	c
11641	Delhi	2018-11-13	265.06	85.28	74.72	111.18	2.74	18.72	37.80	454.0	2018	11	
11668	Delhi	2018-12-10	264.63	89.36	66.58	121.45	2.75	16.74	42.51	431.0	2018	12	
11669	Delhi	2018-12-11	274.97	93.61	68.87	124.90	2.78	15.49	41.90	464.0	2018	12	
11670	Delhi	2018-12-12	209.63	56.21	55.10	80.05	1.95	14.64	36.34	443.0	2018	12	
11678	Delhi	2018-12-20	258.10	114.21	68.97	139.08	2.64	17.68	40.50	423.0	2018	12	
11679	Delhi	2018-12-21	252.20	74.98	66.03	103.06	2.22	15.90	38.38	422.0	2018	12	
11680	Delhi	2018-12-22	326.79	85.55	78.71	117.64	2.59	18.47	44.98	468.0	2018	12	
11681	Delhi	2018-12-23	401.58	112.95	77.63	145.67	3.36	18.21	50.65	557.0	2018	12	
11682	Delhi	2018-12-24	378.68	123.73	75.26	152.77	3.36	17.86	43.25	577.0	2018	12	
11683	Delhi	2018-12-25	268.20	45.37	61.20	74.60	1.90	14.14	36.37	475.0	2018	12	
11684	Delhi	2018-12-26	241.00	59.69	65.20	89.51	1.99	15.37	39.09	405.0	2018	12	
11685	Delhi	2018-12-27	290.10	110.91	78.43	143.91	2.87	18.03	40.96	458.0	2018	12	
11686	Delhi	2018-12-28	249.23	39.19	60.59	68.50	1.66	14.80	37.38	413.0	2018	12	
11687	Delhi	2018-12-29	248.12	74.28	64.97	102.00	1.96	16.49	39.29	420.0	2018	12	
11688	Delhi	2018-12-30	336.42	127.84	76.89	157.78	2.94	19.91	42.87	484.0	2018	12	
11689	Delhi	2018-	319.69	105.28	70.51	133.87	2.70	17.38	41.44	509.0	2018	12	

In [36]:



```
1 sns.barplot(x=city.weekday,y=city.AQI);
```

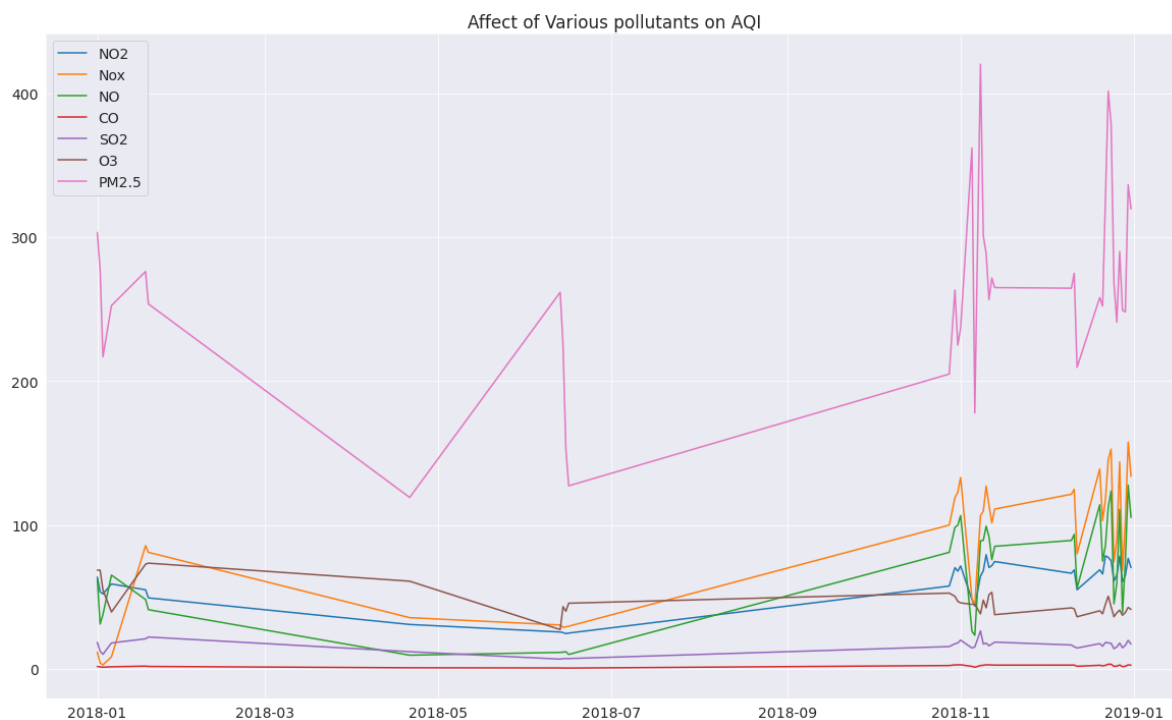


**Q2: Which pollutant has significantly increase during the years in Delhi??**



In [37]:

```
1 plt.plot(city.Date,city.NO2);
2 plt.plot(city.Date,city.NOx);
3 plt.plot(city.Date,city.NO);
4 plt.plot(city.Date,city.CO);
5 plt.plot(city.Date,city.SO2);
6 plt.plot(city.Date,city.O3);
7 pm=city['PM2.5'];
8 plt.plot(city.Date,pm);
9 plt.legend(['NO2','Nox','NO','CO','SO2','O3','PM2.5']);
10 plt.title('Affect of Various pollutants on AQI');
```



The plot tells that the PM2.5 is the pollutant which is majorly affecting the AQI, the Nox is the gas which has significantly increased during past years.

**Q3: When Ahemdabad shows the maximum AQI?**

In [38]:

```
1 ahm=dfs[dfs.City=='Ahmedabad']
```

In [39]:



```
1 ahm
```

Out[39]:

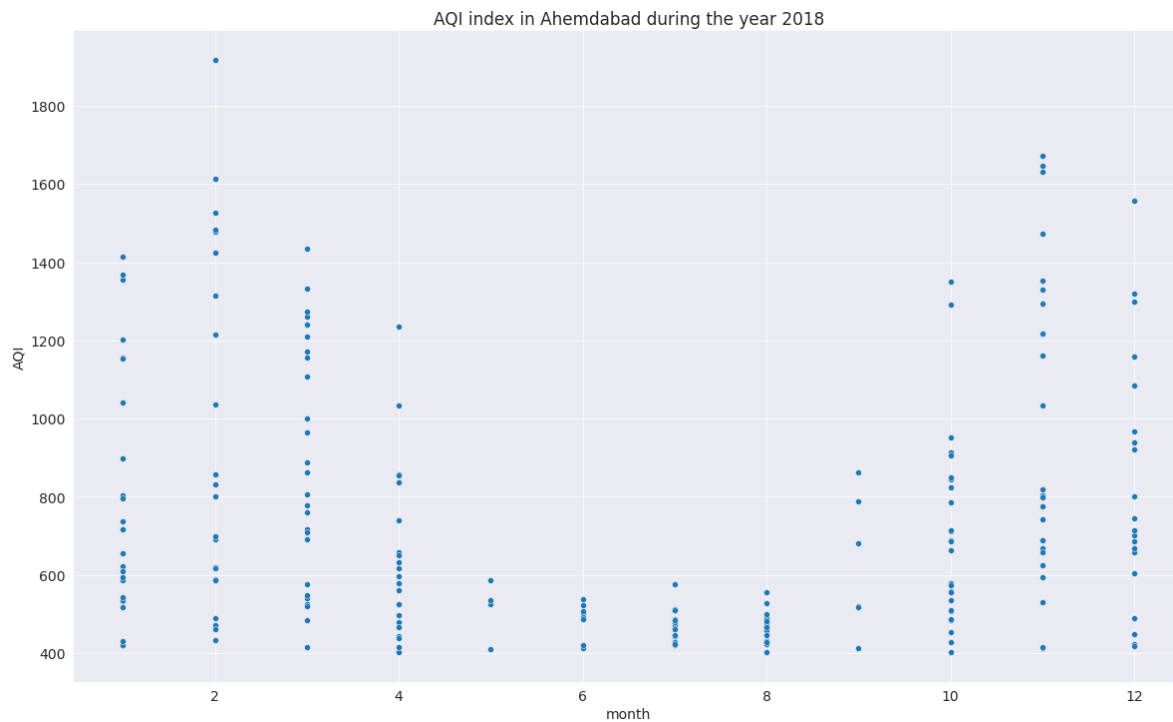
	City	Date	PM2.5	NO	NO2	NOx	CO	SO2	O3	AQI	year	month
30	Ahmedabad	2015-01-31	94.52	24.39	32.66	52.61	24.39	67.39	111.33	514.0	2015	1
31	Ahmedabad	2015-02-01	135.99	43.48	42.08	84.57	43.48	75.23	102.70	782.0	2015	2
32	Ahmedabad	2015-02-02	178.33	54.56	35.31	72.80	54.56	55.04	107.38	914.0	2015	2
33	Ahmedabad	2015-02-03	139.70	30.61	28.40	56.73	30.61	33.79	73.60	660.0	2015	2
43	Ahmedabad	2015-02-13	103.36	18.18	39.56	57.33	18.18	80.43	40.11	510.0	2015	2
...	...	...	...	...	...	...	...	...	...	...	...	...
1886	Ahmedabad	2020-03-01	53.06	22.34	59.64	45.76	22.34	70.80	41.70	475.0	2020	3
1887	Ahmedabad	2020-03-02	56.20	23.13	62.85	47.92	23.13	81.42	34.18	508.0	2020	3
1888	Ahmedabad	2020-03-03	77.58	21.26	60.32	45.29	21.26	119.95	39.83	460.0	2020	3
1903	Ahmedabad	2020-03-18	58.54	15.45	44.39	33.18	15.45	42.86	81.35	487.0	2020	3
1904	Ahmedabad	2020-03-19	101.09	15.07	43.66	31.13	15.07	42.07	44.44	988.0	2020	3

573 rows × 14 columns



In [40]:

```
1 sns.scatterplot(x=ahm.month[ahm.year==2018],y=ahm.AQI);  
2 plt.title('AQI index in Ahemdabad during the year 2018');
```



The Ahemedabad shows the highest AQI during the feb month of the year 2018

**Q4: How was the variation of different pollutants during the various year in Kolkata?**

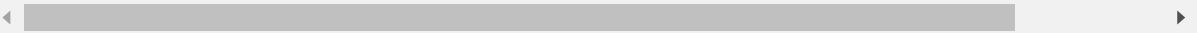
In [41]:



```
1 klk=dfs[dfs.City=='Kolkata']
2 klk
```

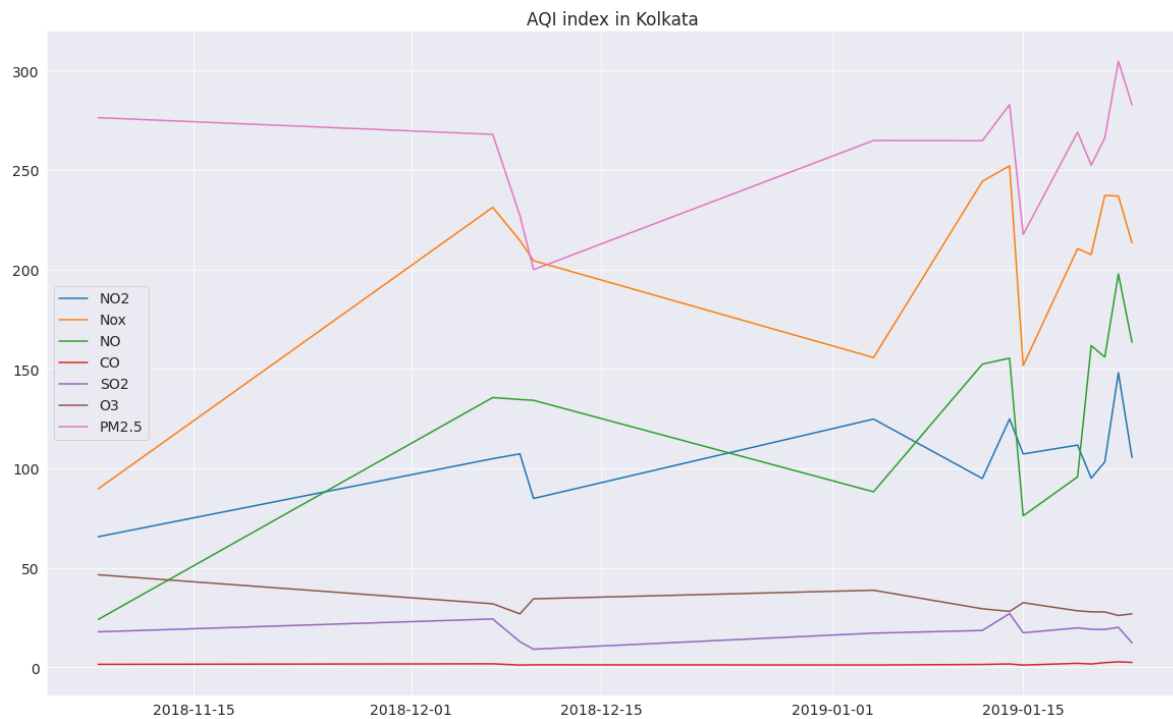
Out[41]:

	City	Date	PM2.5	NO	NO2	NOx	CO	SO2	O3	AQI	year	month
19244	Kolkata	2018-11-08	276.33	24.08	65.65	89.73	1.48	17.82	46.50	431.0	2018	11
19273	Kolkata	2018-12-07	267.93	135.62	104.89	231.23	1.74	24.28	31.93	425.0	2018	12
19275	Kolkata	2018-12-09	227.13	134.63	107.29	214.57	1.11	12.85	26.85	418.0	2018	12
19276	Kolkata	2018-12-10	199.91	134.22	84.86	204.40	1.24	9.08	34.42	407.0	2018	12
19301	Kolkata	2019-01-04	264.84	88.23	124.79	155.73	1.13	17.16	38.73	431.0	2019	1
19309	Kolkata	2019-01-12	264.75	152.42	94.91	244.34	1.42	18.51	29.41	405.0	2019	1
19311	Kolkata	2019-01-14	282.90	155.42	124.83	252.14	1.67	26.99	28.09	408.0	2019	1
19312	Kolkata	2019-01-15	217.59	76.15	107.24	151.68	1.10	17.35	32.47	406.0	2019	1
19316	Kolkata	2019-01-19	269.06	95.75	111.71	210.46	1.99	19.80	28.38	421.0	2019	1
19317	Kolkata	2019-01-20	252.42	161.72	95.09	207.49	1.64	19.09	27.88	407.0	2019	1
19318	Kolkata	2019-01-21	266.10	155.98	103.21	237.22	2.32	19.00	27.81	425.0	2019	1
19319	Kolkata	2019-01-22	304.74	197.73	148.03	236.91	2.75	20.10	26.05	463.0	2019	1
19320	Kolkata	2019-01-23	282.80	163.39	105.50	213.42	2.44	12.30	26.88	475.0	2019	1



In [42]:

```
1 plt.plot(klk.Date,klk.NO2);
2 plt.plot(klk.Date,klk.NOx);
3 plt.plot(klk.Date,klk.NO);
4 plt.plot(klk.Date,klk.CO);
5 plt.plot(klk.Date,klk.SO2);
6 plt.plot(klk.Date,klk.O3);
7 pm=klk['PM2.5'];
8 plt.plot(klk.Date,pm);
9 plt.legend(['NO2','Nox','NO','CO','SO2','O3','PM2.5']);
10 plt.title('AQI index in Kolkata');
```



As in Delhi, the NOx and the PM2.5 are the majorly affecting the AQI

Let us save and upload our work to Jovian before continuing.

In [43]:



```
1 import jovian
```

In [44]:



```
1 jovian.commit()
```

&lt;IPython.core.display.Javascript object&gt;

```
[jovian] Attempting to save notebook..  
[jovian] Updating notebook "vedangi001/aqi-in-india" on https://jovian.ai (https://jovian.ai)  
[jovian] Uploading notebook..  
[jovian] Committed successfully! https://jovian.ai/vedangi001/aqi-in-india  
(https://jovian.ai/vedangi001/aqi-in-india)
```

Out[44]:

```
'https://jovian.ai/vedangi001/aqi-in-india'
```

## Inferences and Conclusion

The inference I have taken out of this project is that the AQI of Ahmedabad was worst during past years, the highest Average was shown in Kolkata, also The PM2.5 and NOx was the major pollutant affecting the AQI.

In [45]:



```
1 import jovian
```

In [46]:



```
1 jovian.commit()
```

&lt;IPython.core.display.Javascript object&gt;

```
[jovian] Attempting to save notebook..  
[jovian] Updating notebook "vedangi001/aqi-in-india" on https://jovian.ai (https://jovian.ai)  
[jovian] Uploading notebook..  
[jovian] Committed successfully! https://jovian.ai/vedangi001/aqi-in-india  
(https://jovian.ai/vedangi001/aqi-in-india)
```

Out[46]:

```
'https://jovian.ai/vedangi001/aqi-in-india'
```

In [47]:



```
1 import jovian
```

In [ ]:



```
1 jovian.commit()
```

<IPython.core.display.Javascript object>

[jovian] Attempting to save notebook..

In [ ]:



```
1
```

In [ ]:



```
1
```