```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import geopandas as gpd
import geoplot
```

```
!pip install geoplot
```

```
Collecting geoplot
  Downloading geoplot-0.5.1-py3-none-any.whl (28 kB)
Requirement already satisfied: matplotlib>=3.1.2 in /usr/local/lib/python3.10/dist-packages (from geoplot) (3.7.1)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (from geoplot) (0.12.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (from geoplot) (1.5.3)
Requirement already satisfied: geopandas>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from geoplot) (0.13.2)
Collecting cartopy (from geoplot)
  Downloading Cartopy-0.22.0-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.8 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.8/11.8 MB 124.3 MB/s eta 0:00:00
Collecting mapclassify>=2.1 (from geoplot)
  Downloading mapclassify-2.6.1-py3-none-any.whl (38 kB)
Collecting contextily>=1.0.0 (from geoplot)
  Downloading contextily-1.4.0-py3-none-any.whl (16 kB)
Requirement already satisfied: geopy in /usr/local/lib/python3.10/dist-packages (from contextily>=1.0.0->geoplot) (2.3.0)
Collecting mercantile (from contextily>=1.0.0->geoplot)
  Downloading mercantile-1.2.1-py3-none-any.whl (14 kB)
Requirement already satisfied: pillow in /usr/local/lib/python3.10/dist-packages (from contextily>=1.0.0->geoplot) (9.4.0)
Collecting rasterio (from contextily>=1.0.0->geoplot)
  Downloading rasterio-1.3.9-cp310-cp310-manylinux2014_x86_64.whl (20.6 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 20.6/20.6 MB 109.4 MB/s eta 0:00:00
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from contextily>=1.0.0->geoplot) (2.31.0)
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from contextily>=1.0.0->geoplot) (1.3.2)
Requirement already satisfied: xyzservices in /usr/local/lib/python3.10/dist-packages (from contextily>=1.0.0->geoplot) (2023.10.1)
Requirement already satisfied: fiona>=1.8.19 in /usr/local/lib/python3.10/dist-packages (from geopandas>=0.9.0->geoplot) (1.9.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geopandas>=0.9.0->geoplot) (23.2)
Requirement already satisfied: pyproj>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from geopandas>=0.9.0->geoplot) (3.6.1)
Requirement already satisfied: shapely>=1.7.1 in /usr/local/lib/python3.10/dist-packages (from geopandas>=0.9.0->geoplot) (2.0.2)
Requirement already satisfied: networkx>=2.7 in /usr/local/lib/python3.10/dist-packages (from mapclassify>=2.1->geoplot) (3.2.1)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.10/dist-packages (from mapclassify>=2.1->geoplot) (1.23.5)
Requirement already satisfied: scikit-learn>=1.0 in /usr/local/lib/python3.10/dist-packages (from mapclassify>=2.1->geoplot) (1.2.2)
Requirement already satisfied: scipy>=1.8 in /usr/local/lib/python3.10/dist-packages (from mapclassify>=2.1->geoplot) (1.11.3)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.2->geoplot) (1.2.0)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.2->geoplot) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.2->geoplot) (4.44.0
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.2->geoplot) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.2->geoplot) (3.1.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib>=3.1.2->geoplot) (2.8
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas->geoplot) (2023.3.post1)
Requirement already satisfied: pyshp>=2.1 in /usr/local/lib/python3.10/dist-packages (from cartopy->geoplot) (2.3.1)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas>=0.9.0->geoplo
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas>=0.9.0->geoplot) (20
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas>=0.9.0->geoplot)
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas>=0.9.0->g
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas>=0.9.0->geoplot)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas>=0.9.0->geoplot) (1.16.0
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas>=0.9.0->geoplot)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=1.0->mapclassify>=
Requirement already satisfied: geographiclib<3,>=1.52 in /usr/local/lib/python3.10/dist-packages (from geopy->contextily>=1.0.0->geop
Collecting affine (from rasterio->contextily>=1.0.0->geoplot)
  Downloading affine-2.4.0-py3-none-any.whl (15 kB)
Collecting snuggs>=1.4.1 (from rasterio->contextily>=1.0.0->geoplot)
  Downloading snuggs-1.4.7-py3-none-any.whl (5.4 kB)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->contextily>=1.0.0-
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->contextily>=1.0.0->geoplot) (3
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests->contextily>=1.0.0->geopl
Installing collected packages: snuggs, mercantile, affine, rasterio, mapclassify, contextily, cartopy, geoplot
Successfully installed affine-2.4.0 cartopy-0.22.0 contextily-1.4.0 geoplot-0.5.1 mapclassify-2.6.1 mercantile-1.2.1 rasterio-1.3.9 s
```

```
df = pd.read_csv("city_day.csv")
```

```
df.head()
```

	City	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene
0	Ahmedabad	2015-01-01	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00

```
df['Date'] = pd.to_datetime(df['Date'],format='%Y-%m-%d') # date parse
df['year'] = df['Date'].dt.year # year
df['year'] = df['year'].fillna(df["year"].min())
df['year'] = df['year'].values.astype(int)
```

```
print(df.dtypes.value_counts())
```

```
float64           13
object             2
datetime64[ns]     1
int64              1
dtype: int64
```

```
def printNullValues(df):
    total = df.isnull().sum().sort_values(ascending = False)
    total = total[df.isnull().sum().sort_values(ascending = False) != 0]
    percent = total / len(df) * 100
    percent = percent[df.isnull().sum().sort_values(ascending = False) != 0]
    concat = pd.concat([total, percent], axis=1, keys=['Total','Percent'])
    print (concat)
    print ( "-------------")
```

```
printNullValues(df)
```

```
              Total    Percent
Xylene      18109  61.322001
PM10        11140  37.723071
NH3         10328  34.973418
Toluene      8041  27.229014
Benzene      5623  19.041008
AQI          4681  15.851139
AQI_Bucket   4681  15.851139
PM2.5        4598  15.570079
NOx          4185  14.171549
O3           4022  13.619586
SO2          3854  13.050692
NO2          3585  12.139785
NO           3582  12.129626
CO           2059   6.972334
-------------
```

```
print(df.columns)
```

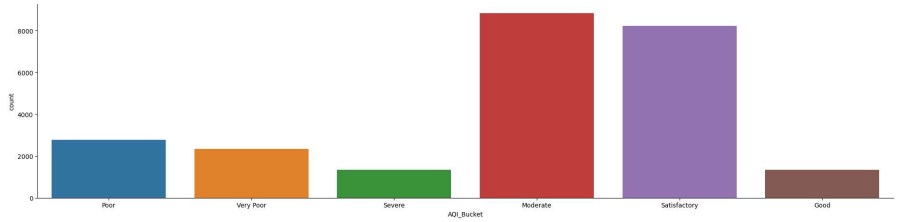
```
Index(['City', 'Date', 'PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'NH3', 'CO', 'SO2',
       'O3', 'Benzene', 'Toluene', 'Xylene', 'AQI', 'AQI_Bucket', 'year'],
      dtype='object')
```

```
df["AQI_Bucket"].value_counts()
```

```
Moderate        8829
Satisfactory    8224
Poor            2781
Very Poor       2337
Good            1341
Severe          1338
Name: AQI_Bucket, dtype: int64
```

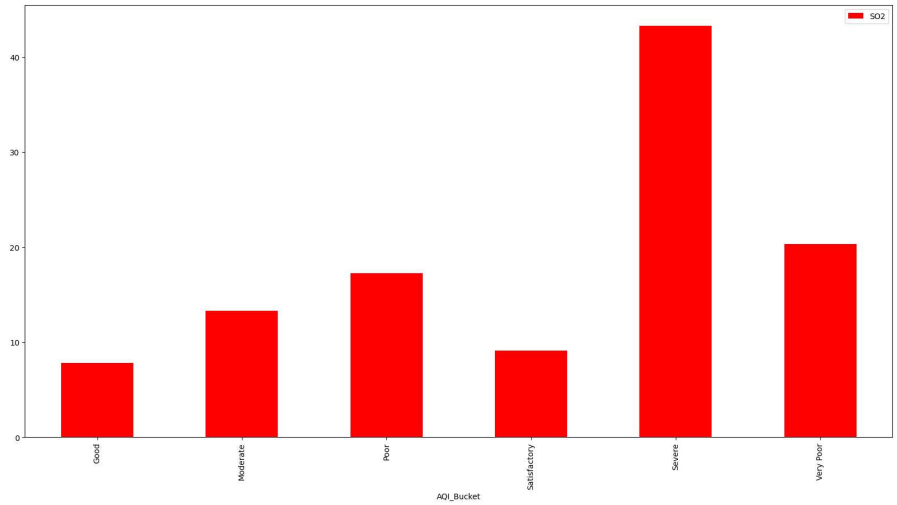
```
sns.catplot(x = "AQI_Bucket", kind = "count",  data = df, height=5, aspect = 4)
```

```
<seaborn.axisgrid.FacetGrid at 0x7f2dd602d750>
```



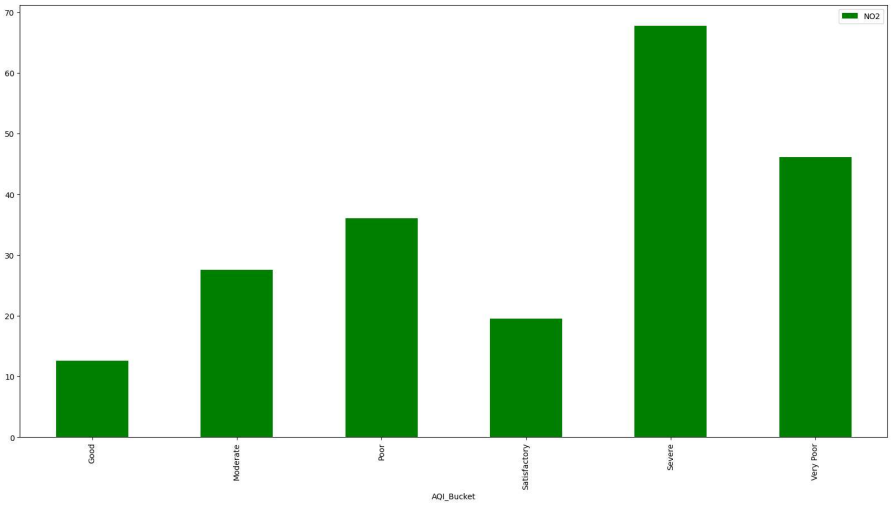

```
grp = df.groupby(["AQI_Bucket"]).mean()["SO2"].to_frame()
grp.plot.bar(figsize = (20,10), color={"red"})
```

```
<ipython-input-21-3cd1b0926c14>:1: FutureWarning: The default value of numeric_only in
  grp = df.groupby(["AQI_Bucket"]).mean()["SO2"].to_frame()
<Axes: xlabel='AQI_Bucket'>
```



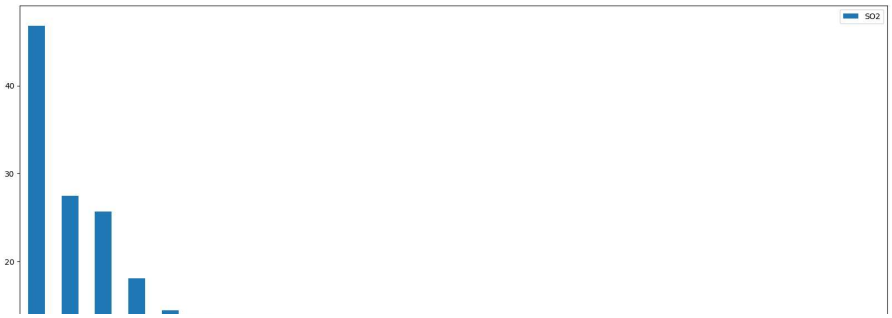

```
grp = df.groupby(["AQI_Bucket"]).mean()["NO2"].to_frame()
grp.plot.bar(figsize = (20,10), color={"green"})
```

```
<ipython-input-22-aa0f70ac1c40>:1: FutureWarning: The default value of numeric_only in
  grp = df.groupby(["AQI_Bucket"]).mean()["NO2"].to_frame()
<Axes: xlabel='AQI_Bucket'>
```



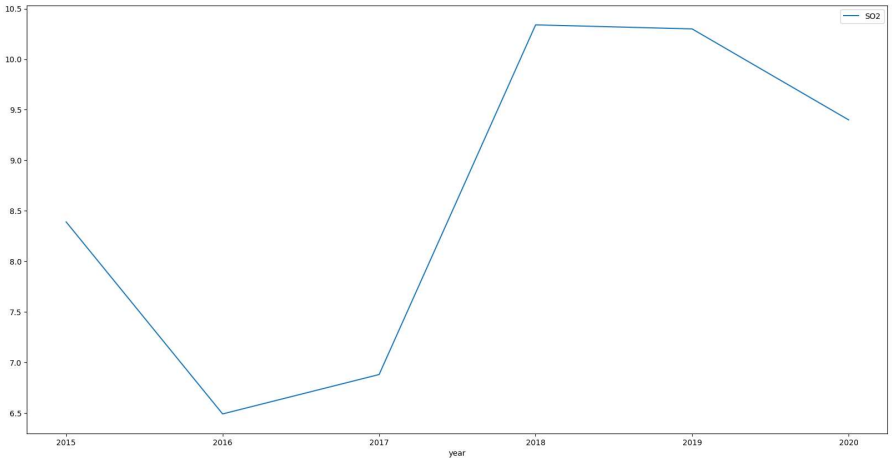

```
df[['SO2', 'City']].groupby(['City']).median().sort_values("SO2", ascending = False).plot.bar(figsize=(20,10))
```

```
<Axes: xlabel='City'>
```



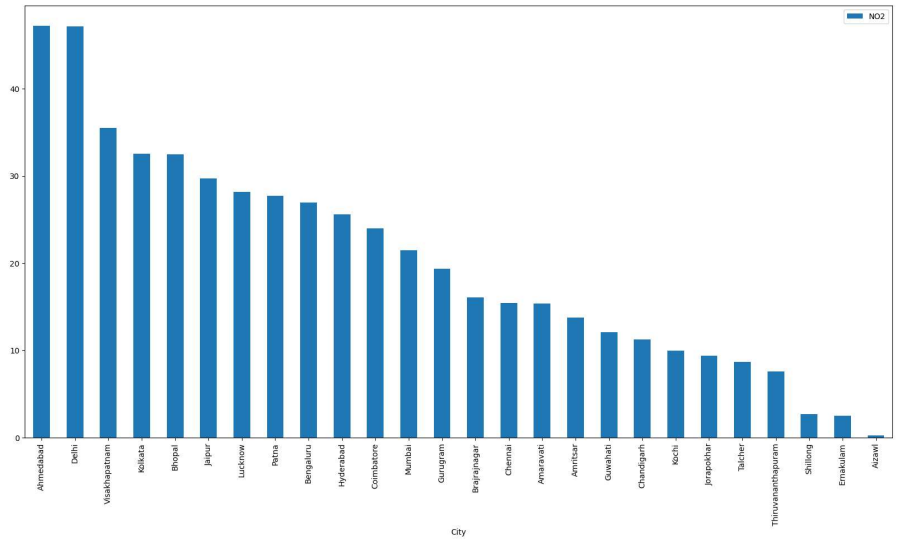

```
df[['SO2','year','City']].groupby(["year"]).median().sort_values(by='year',ascending=False).plot(figsize=(20,10))
```

```
<ipython-input-25-75ff675545a4>:1: FutureWarning: The default value of numeric_only in
  df[['SO2','year','City']].groupby(["year"]).median().sort_values(by='year',ascending=
<Axes: xlabel='year'>
```



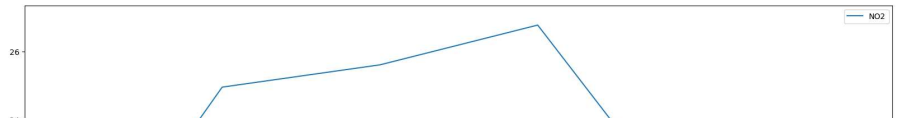

```
df[['NO2', 'City']].groupby(['City']).median().sort_values("NO2", ascending = False).plot.bar(figsize=(20,10))
```

```
<Axes: xlabel='City'>
```



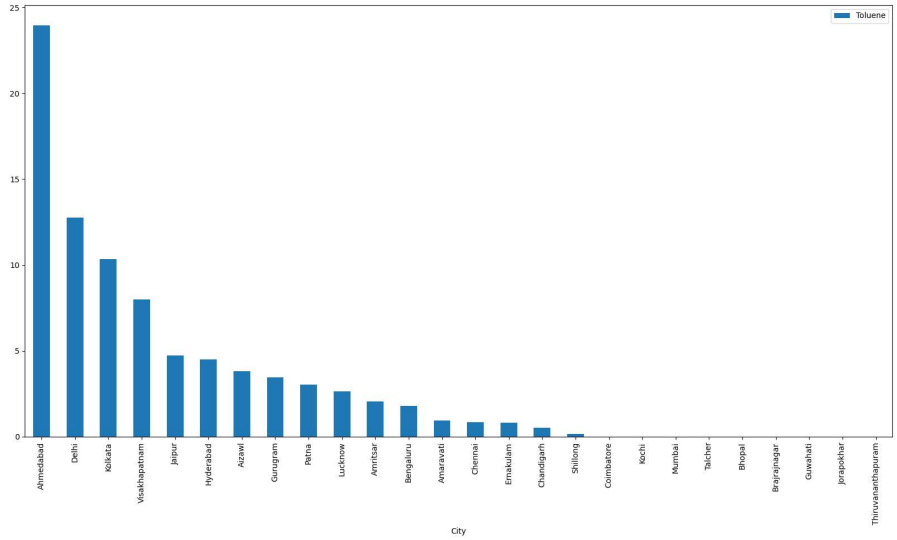

```
df[['NO2','year','City']].groupby(["year"]).median().sort_values(by='year',ascending=False).plot(figsize=(20,10))
```

```
<ipython-input-30-0ed0ee3c885e>:1: FutureWarning: The default value of numeric_only in
  df[['NO2','year','City']].groupby(["year"]).median().sort_values(by='year',ascending=
<Axes: xlabel='year'>
```



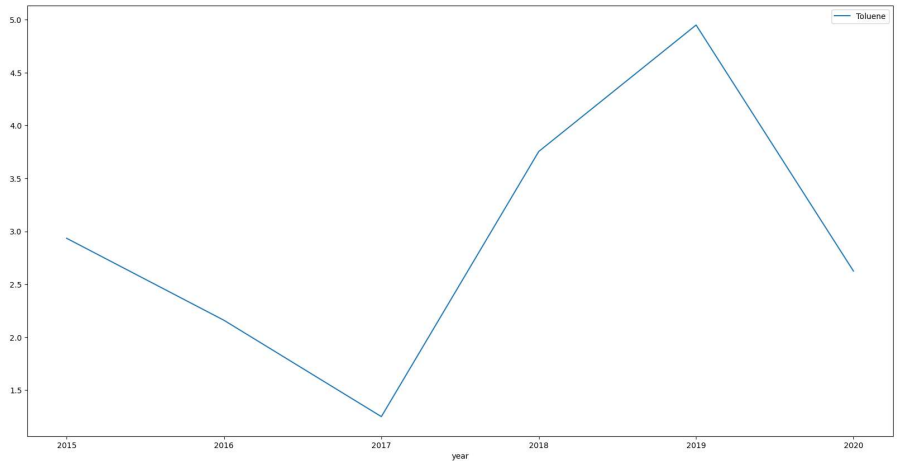

```
df[['Toluene', 'City']].groupby(['City']).median().sort_values("Toluene", ascending = False).plot.bar(figsize=(20,10))
```

```
<Axes: xlabel='City'>
```




```
df[['Toluene','year','City']].groupby(["year"]).median().sort_values(by='year',ascending=False).plot(figsize=(20,10))
```

```
<ipython-input-35-160375f096a0>:1: FutureWarning: The default value of numeric_only in
  df[['Toluene','year','City']].groupby(["year"]).median().sort_values(by='year',ascend
<Axes: xlabel='year'>
```




```
fig, ax = plt.subplots(figsize=(20,10))
sns.heatmap(df.pivot_table('SO2', index='City',columns=['year'],aggfunc='median',margins=True),ax = ax,annot=True, linewidths=.5)
```

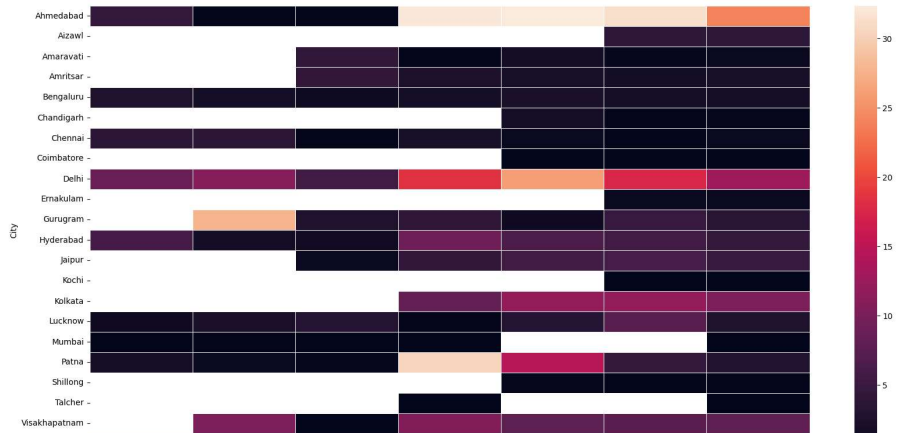
```
<Axes: xlabel='year', ylabel='City'>
```

```
fig, ax = plt.subplots(figsize=(20,10))
sns.heatmap(df.pivot_table('NO2', index='City',columns=['year'],aggfunc='median',margins=True),ax = ax,annot=True, linewidths=.5)
```

```
<Axes: xlabel='year', ylabel='City'>
```

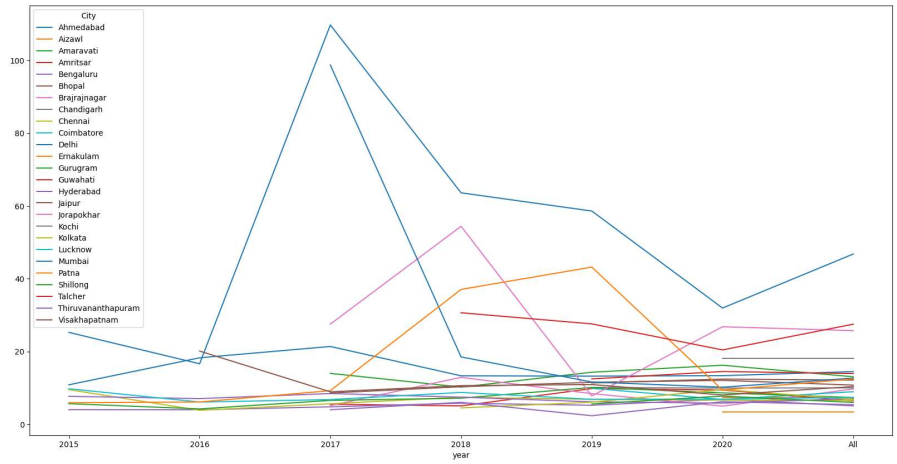

```
fig, ax = plt.subplots(figsize=(20,10))
sns.heatmap(df.pivot_table('Toluene', index='City',columns=['year'],aggfunc='median',margins=True),ax = ax,annot=False, linewidths=.5)
```

```
<Axes: xlabel='year', ylabel='City'>
```



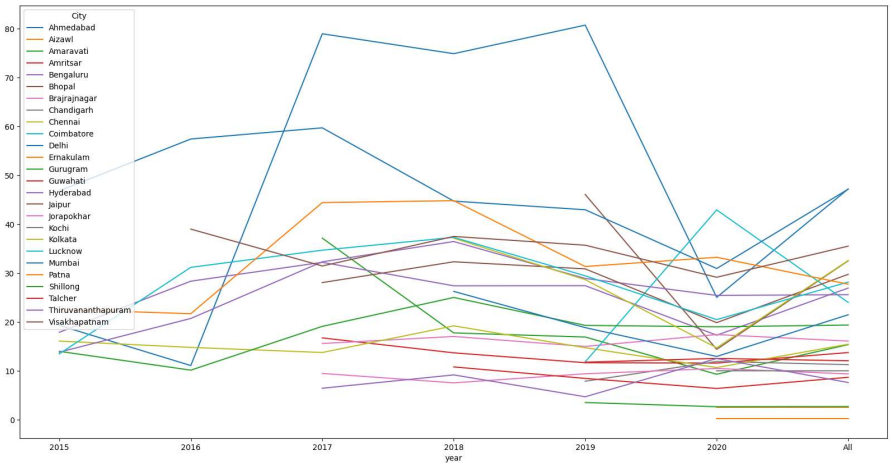

```
temp = df.pivot_table('SO2', index='year',columns=['City'],aggfunc='median',margins=True).reset_index()
temp = temp.drop("All", axis = 1)
temp = temp.set_index("year")
temp.plot(figsize=(20,10))
```

```
<Axes: xlabel='year'>
```




```
temp = df.pivot_table('NO2', index='year',columns=['City'],aggfunc='median',margins=True).reset_index()
temp = temp.drop("All", axis = 1)
temp = temp.set_index("year")
temp.plot(figsize=(20,10))
```

```
<Axes: xlabel='year'>
```




```
temp = df.pivot_table('Toluene', index='year',columns=['City'],aggfunc='median',margins=True).reset_index()
temp = temp.drop("All", axis = 1)
temp = temp.set_index("year")
temp.plot(figsize=(20,10))
```

```
<Axes: xlabel='year'>
```