VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Computer Science and Engineering

J Component report

Programme : M. Tech Integrated CSE with Business Analytics

Course Title : Exploratory Data Analytics

Course Code : CSE3040

Slot : F1

Title : Missing data handling and outlier analysis

Team Members: Keerthana M | 20MIA1082

Varsini SR | 20MIA1087

Sonalika P | 20MIA1089

Faculty : Sweetlin Hemalatha C

Sign:

Date: 29/4/22

MISSING DATA HANDLING AND OUTLIER ANALYSIS

Data preprocessing is the first and foremost step in data analysis. Data preprocessing converts the data in its raw form into a more readable format (graphs, documents, etc.),which can be interpreted and analyzed in the further stages.

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. They also have a real-time monitoring app: https://app.cpcbccr.com/AQI_India/

Why is handling missing data important?

1. Missing data reduces statistical power, which refers to the probability that the test will reject the null hypothesis when it is false.
2. The lost data can cause bias in the estimation of parameters.
3. It can reduce the representativeness of the samples.
4. It may complicate the analysis of the study.

Each of these distortions may threaten the validity of the trials and can lead to invalid conclusions.

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

```
df = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')
```

```
df.head()
```

```
             From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0  01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1  02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2  03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3  04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4  05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN

   NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0        2.36         9.78        NaN          NaN         2.11
NaN
1        2.33         8.21        NaN          NaN         2.86
NaN
```

```
2       11.39        17.28        NaN          NaN         7.73
NaN
3        6.06        12.32        NaN          NaN         2.72
NaN
4        5.58        12.67        NaN          NaN         2.65
NaN
```

```
   Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0          24.63             0.52             2.95
1          20.49             0.13             2.01
2          13.04             0.45             3.52
3          19.42             0.65             3.98
4          25.89             0.60             3.53
```

```
df.shape
```

```
(1885, 13)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1841 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   NH3 (ug/m3)      306 non-null    float64
 8   SO2 (ug/m3)      1830 non-null   float64
 9   CO (mg/m3)       1687 non-null   float64
 10  Ozone (ug/m3)    1833 non-null   float64
 11  Benzene (ug/m3)  1859 non-null   float64
 12  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB
```

```
#no of null values in each column
df.isna().sum()
```

```
From Date             0
To Date               0
PM2.5 (ug/m3)        44
PM10 (ug/m3)       1579
NO (ug/m3)           41
NO2 (ug/m3)          42
NOx (ppb)           287
```

```
NH3 (ug/m3)        1579
SO2 (ug/m3)          55
CO (mg/m3)          198
Ozone (ug/m3)        52
Benzene (ug/m3)      26
Toluene (ug/m3)      26
dtype: int64
```

```
df.mean(numeric_only=True)
```

```
PM2.5 (ug/m3)      30.500435
PM10 (ug/m3)       58.150719
NO (ug/m3)          6.952950
NO2 (ug/m3)        12.557347
NOx (ppb)          16.026608
NH3 (ug/m3)        12.271961
SO2 (ug/m3)         6.551770
CO (mg/m3)          0.827825
Ozone (ug/m3)      27.580562
Benzene (ug/m3)     0.583018
Toluene (ug/m3)     1.940172
dtype: float64
```

```
df.median(numeric_only=True)
```

```
PM2.5 (ug/m3)      27.280
PM10 (ug/m3)       67.250
NO (ug/m3)          5.460
NO2 (ug/m3)        11.280
NOx (ppb)          14.615
NH3 (ug/m3)        14.455
SO2 (ug/m3)         4.880
CO (mg/m3)          0.740
Ozone (ug/m3)      24.680
Benzene (ug/m3)     0.000
Toluene (ug/m3)     0.190
dtype: float64
```

```
df.var(numeric_only=True)
```

```
PM2.5 (ug/m3)      411.678015
PM10 (ug/m3)       672.376420
NO (ug/m3)          31.483505
NO2 (ug/m3)         75.819259
NOx (ppb)           77.394611
NH3 (ug/m3)         30.887305
SO2 (ug/m3)         25.583531
CO (mg/m3)           2.167166
Ozone (ug/m3)      282.994183
Benzene (ug/m3)      4.968048
```

```
Toluene (ug/m3)     19.185895
dtype: float64

df.std(numeric_only=True)

PM2.5 (ug/m3)      20.289850
PM10 (ug/m3)       25.930222
NO (ug/m3)          5.611016
NO2 (ug/m3)         8.707426
NOx (ppb)           8.797421
NH3 (ug/m3)         5.557635
SO2 (ug/m3)         5.058016
CO (mg/m3)          1.472130
Ozone (ug/m3)      16.822431
Benzene (ug/m3)     2.228912
Toluene (ug/m3)     4.380171
dtype: float64

df.cov()

                 PM2.5 (ug/m3)  PM10 (ug/m3)  NO (ug/m3)  NO2 (ug/m3)
\
PM2.5 (ug/m3)       411.678015     30.890050   16.066220    51.241957

PM10 (ug/m3)         30.890050    672.376420  -44.806987   -24.387463

NO (ug/m3)           16.066220    -44.806987   31.483505    22.519439

NO2 (ug/m3)          51.241957    -24.387463   22.519439    75.819259

NOx (ppb)            40.344852    -61.393595   36.589019    55.207391

NH3 (ug/m3)          13.933097    -18.579278   19.546962    22.055663

SO2 (ug/m3)          -0.923446      0.531256   -1.680204    -3.889626

CO (mg/m3)           -1.522403      0.001428   -0.332261    -0.253260

Ozone (ug/m3)        60.713479     38.184933   -0.932329    14.139986

Benzene (ug/m3)       0.625041      0.034594    1.625605     1.354755

Toluene (ug/m3)      10.737620      0.074053    4.639590     9.701272


                 NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO (mg/m3)  \
PM2.5 (ug/m3)    40.344852    13.933097    -0.923446   -1.522403
PM10 (ug/m3)    -61.393595   -18.579278     0.531256    0.001428
NO (ug/m3)       36.589019    19.546962    -1.680204   -0.332261
NO2 (ug/m3)      55.207391    22.055663    -3.889626   -0.253260
```

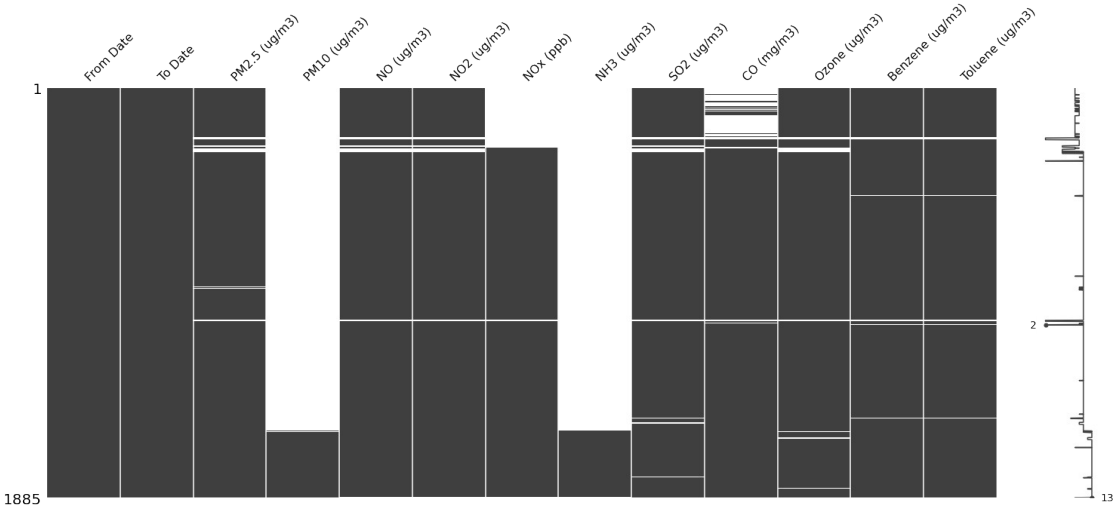
```
NOx (ppb)        77.394611    39.637911    -3.141972    0.158700
NH3 (ug/m3)      39.637911    30.887305     8.791461    0.259349
SO2 (ug/m3)      -3.141972     8.791461    25.583531   -0.472042
CO (mg/m3)        0.158700     0.259349    -0.472042    2.167166
Ozone (ug/m3)     5.818156    23.173855     2.729160   -0.128536
Benzene (ug/m3)   2.326780     0.043199    -1.098227   -0.030333
Toluene (ug/m3)  11.059783     0.036882    -3.746441   -0.073975

                 Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
PM2.5 (ug/m3)        60.713479         0.625041        10.737620
PM10 (ug/m3)         38.184933         0.034594         0.074053
NO (ug/m3)           -0.932329         1.625605         4.639590
NO2 (ug/m3)          14.139986         1.354755         9.701272
NOx (ppb)             5.818156         2.326780        11.059783
NH3 (ug/m3)          23.173855         0.043199         0.036882
SO2 (ug/m3)           2.729160        -1.098227        -3.746441
CO (mg/m3)           -0.128536        -0.030333        -0.073975
Ozone (ug/m3)       282.994183        -0.550818        -0.989454
Benzene (ug/m3)      -0.550818         4.968048         4.877070
Toluene (ug/m3)      -0.989454         4.877070        19.185895

df.corr()

                 PM2.5 (ug/m3)  PM10 (ug/m3)  NO (ug/m3)  NO2 (ug/m3)
\
PM2.5 (ug/m3)         1.000000      0.077045    0.140907     0.290013

PM10 (ug/m3)          0.077045      1.000000   -0.342711    -0.205761

NO (ug/m3)            0.140907     -0.342711    1.000000     0.460980

NO2 (ug/m3)           0.290013     -0.205761    0.460980     1.000000

NOx (ppb)             0.217120     -0.272646    0.714484     0.724070

NH3 (ug/m3)           0.161947     -0.128751    0.702132     0.871728

SO2 (ug/m3)          -0.009008      0.004149   -0.059458    -0.089198

CO (mg/m3)           -0.049465      0.000288   -0.039316    -0.019531

Ozone (ug/m3)         0.178225      0.058454   -0.009927     0.097121

Benzene (ug/m3)       0.013715      0.011016    0.129351     0.069390

Toluene (ug/m3)       0.120031      0.022940    0.188078     0.253154


                 NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO (mg/m3)  \
```

PM2.5 (ug/m3)	0.217120	0.161947	-0.009008	-0.049465
PM10 (ug/m3)	-0.272646	-0.128751	0.004149	0.000288
NO (ug/m3)	0.714484	0.702132	-0.059458	-0.039316
NO2 (ug/m3)	0.724070	0.871728	-0.089198	-0.019531
NOx (ppb)	1.000000	0.825683	-0.068077	0.011973
NH3 (ug/m3)	0.825683	1.000000	0.323777	0.242040
SO2 (ug/m3)	-0.068077	0.323777	1.000000	-0.061407
CO (mg/m3)	0.011973	0.242040	-0.061407	1.000000
Ozone (ug/m3)	0.038354	0.170788	0.032032	-0.008149
Benzene (ug/m3)	0.122707	0.064346	-0.096417	-0.008974
Toluene (ug/m3)	0.278826	0.053442	-0.167630	-0.010942

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
PM2.5 (ug/m3)	0.178225	0.013715	0.120031
PM10 (ug/m3)	0.058454	0.011016	0.022940
NO (ug/m3)	-0.009927	0.129351	0.188078
NO2 (ug/m3)	0.097121	0.069390	0.253154
NOx (ppb)	0.038354	0.122707	0.278826
NH3 (ug/m3)	0.170788	0.064346	0.053442
SO2 (ug/m3)	0.032032	-0.096417	-0.167630
CO (mg/m3)	-0.008149	-0.008974	-0.010942
Ozone (ug/m3)	1.000000	-0.014557	-0.013333
Benzene (ug/m3)	-0.014557	1.000000	0.499545
Toluene (ug/m3)	-0.013333	0.499545	1.000000

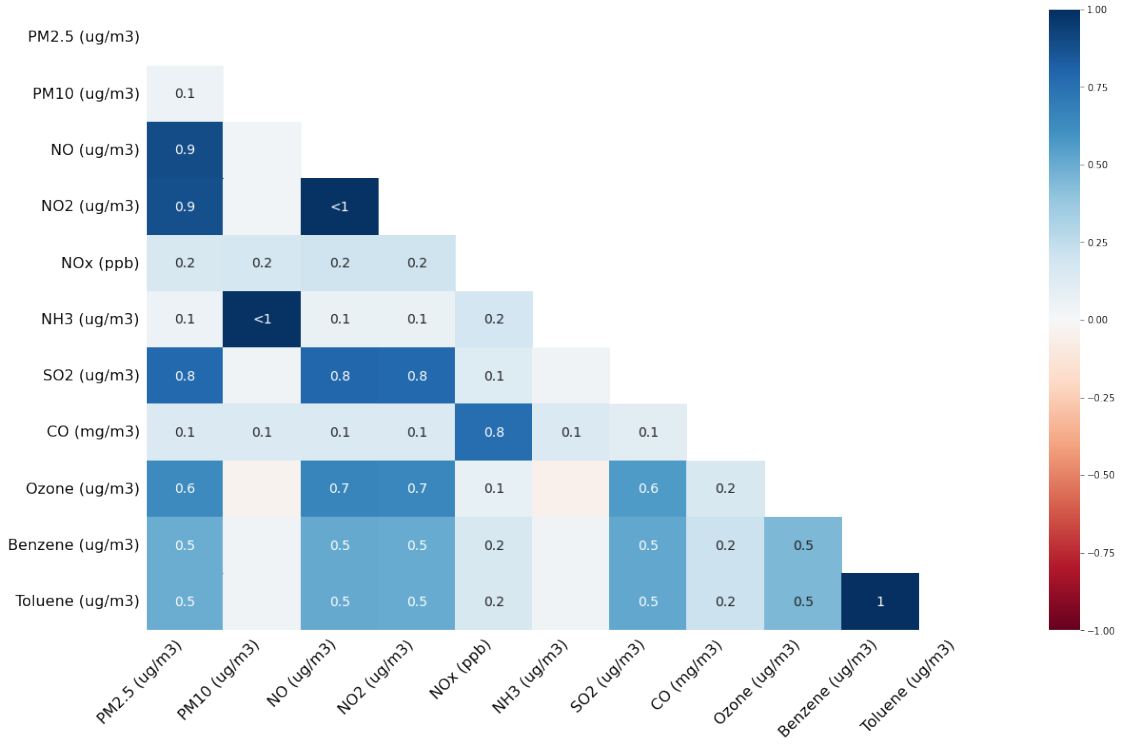
```
#this is plot shows how amuch missing values in the dataset, and its
clearly visible that almost PM10 and NH3 has no data.
import missingno as msno
msno.matrix(df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad886b27d0>
```




```
#helps in visualizing the correlation betwwen all the columns
# Both NO and NO2 has strong positive correlation with PM2.5
msno.heatmap(df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad85c1aa90>
```



MISSING DATA HANDLING

#Listwise deletion

The process of deleting data for any case that has one or more missing values. It is also known as complete case analysis.

```
df1 = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')
```

```
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1841 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   NH3 (ug/m3)      306 non-null    float64
```

```
 8   SO2 (ug/m3)      1830 non-null   float64
 9   CO (mg/m3)       1687 non-null   float64
 10  Ozone (ug/m3)    1833 non-null   float64
 11  Benzene (ug/m3)  1859 non-null   float64
 12  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB
```

```
df1.isnull().sum() #finding the count of null values
```

```
From Date             0
To Date               0
PM2.5 (ug/m3)        44
PM10 (ug/m3)       1579
NO (ug/m3)           41
NO2 (ug/m3)          42
NOx (ppb)           287
NH3 (ug/m3)        1579
SO2 (ug/m3)          55
CO (mg/m3)          198
Ozone (ug/m3)        52
Benzene (ug/m3)      26
Toluene (ug/m3)      26
dtype: int64
```

```
df1.shape
```

```
(1885, 13)
```

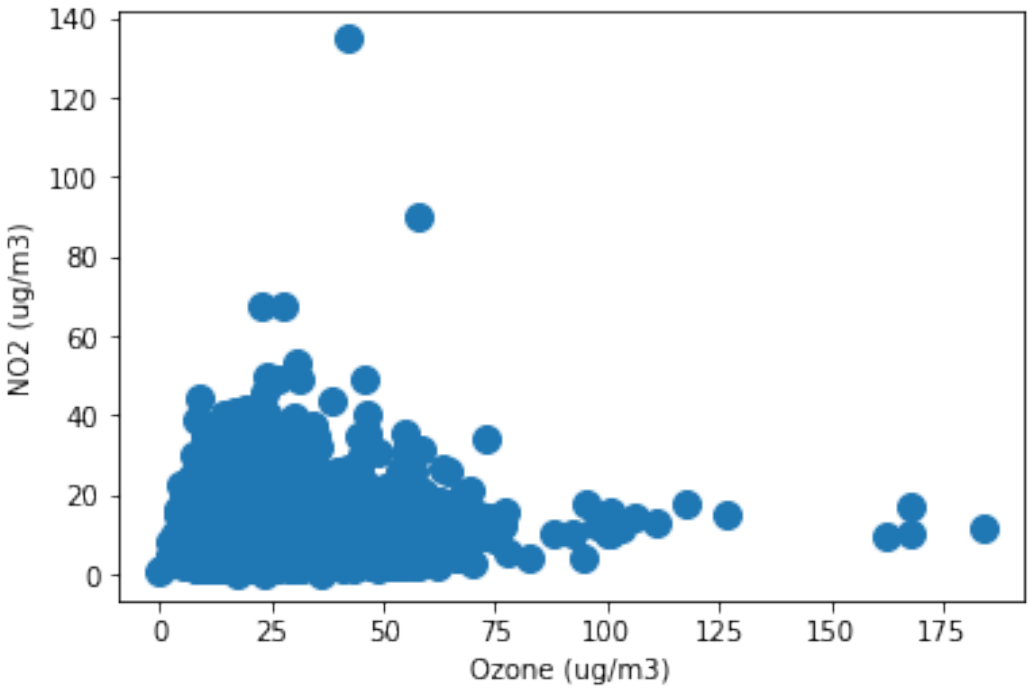
```
#deleting all the Rows which have missing values
df1_new =df1.dropna()
```

```
df1_new.shape
```

```
(286, 13)
```

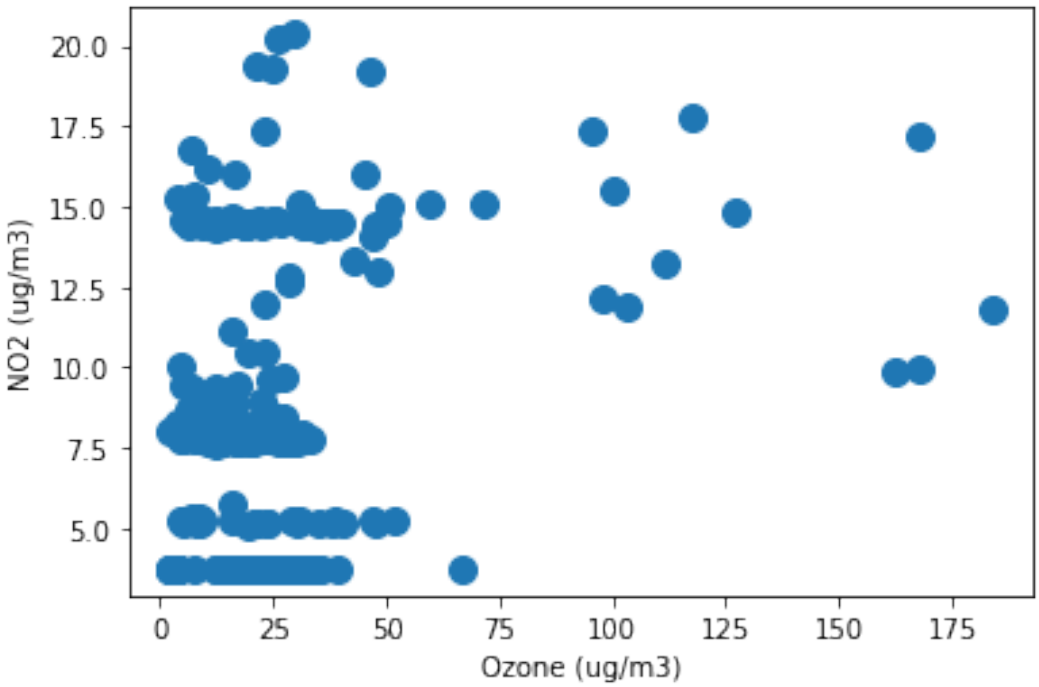
The size of the data has drastically reduced after deleting the missing values.If we delete all the rows where we have NaN value the sample size is 286, which is very less value to evaluate the model.

```
df.plot.scatter(x = "Ozone (ug/m3)", y = 'NO2 (ug/m3)', s = 100);
```



Scatter plot of Ozone and No2 levels before deletion.

```
df1_new.plot.scatter(x = "Ozone (ug/m3)", y = 'NO2 (ug/m3)', s = 100);
```



Listwise deletion scatter plot of Ozone and No2 levels.

Pairwise deletion

A method in which data for a variable pertinent to a specific assessment are included, even if values for the same individual on other variables are missing.Its is also known as availiable case analysis.

```
df2 = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')

#finding percentage of misssing vale in each column
Percent_Missing_Value = df2.isnull().sum()*100/len(df2)
Percent_Missing_Value
```

```
From Date           0.000000
To Date             0.000000
PM2.5 (ug/m3)       2.334218
PM10 (ug/m3)       83.766578
NO (ug/m3)          2.175066
NO2 (ug/m3)         2.228117
NOx (ppb)          15.225464
NH3 (ug/m3)        83.766578
SO2 (ug/m3)         2.917772
CO (mg/m3)         10.503979
Ozone (ug/m3)       2.758621
Benzene (ug/m3)     1.379310
Toluene (ug/m3)     1.379310
dtype: float64
```

If columns have more than half of rows as null then the entire column can be dropped.In PM10 and Nh3 most of the columns almost 84% of the data is missing,so dropping those columns .

```
df2.head()
```

```
             From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0  01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1  02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2  03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3  04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4  05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN

   NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0        2.36         9.78        NaN          NaN         2.11
NaN
```

```
1        2.33         8.21        NaN          NaN         2.86
NaN
2       11.39        17.28        NaN          NaN         7.73
NaN
3        6.06        12.32        NaN          NaN         2.72
NaN
4        5.58        12.67        NaN          NaN         2.65
NaN
```

```
   Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0          24.63             0.52             2.95
1          20.49             0.13             2.01
2          13.04             0.45             3.52
3          19.42             0.65             3.98
4          25.89             0.60             3.53
```

```
#droping PM10 column
df_2 = df2.drop("PM10 (ug/m3)",axis=1)

#droping Nh3 column
df_2 = df2.drop("NH3 (ug/m3)",axis=1)

df_2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1841 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   SO2 (ug/m3)      1830 non-null   float64
 8   CO (mg/m3)       1687 non-null   float64
 9   Ozone (ug/m3)    1833 non-null   float64
 10  Benzene (ug/m3)  1859 non-null   float64
 11  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(10), object(2)
memory usage: 176.8+ KB
```

To find out the relation between Ozone and No2,deleting the missing values pertinent to the columns.

```
#deleting the missing values from ozone
df2.dropna(subset=['Ozone (ug/m3)'],how='any',inplace=True)
df2['Ozone (ug/m3)'].isnull().sum()
```

```
0
```

```
#deleting the missing values from Nh3
df2.dropna(subset=['NO2 (ug/m3)'],how='any',inplace=True)
df2['NO2 (ug/m3)'].isnull().sum()
```

```
0
```

```
df.shape #intial size of the data
```

```
(1885, 13)
```

```
df2.shape #size after deletion
```

```
(1822, 13)
```

```
df.describe()
```

```
       PM2.5 (ug/m3)  PM10 (ug/m3)   NO (ug/m3)  NO2 (ug/m3)    NOx
(ppb)  \
count    1841.000000    306.000000  1844.000000  1843.000000
1598.000000
mean       30.500435     58.150719     6.952950    12.557347
16.026608
std        20.289850     25.930222     5.611016     8.707426
8.797421
min         0.410000     21.600000     0.010000     0.020000
0.000000
25%        16.540000     36.850000     3.295000     6.510000
9.860000
50%        27.280000     67.250000     5.460000    11.280000
14.615000
75%        39.530000     69.415000     9.400000    16.470000
20.872500
max       278.970000    371.610000    98.620000   134.760000
106.740000

       NH3 (ug/m3)  SO2 (ug/m3)   CO (mg/m3)  Ozone (ug/m3)  Benzene
(ug/m3)  \
count   306.000000  1830.000000  1687.000000    1833.000000
1859.000000
mean     12.271961     6.551770     0.827825      27.580562
0.583018
std       5.557635     5.058016     1.472130      16.822431
2.228912
min       5.360000     0.090000     0.000000       0.100000
0.000000
25%       5.360000     3.890000     0.600000      16.770000
0.000000
50%      14.455000     4.880000     0.740000      24.680000
0.000000
75%      16.675000     7.205000     0.910000      34.770000
```

```
0.365000
max      33.680000    37.180000    48.020000     183.990000
46.230000

       Toluene (ug/m3)
count      1859.000000
mean          1.940172
std           4.380171
min           0.000000
25%           0.000000
50%           0.190000
75%           2.755000
max         121.150000
```

Computing the statistical variables after deletion

```
df['NO2 (ug/m3)'].describe()
```

```
count    1843.000000
mean       12.557347
std         8.707426
min         0.020000
25%         6.510000
50%        11.280000
75%        16.470000
max       134.760000
Name: NO2 (ug/m3), dtype: float64
```

```
df2['NO2 (ug/m3)'].describe()
```

```
count    1822.000000
mean       12.634682
std         8.704530
min         0.020000
25%         6.672500
50%        11.360000
75%        16.527500
max       134.760000
Name: NO2 (ug/m3), dtype: float64
```

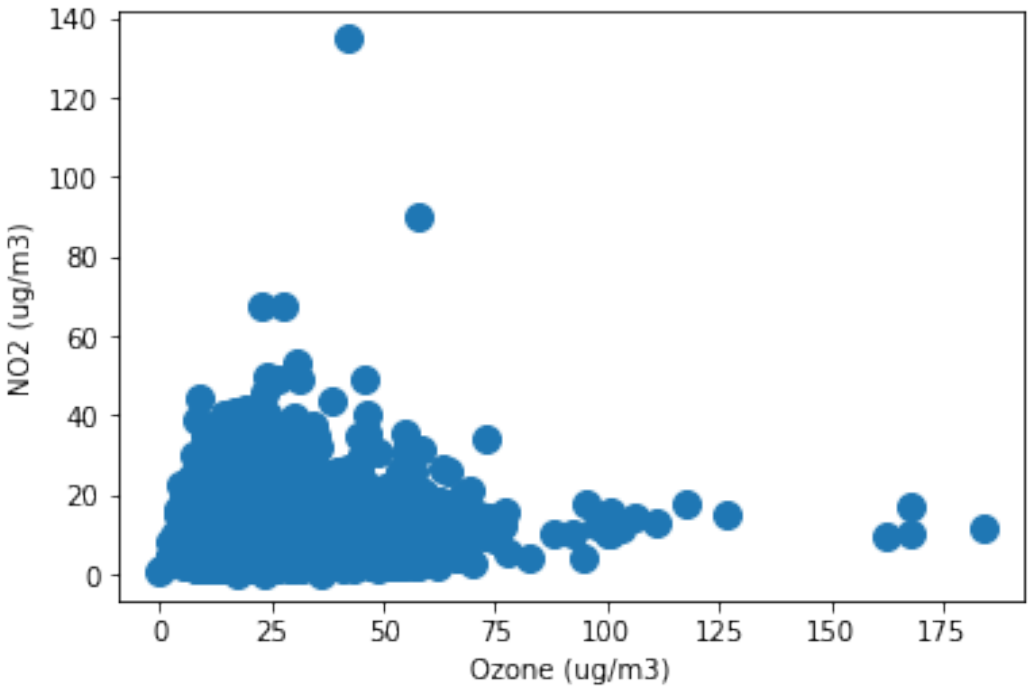
```
df['Ozone (ug/m3)'].describe()
```

```
count    1833.000000
mean       27.580562
std        16.822431
min         0.100000
25%        16.770000
50%        24.680000
75%        34.770000
max       183.990000
Name: Ozone (ug/m3), dtype: float64
```

```
df2['Ozone (ug/m3)'].describe()
```

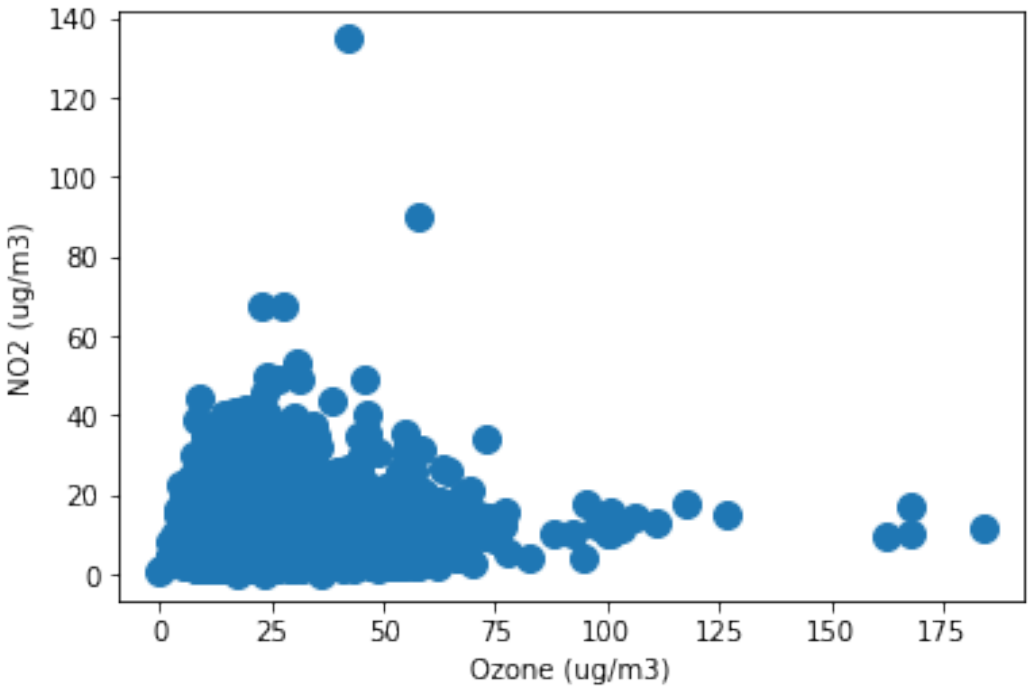
```
count    1822.000000
mean       27.633079
std        16.725983
min         0.100000
25%        16.832500
50%        24.760000
75%        34.792500
max       183.990000
Name: Ozone (ug/m3), dtype: float64
```

```
df.plot.scatter(x = "Ozone (ug/m3)", y = 'NO2 (ug/m3)', s = 100);
```



Scatter plot of Ozone and No2 levels before deletion.

```
df2.plot.scatter(x = "Ozone (ug/m3)", y = 'NO2 (ug/m3)', s = 100);
```



Pairwise deletion scatter plot of Ozone and No2 levels. After deletion this resembles similar to the original dataset whereas in listwise the data is more dispersed and scattered.

#Mean imputation

Mean imputation is a method in which the missing values are replaced with the mean value of the entire feature column.

```
df3 = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')

#all the missing values in the columns are replaced withe the mean
values of the same column
x=['PM2.5 (ug/m3)','PM10 (ug/m3)','NO (ug/m3)','NO2 (ug/m3)','NOx
(ppb)','NH3 (ug/m3)','SO2 (ug/m3)','CO (mg/m3)','Ozone
(ug/m3)','Benzene (ug/m3)','Toluene (ug/m3)']
df3[x] = df3[x].fillna(df3[x].mean())

df3.head(100)
```

```
              From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0   01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
58.150719
1   02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
58.150719
2   03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
58.150719
3   04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
58.150719
```

```
4   05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
58.150719
..                  ...                  ...            ...
...
95  06-Apr-2017 - 00:00  07-Apr-2017 - 00:00          42.08
58.150719
96  07-Apr-2017 - 00:00  08-Apr-2017 - 00:00          37.94
58.150719
97  08-Apr-2017 - 00:00  09-Apr-2017 - 00:00          35.41
58.150719
98  09-Apr-2017 - 00:00  10-Apr-2017 - 00:00          34.46
58.150719
99  10-Apr-2017 - 00:00  11-Apr-2017 - 00:00          40.33
58.150719

    NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0         2.36         9.78  16.026608    12.271961         2.11
0.827825
1         2.33         8.21  16.026608    12.271961         2.86
0.827825
2        11.39        17.28  16.026608    12.271961         7.73
0.827825
3         6.06        12.32  16.026608    12.271961         2.72
0.827825
4         5.58        12.67  16.026608    12.271961         2.65
0.827825
..         ...          ...        ...          ...          ...
...
95        4.36         7.03  16.026608    12.271961         3.16
0.827825
96        4.06         8.86  16.026608    12.271961         3.73
1.230000
97        4.75        13.42  16.026608    12.271961         3.85
1.270000
98        3.42         9.20  16.026608    12.271961         3.63
1.150000
99        4.31         9.92  16.026608    12.271961         3.93
1.240000

    Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0           24.63             0.52             2.95
1           20.49             0.13             2.01
2           13.04             0.45             3.52
3           19.42             0.65             3.98
4           25.89             0.60             3.53
..            ...              ...              ...
95          25.22             1.41             1.78
96          26.72             1.40             1.77
97          26.48             1.41             1.79
```

```
98          26.56             1.40             1.78
99          31.01             1.45             1.83

[100 rows x 13 columns]
```

```
df.describe()
```

```
       PM2.5 (ug/m3)  PM10 (ug/m3)   NO (ug/m3)  NO2 (ug/m3)    NOx
(ppb)  \
count    1841.000000    306.000000  1844.000000  1843.000000
1598.000000
mean       30.500435     58.150719     6.952950    12.557347
16.026608
std        20.289850     25.930222     5.611016     8.707426
8.797421
min         0.410000     21.600000     0.010000     0.020000
0.000000
25%        16.540000     36.850000     3.295000     6.510000
9.860000
50%        27.280000     67.250000     5.460000    11.280000
14.615000
75%        39.530000     69.415000     9.400000    16.470000
20.872500
max       278.970000    371.610000    98.620000   134.760000
106.740000

       NH3 (ug/m3)  SO2 (ug/m3)   CO (mg/m3)  Ozone (ug/m3)  Benzene
(ug/m3)  \
count   306.000000  1830.000000  1687.000000    1833.000000
1859.000000
mean     12.271961     6.551770     0.827825      27.580562
0.583018
std       5.557635     5.058016     1.472130      16.822431
2.228912
min       5.360000     0.090000     0.000000       0.100000
0.000000
25%       5.360000     3.890000     0.600000      16.770000
0.000000
50%      14.455000     4.880000     0.740000      24.680000
0.000000
75%      16.675000     7.205000     0.910000      34.770000
0.365000
max      33.680000    37.180000    48.020000     183.990000
46.230000

       Toluene (ug/m3)
count      1859.000000
mean          1.940172
std           4.380171
min           0.000000
```

```
25%           0.000000
50%           0.190000
75%           2.755000
max         121.150000
```

```
df3.mean() #finding the mean after imputation
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError.  Select only valid columns before calling the
reduction.
  """Entry point for launching an IPython kernel.
```

```
PM2.5 (ug/m3)      30.500435
PM10 (ug/m3)       58.150719
NO (ug/m3)          6.952950
NO2 (ug/m3)        12.557347
NOx (ppb)          16.026608
NH3 (ug/m3)        12.271961
SO2 (ug/m3)         6.551770
CO (mg/m3)          0.827825
Ozone (ug/m3)      27.580562
Benzene (ug/m3)     0.583018
Toluene (ug/m3)     1.940172
dtype: float64
```

```
df3.median() #finding the median after imputation
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError.  Select only valid columns before calling the
reduction.
  """Entry point for launching an IPython kernel.
```

```
PM2.5 (ug/m3)      27.590000
PM10 (ug/m3)       58.150719
NO (ug/m3)          5.580000
NO2 (ug/m3)        11.510000
NOx (ppb)          16.026608
NH3 (ug/m3)        12.271961
SO2 (ug/m3)         4.950000
CO (mg/m3)          0.770000
Ozone (ug/m3)      25.150000
Benzene (ug/m3)     0.000000
Toluene (ug/m3)     0.240000
dtype: float64
```

```
df3.std() #finding the standard deviation after imputation
```

```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:1:
FutureWarning: Dropping of nuisance columns in DataFrame reductions
(with 'numeric_only=None') is deprecated; in a future version this
will raise TypeError.  Select only valid columns before calling the
reduction.
  """Entry point for launching an IPython kernel.
```

```
PM2.5 (ug/m3)      20.051520
PM10 (ug/m3)       10.433156
NO (ug/m3)          5.549627
NO2 (ug/m3)         8.609821
NOx (ppb)           8.099671
NH3 (ug/m3)         2.236143
SO2 (ug/m3)         4.983640
CO (mg/m3)          1.392626
Ozone (ug/m3)      16.588650
Benzene (ug/m3)     2.213479
Toluene (ug/m3)     4.349842
dtype: float64
```

```
df3.cov() #finding the covariance after imputation
```

```
                 PM2.5 (ug/m3)  PM10 (ug/m3)  NO (ug/m3)  NO2 (ug/m3)
\
PM2.5 (ug/m3)       402.063454      4.790271   15.665399    49.936239

PM10 (ug/m3)          4.790271    108.850747   -7.162718    -3.856727

NO (ug/m3)           15.665399     -7.162718   30.798354    22.017413

NO2 (ug/m3)          49.936239     -3.856727   22.017413    74.129021

NOx (ppb)            33.779861     -9.804390   30.740263    46.359606

NH3 (ug/m3)           2.281140     -2.979065    3.164450     3.570582

SO2 (ug/m3)          -0.894573      0.284655   -1.628816    -3.769204

CO (mg/m3)           -1.341713      0.000231   -0.291711    -0.226026

Ozone (ug/m3)        58.637250      5.934287   -0.898932    13.671079

Benzene (ug/m3)       0.607667      0.004987    1.582438     1.318097

Toluene (ug/m3)      10.436053      0.009948    4.516390     9.438699


                 NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO (mg/m3)  \
PM2.5 (ug/m3)    33.779861     2.281140    -0.894573   -1.341713
PM10 (ug/m3)     -9.804390    -2.979065     0.284655    0.000231
```

```
NO (ug/m3)       30.740263     3.164450    -1.628816   -0.291711
NO2 (ug/m3)      46.359606     3.570582    -3.769204   -0.226026
NOx (ppb)        65.604667     6.416965    -2.588604    0.133991
NH3 (ug/m3)       6.416965     5.000333     1.378188    0.041986
SO2 (ug/m3)      -2.588604     1.378188    24.836665   -0.409550
CO (mg/m3)        0.133991     0.041986    -0.409550    1.939407
Ozone (ug/m3)     4.819938     3.464105     2.621455   -0.107126
Benzene (ug/m3)   1.960402     0.004861    -1.062622   -0.026996
Toluene (ug/m3)   9.320662    -0.001140    -3.624998   -0.065842

                 Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
PM2.5 (ug/m3)        58.637250         0.607667        10.436053
PM10 (ug/m3)          5.934287         0.004987         0.009948
NO (ug/m3)           -0.898932         1.582438         4.516390
NO2 (ug/m3)          13.671079         1.318097         9.438699
NOx (ppb)             4.819938         1.960402         9.320662
NH3 (ug/m3)           3.464105         0.004861        -0.001140
SO2 (ug/m3)           2.621455        -1.062622        -3.624998
CO (mg/m3)           -0.107126        -0.026996        -0.065842
Ozone (ug/m3)       275.183304        -0.532692        -0.956679
Benzene (ug/m3)      -0.532692         4.899487         4.809764
Toluene (ug/m3)      -0.956679         4.809764        18.921121
```

```
df3.corr() #finding the correlation after imputation
```

```
                 PM2.5 (ug/m3)  PM10 (ug/m3)  NO (ug/m3)  NO2 (ug/m3)
\
PM2.5 (ug/m3)         1.000000      0.022898    0.140777     0.289251

PM10 (ug/m3)          0.022898      1.000000   -0.123708    -0.042935

NO (ug/m3)            0.140777     -0.123708    1.000000     0.460796

NO2 (ug/m3)           0.289251     -0.042935    0.460796     1.000000

NOx (ppb)             0.207990     -0.116021    0.683875     0.664780

NH3 (ug/m3)           0.050875     -0.127692    0.254997     0.185458

SO2 (ug/m3)          -0.008952      0.005475   -0.058893    -0.087843

CO (mg/m3)           -0.048048      0.000016   -0.037745    -0.018851

Ozone (ug/m3)         0.176285      0.034288   -0.009765     0.095719

Benzene (ug/m3)       0.013691      0.000216    0.128821     0.069164

Toluene (ug/m3)       0.119651      0.000219    0.187092     0.252026
```

```
                  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO (mg/m3)  \
PM2.5 (ug/m3)      0.207990     0.050875    -0.008952   -0.048048
PM10 (ug/m3)      -0.116021    -0.127692     0.005475    0.000016
NO (ug/m3)         0.683875     0.254997    -0.058893   -0.037745
NO2 (ug/m3)        0.664780     0.185458    -0.087843   -0.018851
NOx (ppb)          1.000000     0.354293    -0.064129    0.011879
NH3 (ug/m3)        0.354293     1.000000     0.123669    0.013482
SO2 (ug/m3)       -0.064129     0.123669     1.000000   -0.059010
CO (mg/m3)         0.011879     0.013482    -0.059010    1.000000
Ozone (ug/m3)      0.035873     0.093386     0.031709   -0.004637
Benzene (ug/m3)    0.109346     0.000982    -0.096329   -0.008758
Toluene (ug/m3)    0.264549    -0.000117    -0.167220   -0.010869

                  Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
PM2.5 (ug/m3)          0.176285         0.013691         0.119651
PM10 (ug/m3)           0.034288         0.000216         0.000219
NO (ug/m3)            -0.009765         0.128821         0.187092
NO2 (ug/m3)            0.095719         0.069164         0.252026
NOx (ppb)              0.035873         0.109346         0.264549
NH3 (ug/m3)            0.093386         0.000982        -0.000117
SO2 (ug/m3)            0.031709        -0.096329        -0.167220
CO (mg/m3)            -0.004637        -0.008758        -0.010869
Ozone (ug/m3)          1.000000        -0.014507        -0.013258
Benzene (ug/m3)       -0.014507         1.000000         0.499545
Toluene (ug/m3)       -0.013258         0.499545         1.000000
```

This method reduces the variance of the imputed variables.It doesnt preserve the relationship between variables such as correlation.This method is ineffective if the data has outliers since mean gets affected by outliers.

Hockdeck imputation

Hot deck imputation involves replacing missing values of one or more variables for a non-respondent (called the recipient) with observed values from a respondent (the donor) that is similar to the non-respondent with respect to characteristics observed by both cases.

```
df6= pd.read_excel (r'/content/Data 2017-2022.xlsx')
from sklearn.impute import KNNImputer
imputer = KNNImputer(n_neighbors=2, weights="uniform")
df6['PM2.5 (ug/m3)'] = imputer.fit_transform(df6[['PM2.5
(ug/m3)','PM10 (ug/m3)','NO (ug/m3)','NO2 (ug/m3)','NOx (ppb)','NH3
(ug/m3)','SO2 (ug/m3)','CO (mg/m3)','Ozone (ug/m3)','Benzene
(ug/m3)','Toluene (ug/m3)']])
print(df6['PM2.5 (ug/m3)'])
df6.head() #after imputation
```

```
0        32.61
1        22.93
```

```
2        24.19
3        33.61
4       129.38
         ...
1880     32.05
1881     38.95
1882     38.40
1883     27.51
1884     34.58
Name: PM2.5 (ug/m3), Length: 1885, dtype: float64
```

```
             From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0  01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1  02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2  03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3  04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4  05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN

   NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0        2.36         9.78        NaN          NaN         2.11
NaN
1        2.33         8.21        NaN          NaN         2.86
NaN
2       11.39        17.28        NaN          NaN         7.73
NaN
3        6.06        12.32        NaN          NaN         2.72
NaN
4        5.58        12.67        NaN          NaN         2.65
NaN

   Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0          24.63             0.52             2.95
1          20.49             0.13             2.01
2          13.04             0.45             3.52
3          19.42             0.65             3.98
4          25.89             0.60             3.53
```

Here it is not very effective because there are still many NaNs as some values in the row may not have similar characteristics to any other for imputing.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
```

```
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1841 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   NH3 (ug/m3)      306 non-null    float64
 8   SO2 (ug/m3)      1830 non-null   float64
 9   CO (mg/m3)       1687 non-null   float64
 10  Ozone (ug/m3)    1833 non-null   float64
 11  Benzene (ug/m3)  1859 non-null   float64
 12  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB

df6.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1885 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   NH3 (ug/m3)      306 non-null    float64
 8   SO2 (ug/m3)      1830 non-null   float64
 9   CO (mg/m3)       1687 non-null   float64
 10  Ozone (ug/m3)    1833 non-null   float64
 11  Benzene (ug/m3)  1859 non-null   float64
 12  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB

df.describe()

       PM2.5 (ug/m3)  PM10 (ug/m3)   NO (ug/m3)  NO2 (ug/m3)    NOx
(ppb)  \
count    1841.000000    306.000000  1844.000000  1843.000000
1598.000000
mean       30.500435     58.150719     6.952950    12.557347
16.026608
std        20.289850     25.930222     5.611016     8.707426
```

```
8.797421
min         0.410000     21.600000     0.010000     0.020000
0.000000
25%        16.540000     36.850000     3.295000     6.510000
9.860000
50%        27.280000     67.250000     5.460000    11.280000
14.615000
75%        39.530000     69.415000     9.400000    16.470000
20.872500
max       278.970000    371.610000    98.620000   134.760000
106.740000

       NH3 (ug/m3)  SO2 (ug/m3)   CO (mg/m3)  Ozone (ug/m3)  Benzene
(ug/m3)  \
count   306.000000  1830.000000  1687.000000    1833.000000
1859.000000
mean     12.271961     6.551770     0.827825      27.580562
0.583018
std       5.557635     5.058016     1.472130      16.822431
2.228912
min       5.360000     0.090000     0.000000       0.100000
0.000000
25%       5.360000     3.890000     0.600000      16.770000
0.000000
50%      14.455000     4.880000     0.740000      24.680000
0.000000
75%      16.675000     7.205000     0.910000      34.770000
0.365000
max      33.680000    37.180000    48.020000     183.990000
46.230000

       Toluene (ug/m3)
count      1859.000000
mean          1.940172
std           4.380171
min           0.000000
25%           0.000000
50%           0.190000
75%           2.755000
max         121.150000

df6.describe()

       PM2.5 (ug/m3)  PM10 (ug/m3)   NO (ug/m3)  NO2 (ug/m3)    NOx
(ppb)  \
count    1885.000000    306.000000  1844.000000  1843.000000
1598.000000
mean       30.467786     58.150719     6.952950    12.557347
16.026608
std        20.103691     25.930222     5.611016     8.707426
```

```
8.797421
min         0.410000     21.600000     0.010000     0.020000
0.000000
25%        16.730000     36.850000     3.295000     6.510000
9.860000
50%        27.520000     67.250000     5.460000    11.280000
14.615000
75%        39.250000     69.415000     9.400000    16.470000
20.872500
max       278.970000    371.610000    98.620000   134.760000
106.740000

       NH3 (ug/m3)  SO2 (ug/m3)   CO (mg/m3)  Ozone (ug/m3)  Benzene
(ug/m3)  \
count   306.000000  1830.000000  1687.000000    1833.000000
1859.000000
mean     12.271961     6.551770     0.827825      27.580562
0.583018
std       5.557635     5.058016     1.472130      16.822431
2.228912
min       5.360000     0.090000     0.000000       0.100000
0.000000
25%       5.360000     3.890000     0.600000      16.770000
0.000000
50%      14.455000     4.880000     0.740000      24.680000
0.000000
75%      16.675000     7.205000     0.910000      34.770000
0.365000
max      33.680000    37.180000    48.020000     183.990000
46.230000

       Toluene (ug/m3)
count      1859.000000
mean          1.940172
std           4.380171
min           0.000000
25%           0.000000
50%           0.190000
75%           2.755000
max         121.150000

df.cov()

                PM2.5 (ug/m3)  PM10 (ug/m3)  NO (ug/m3)  NO2 (ug/m3)
\
PM2.5 (ug/m3)       411.678015     30.890050   16.066220    51.241957

PM10 (ug/m3)         30.890050    672.376420  -44.806987   -24.387463

NO (ug/m3)           16.066220    -44.806987   31.483505    22.519439
```

```
NO2 (ug/m3)          51.241957    -24.387463   22.519439    75.819259

NOx (ppb)            40.344852    -61.393595   36.589019    55.207391

NH3 (ug/m3)          13.933097    -18.579278   19.546962    22.055663

SO2 (ug/m3)          -0.923446      0.531256   -1.680204    -3.889626

CO (mg/m3)           -1.522403      0.001428   -0.332261    -0.253260

Ozone (ug/m3)        60.713479     38.184933   -0.932329    14.139986

Benzene (ug/m3)       0.625041      0.034594    1.625605     1.354755

Toluene (ug/m3)      10.737620      0.074053    4.639590     9.701272


                 NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO (mg/m3)  \
PM2.5 (ug/m3)    40.344852    13.933097    -0.923446   -1.522403
PM10 (ug/m3)    -61.393595   -18.579278     0.531256    0.001428
NO (ug/m3)       36.589019    19.546962    -1.680204   -0.332261
NO2 (ug/m3)      55.207391    22.055663    -3.889626   -0.253260
NOx (ppb)        77.394611    39.637911    -3.141972    0.158700
NH3 (ug/m3)      39.637911    30.887305     8.791461    0.259349
SO2 (ug/m3)      -3.141972     8.791461    25.583531   -0.472042
CO (mg/m3)        0.158700     0.259349    -0.472042    2.167166
Ozone (ug/m3)     5.818156    23.173855     2.729160   -0.128536
Benzene (ug/m3)   2.326780     0.043199    -1.098227   -0.030333
Toluene (ug/m3)  11.059783     0.036882    -3.746441   -0.073975

                 Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
PM2.5 (ug/m3)        60.713479         0.625041        10.737620
PM10 (ug/m3)         38.184933         0.034594         0.074053
NO (ug/m3)           -0.932329         1.625605         4.639590
NO2 (ug/m3)          14.139986         1.354755         9.701272
NOx (ppb)             5.818156         2.326780        11.059783
NH3 (ug/m3)          23.173855         0.043199         0.036882
SO2 (ug/m3)           2.729160        -1.098227        -3.746441
CO (mg/m3)           -0.128536        -0.030333        -0.073975
Ozone (ug/m3)       282.994183        -0.550818        -0.989454
Benzene (ug/m3)      -0.550818         4.968048         4.877070
Toluene (ug/m3)      -0.989454         4.877070        19.185895

df6.cov() #finding covariance after imputing missing data using
hotdeck

                 PM2.5 (ug/m3)  PM10 (ug/m3)  NO (ug/m3)  NO2 (ug/m3)
\
```

PM2.5 (ug/m3)	404.158399	31.362541	15.908389	51.252191
PM10 (ug/m3)	31.362541	672.376420	-44.806987	-24.387463
NO (ug/m3)	15.908389	-44.806987	31.483505	22.519439
NO2 (ug/m3)	51.252191	-24.387463	22.519439	75.819259
NOx (ppb)	40.121315	-61.393595	36.589019	55.207391
NH3 (ug/m3)	13.795534	-18.579278	19.546962	22.055663
SO2 (ug/m3)	-1.005195	0.531256	-1.680204	-3.889626
CO (mg/m3)	-1.451634	0.001428	-0.332261	-0.253260
Ozone (ug/m3)	61.966990	38.184933	-0.932329	14.139986
Benzene (ug/m3)	0.623286	0.034594	1.625605	1.354755
Toluene (ug/m3)	10.641059	0.074053	4.639590	9.701272

	NOx (ppb)	NH3 (ug/m3)	SO2 (ug/m3)	CO (mg/m3)	\
PM2.5 (ug/m3)	40.121315	13.795534	-1.005195	-1.451634	
PM10 (ug/m3)	-61.393595	-18.579278	0.531256	0.001428	
NO (ug/m3)	36.589019	19.546962	-1.680204	-0.332261	
NO2 (ug/m3)	55.207391	22.055663	-3.889626	-0.253260	
NOx (ppb)	77.394611	39.637911	-3.141972	0.158700	
NH3 (ug/m3)	39.637911	30.887305	8.791461	0.259349	
SO2 (ug/m3)	-3.141972	8.791461	25.583531	-0.472042	
CO (mg/m3)	0.158700	0.259349	-0.472042	2.167166	
Ozone (ug/m3)	5.818156	23.173855	2.729160	-0.128536	
Benzene (ug/m3)	2.326780	0.043199	-1.098227	-0.030333	
Toluene (ug/m3)	11.059783	0.036882	-3.746441	-0.073975	

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
PM2.5 (ug/m3)	61.966990	0.623286	10.641059
PM10 (ug/m3)	38.184933	0.034594	0.074053
NO (ug/m3)	-0.932329	1.625605	4.639590
NO2 (ug/m3)	14.139986	1.354755	9.701272
NOx (ppb)	5.818156	2.326780	11.059783
NH3 (ug/m3)	23.173855	0.043199	0.036882
SO2 (ug/m3)	2.729160	-1.098227	-3.746441
CO (mg/m3)	-0.128536	-0.030333	-0.073975
Ozone (ug/m3)	282.994183	-0.550818	-0.989454
Benzene (ug/m3)	-0.550818	4.968048	4.877070
Toluene (ug/m3)	-0.989454	4.877070	19.185895

```
df.corr()
```

	PM2.5 (ug/m3)	PM10 (ug/m3)	NO (ug/m3)	NO2 (ug/m3)
PM2.5 (ug/m3)	1.000000	0.077045	0.140907	0.290013
PM10 (ug/m3)	0.077045	1.000000	-0.342711	-0.205761
NO (ug/m3)	0.140907	-0.342711	1.000000	0.460980
NO2 (ug/m3)	0.290013	-0.205761	0.460980	1.000000
NOx (ppb)	0.217120	-0.272646	0.714484	0.724070
NH3 (ug/m3)	0.161947	-0.128751	0.702132	0.871728
SO2 (ug/m3)	-0.009008	0.004149	-0.059458	-0.089198
CO (mg/m3)	-0.049465	0.000288	-0.039316	-0.019531
Ozone (ug/m3)	0.178225	0.058454	-0.009927	0.097121
Benzene (ug/m3)	0.013715	0.011016	0.129351	0.069390
Toluene (ug/m3)	0.120031	0.022940	0.188078	0.253154

	NOx (ppb)	NH3 (ug/m3)	SO2 (ug/m3)	CO (mg/m3)
PM2.5 (ug/m3)	0.217120	0.161947	-0.009008	-0.049465
PM10 (ug/m3)	-0.272646	-0.128751	0.004149	0.000288
NO (ug/m3)	0.714484	0.702132	-0.059458	-0.039316
NO2 (ug/m3)	0.724070	0.871728	-0.089198	-0.019531
NOx (ppb)	1.000000	0.825683	-0.068077	0.011973
NH3 (ug/m3)	0.825683	1.000000	0.323777	0.242040
SO2 (ug/m3)	-0.068077	0.323777	1.000000	-0.061407
CO (mg/m3)	0.011973	0.242040	-0.061407	1.000000
Ozone (ug/m3)	0.038354	0.170788	0.032032	-0.008149
Benzene (ug/m3)	0.122707	0.064346	-0.096417	-0.008974
Toluene (ug/m3)	0.278826	0.053442	-0.167630	-0.010942

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
PM2.5 (ug/m3)	0.178225	0.013715	0.120031
PM10 (ug/m3)	0.058454	0.011016	0.022940
NO (ug/m3)	-0.009927	0.129351	0.188078
NO2 (ug/m3)	0.097121	0.069390	0.253154
NOx (ppb)	0.038354	0.122707	0.278826
NH3 (ug/m3)	0.170788	0.064346	0.053442
SO2 (ug/m3)	0.032032	-0.096417	-0.167630
CO (mg/m3)	-0.008149	-0.008974	-0.010942

```
Ozone (ug/m3)         1.000000        -0.014557        -0.013333
Benzene (ug/m3)      -0.014557         1.000000         0.499545
Toluene (ug/m3)      -0.013333         0.499545         1.000000
```

```
df6.corr() #finding correlation after imputing missing data using
hotdeck
```

```
                PM2.5 (ug/m3)  PM10 (ug/m3)  NO (ug/m3)  NO2 (ug/m3)
\
PM2.5 (ug/m3)         1.000000      0.078255    0.139757     0.290068

PM10 (ug/m3)          0.078255      1.000000   -0.342711    -0.205761

NO (ug/m3)            0.139757     -0.342711    1.000000     0.460980

NO2 (ug/m3)           0.290068     -0.205761    0.460980     1.000000

NOx (ppb)             0.217071     -0.272646    0.714484     0.724070

NH3 (ug/m3)           0.160586     -0.128751    0.702132     0.871728

SO2 (ug/m3)          -0.009821      0.004149   -0.059458    -0.089198

CO (mg/m3)           -0.047706      0.000288   -0.039316    -0.019531

Ozone (ug/m3)         0.181929      0.058454   -0.009927     0.097121

Benzene (ug/m3)       0.013835      0.011016    0.129351     0.069390

Toluene (ug/m3)       0.120195      0.022940    0.188078     0.253154


                NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO (mg/m3)  \
PM2.5 (ug/m3)     0.217071     0.160586    -0.009821   -0.047706
PM10 (ug/m3)     -0.272646    -0.128751     0.004149    0.000288
NO (ug/m3)        0.714484     0.702132    -0.059458   -0.039316
NO2 (ug/m3)       0.724070     0.871728    -0.089198   -0.019531
NOx (ppb)         1.000000     0.825683    -0.068077    0.011973
NH3 (ug/m3)       0.825683     1.000000     0.323777    0.242040
SO2 (ug/m3)      -0.068077     0.323777     1.000000   -0.061407
CO (mg/m3)        0.011973     0.242040    -0.061407    1.000000
Ozone (ug/m3)     0.038354     0.170788     0.032032   -0.008149
Benzene (ug/m3)   0.122707     0.064346    -0.096417   -0.008974
Toluene (ug/m3)   0.278826     0.053442    -0.167630   -0.010942

                Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
PM2.5 (ug/m3)        0.181929         0.013835         0.120195
PM10 (ug/m3)         0.058454         0.011016         0.022940
NO (ug/m3)          -0.009927         0.129351         0.188078
```

```
NO2 (ug/m3)           0.097121         0.069390         0.253154
NOx (ppb)             0.038354         0.122707         0.278826
NH3 (ug/m3)           0.170788         0.064346         0.053442
SO2 (ug/m3)           0.032032        -0.096417        -0.167630
CO (mg/m3)           -0.008149        -0.008974        -0.010942
Ozone (ug/m3)         1.000000        -0.014557        -0.013333
Benzene (ug/m3)      -0.014557         1.000000         0.499545
Toluene (ug/m3)      -0.013333         0.499545         1.000000
```

Filling with before observation

This method would fill the missing values with first non-missing value that occurs before it. This will be carried forward until another non-null value is encountered.

```
df.head(50) #before the impuation is done
```

```
               From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0   01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1   02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2   03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3   04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4   05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN
5   06-Jan-2017 - 00:00  07-Jan-2017 - 00:00          64.52
NaN
6   07-Jan-2017 - 00:00  08-Jan-2017 - 00:00          45.01
NaN
7   08-Jan-2017 - 00:00  09-Jan-2017 - 00:00          32.71
NaN
8   09-Jan-2017 - 00:00  10-Jan-2017 - 00:00          33.66
NaN
9   10-Jan-2017 - 00:00  11-Jan-2017 - 00:00          34.82
NaN
10  11-Jan-2017 - 00:00  12-Jan-2017 - 00:00          38.33
NaN
11  12-Jan-2017 - 00:00  13-Jan-2017 - 00:00          42.32
NaN
12  13-Jan-2017 - 00:00  14-Jan-2017 - 00:00         103.51
NaN
13  14-Jan-2017 - 00:00  15-Jan-2017 - 00:00          33.96
NaN
14  15-Jan-2017 - 00:00  16-Jan-2017 - 00:00          54.93
NaN
15  16-Jan-2017 - 00:00  17-Jan-2017 - 00:00          46.01
```

NaN				
16	17-Jan-2017 - 00:00	18-Jan-2017 - 00:00	47.76	
NaN				
17	18-Jan-2017 - 00:00	19-Jan-2017 - 00:00	36.59	
NaN				
18	19-Jan-2017 - 00:00	20-Jan-2017 - 00:00	42.82	
NaN				
19	20-Jan-2017 - 00:00	21-Jan-2017 - 00:00	68.26	
NaN				
20	21-Jan-2017 - 00:00	22-Jan-2017 - 00:00	38.96	
NaN				
21	22-Jan-2017 - 00:00	23-Jan-2017 - 00:00	26.28	
NaN				
22	23-Jan-2017 - 00:00	24-Jan-2017 - 00:00	33.59	
NaN				
23	24-Jan-2017 - 00:00	25-Jan-2017 - 00:00	29.59	
NaN				
24	25-Jan-2017 - 00:00	26-Jan-2017 - 00:00	40.49	
NaN				
25	26-Jan-2017 - 00:00	27-Jan-2017 - 00:00	29.95	
NaN				
26	27-Jan-2017 - 00:00	28-Jan-2017 - 00:00	18.93	
NaN				
27	28-Jan-2017 - 00:00	29-Jan-2017 - 00:00	7.64	
NaN				
28	29-Jan-2017 - 00:00	30-Jan-2017 - 00:00	8.65	
NaN				
29	30-Jan-2017 - 00:00	31-Jan-2017 - 00:00	19.69	
NaN				
30	31-Jan-2017 - 00:00	01-Feb-2017 - 00:00	35.80	
NaN				
31	01-Feb-2017 - 00:00	02-Feb-2017 - 00:00	37.48	
NaN				
32	02-Feb-2017 - 00:00	03-Feb-2017 - 00:00	34.84	
NaN				
33	03-Feb-2017 - 00:00	04-Feb-2017 - 00:00	36.89	
NaN				
34	04-Feb-2017 - 00:00	05-Feb-2017 - 00:00	40.92	
NaN				
35	05-Feb-2017 - 00:00	06-Feb-2017 - 00:00	43.07	
NaN				
36	06-Feb-2017 - 00:00	07-Feb-2017 - 00:00	42.05	
NaN				
37	07-Feb-2017 - 00:00	08-Feb-2017 - 00:00	47.71	
NaN				
38	08-Feb-2017 - 00:00	09-Feb-2017 - 00:00	61.60	
NaN				
39	09-Feb-2017 - 00:00	10-Feb-2017 - 00:00	54.49	
NaN				
40	10-Feb-2017 - 00:00	11-Feb-2017 - 00:00	50.29	

```
NaN
41  11-Feb-2017 - 00:00  12-Feb-2017 - 00:00          44.20
NaN
42  12-Feb-2017 - 00:00  13-Feb-2017 - 00:00          26.51
NaN
43  13-Feb-2017 - 00:00  14-Feb-2017 - 00:00          17.22
NaN
44  14-Feb-2017 - 00:00  15-Feb-2017 - 00:00          22.04
NaN
45  15-Feb-2017 - 00:00  16-Feb-2017 - 00:00          30.00
NaN
46  16-Feb-2017 - 00:00  17-Feb-2017 - 00:00          36.92
NaN
47  17-Feb-2017 - 00:00  18-Feb-2017 - 00:00          38.03
NaN
48  18-Feb-2017 - 00:00  19-Feb-2017 - 00:00          29.63
NaN
49  19-Feb-2017 - 00:00  20-Feb-2017 - 00:00          34.19
NaN

    NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0         2.36         9.78        NaN          NaN         2.11
NaN
1         2.33         8.21        NaN          NaN         2.86
NaN
2        11.39        17.28        NaN          NaN         7.73
NaN
3         6.06        12.32        NaN          NaN         2.72
NaN
4         5.58        12.67        NaN          NaN         2.65
NaN
5         6.91        14.38        NaN          NaN         2.28
NaN
6         5.72         9.66        NaN          NaN         2.19
NaN
7         4.33        10.57        NaN          NaN         2.23
NaN
8         6.02        15.74        NaN          NaN         3.76
NaN
9         6.96        16.31        NaN          NaN         2.71
NaN
10        6.57        18.45        NaN          NaN         2.75
NaN
11       22.63        26.31        NaN          NaN         2.57
NaN
12       32.73        38.26        NaN          NaN         2.65
NaN
13       24.83        27.27        NaN          NaN         2.48
NaN
```

14	16.29	20.86	NaN	NaN	3.54	NaN
15	4.18	14.54	NaN	NaN	3.23	NaN
16	4.08	20.86	NaN	NaN	4.49	NaN
17	4.76	20.39	NaN	NaN	3.52	NaN
18	2.71	15.88	NaN	NaN	2.55	NaN
19	2.86	18.64	NaN	NaN	4.17	NaN
20	2.37	15.49	NaN	NaN	2.53	NaN
21	2.47	15.81	NaN	NaN	2.99	NaN
22	3.12	15.40	NaN	NaN	4.70	NaN
23	2.83	14.93	NaN	NaN	3.77	NaN
24	2.37	12.49	NaN	NaN	2.63	NaN
25	2.44	12.68	NaN	NaN	2.60	NaN
26	2.53	12.44	NaN	NaN	2.46	NaN
27	2.59	11.63	NaN	NaN	2.35	NaN
28	6.07	13.59	NaN	NaN	2.90	NaN
29	6.07	18.00	NaN	NaN	2.77	NaN
30	6.47	17.91	NaN	NaN	3.97	NaN
31	3.63	16.19	NaN	NaN	2.27	1.91
32	6.89	18.19	NaN	NaN	2.41	1.01
33	6.19	18.27	NaN	NaN	3.09	NaN
34	6.62	17.96	NaN	NaN	2.64	NaN
35	5.41	18.31	NaN	NaN	2.90	NaN
36	6.82	21.76	NaN	NaN	2.83	NaN
37	9.01	25.18	NaN	NaN	4.03	NaN
38	9.12	23.48	NaN	NaN	2.85	NaN

```
39         5.63         22.22         NaN           NaN          2.97
NaN
40         6.47         21.84         NaN           NaN          2.74
NaN
41        10.70         22.33         NaN           NaN          2.32
NaN
42         5.93         17.56         NaN           NaN          2.20
NaN
43         5.46         15.74         NaN           NaN          2.54
NaN
44         2.54         12.35         NaN           NaN          2.17
NaN
45         3.77         14.66         NaN           NaN          2.57
NaN
46         3.26         14.88         NaN           NaN          2.20
NaN
47         5.55         18.64         NaN           NaN          2.10
1.21
48         5.58         17.27         NaN           NaN          1.99
1.21
49         5.52         18.13         NaN           NaN          3.60
NaN
```

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
0	24.63	0.52	2.95
1	20.49	0.13	2.01
2	13.04	0.45	3.52
3	19.42	0.65	3.98
4	25.89	0.60	3.53
5	25.16	0.74	3.80
6	35.32	0.36	3.20
7	31.22	0.34	1.94
8	24.32	0.51	3.27
9	21.71	0.55	3.88
10	24.67	0.74	3.79
11	30.78	0.74	3.52
12	30.94	6.60	4.35
13	44.53	0.23	0.77
14	45.14	1.00	2.82
15	50.70	0.44	1.41
16	52.13	0.55	2.52
17	50.51	0.88	2.55
18	63.89	0.00	1.10
19	62.39	0.26	2.05
20	55.96	0.00	1.59
21	62.37	0.00	2.37
22	54.17	0.42	2.69
23	65.30	0.31	2.15
24	71.66	0.00	0.84
25	51.52	0.00	0.77

```
26          24.48             0.00             1.33
27          16.77             0.00             1.06
28          15.22             0.21             1.67
29          21.93             0.41             2.19
30          23.68             1.05             3.39
31          33.67             0.06             2.74
32          27.56             0.51             2.93
33          33.36             0.54             3.29
34          38.83             0.47             2.02
35          32.93             0.65             0.78
36          32.11             0.84             0.54
37          35.07             1.32             1.34
38          40.37             1.50             2.12
39          39.35             1.32             1.89
40          32.65             1.34             1.90
41          29.16             1.34             1.90
42          34.39             1.41             1.96
43          34.28             1.90             2.33
44          52.83             2.23             2.59
45          49.97             1.94             2.37
46          67.63             1.67             2.16
47          47.98             1.34             1.90
48          39.44             1.34             1.90
49          32.65             1.39             1.95
```

```
df7 = pd.read_excel (r'/content/Data 2017-2022.xlsx')
df7.ffill(inplace=True)
df7.head(50) #after imputation
```

```
               From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0   01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1   02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2   03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3   04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4   05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN
5   06-Jan-2017 - 00:00  07-Jan-2017 - 00:00          64.52
NaN
6   07-Jan-2017 - 00:00  08-Jan-2017 - 00:00          45.01
NaN
7   08-Jan-2017 - 00:00  09-Jan-2017 - 00:00          32.71
NaN
8   09-Jan-2017 - 00:00  10-Jan-2017 - 00:00          33.66
NaN
9   10-Jan-2017 - 00:00  11-Jan-2017 - 00:00          34.82
NaN
```

10	11-Jan-2017 - 00:00	12-Jan-2017 - 00:00	38.33	NaN
11	12-Jan-2017 - 00:00	13-Jan-2017 - 00:00	42.32	NaN
12	13-Jan-2017 - 00:00	14-Jan-2017 - 00:00	103.51	NaN
13	14-Jan-2017 - 00:00	15-Jan-2017 - 00:00	33.96	NaN
14	15-Jan-2017 - 00:00	16-Jan-2017 - 00:00	54.93	NaN
15	16-Jan-2017 - 00:00	17-Jan-2017 - 00:00	46.01	NaN
16	17-Jan-2017 - 00:00	18-Jan-2017 - 00:00	47.76	NaN
17	18-Jan-2017 - 00:00	19-Jan-2017 - 00:00	36.59	NaN
18	19-Jan-2017 - 00:00	20-Jan-2017 - 00:00	42.82	NaN
19	20-Jan-2017 - 00:00	21-Jan-2017 - 00:00	68.26	NaN
20	21-Jan-2017 - 00:00	22-Jan-2017 - 00:00	38.96	NaN
21	22-Jan-2017 - 00:00	23-Jan-2017 - 00:00	26.28	NaN
22	23-Jan-2017 - 00:00	24-Jan-2017 - 00:00	33.59	NaN
23	24-Jan-2017 - 00:00	25-Jan-2017 - 00:00	29.59	NaN
24	25-Jan-2017 - 00:00	26-Jan-2017 - 00:00	40.49	NaN
25	26-Jan-2017 - 00:00	27-Jan-2017 - 00:00	29.95	NaN
26	27-Jan-2017 - 00:00	28-Jan-2017 - 00:00	18.93	NaN
27	28-Jan-2017 - 00:00	29-Jan-2017 - 00:00	7.64	NaN
28	29-Jan-2017 - 00:00	30-Jan-2017 - 00:00	8.65	NaN
29	30-Jan-2017 - 00:00	31-Jan-2017 - 00:00	19.69	NaN
30	31-Jan-2017 - 00:00	01-Feb-2017 - 00:00	35.80	NaN
31	01-Feb-2017 - 00:00	02-Feb-2017 - 00:00	37.48	NaN
32	02-Feb-2017 - 00:00	03-Feb-2017 - 00:00	34.84	NaN
33	03-Feb-2017 - 00:00	04-Feb-2017 - 00:00	36.89	NaN
34	04-Feb-2017 - 00:00	05-Feb-2017 - 00:00	40.92	NaN

```
35  05-Feb-2017 - 00:00  06-Feb-2017 - 00:00          43.07
NaN
36  06-Feb-2017 - 00:00  07-Feb-2017 - 00:00          42.05
NaN
37  07-Feb-2017 - 00:00  08-Feb-2017 - 00:00          47.71
NaN
38  08-Feb-2017 - 00:00  09-Feb-2017 - 00:00          61.60
NaN
39  09-Feb-2017 - 00:00  10-Feb-2017 - 00:00          54.49
NaN
40  10-Feb-2017 - 00:00  11-Feb-2017 - 00:00          50.29
NaN
41  11-Feb-2017 - 00:00  12-Feb-2017 - 00:00          44.20
NaN
42  12-Feb-2017 - 00:00  13-Feb-2017 - 00:00          26.51
NaN
43  13-Feb-2017 - 00:00  14-Feb-2017 - 00:00          17.22
NaN
44  14-Feb-2017 - 00:00  15-Feb-2017 - 00:00          22.04
NaN
45  15-Feb-2017 - 00:00  16-Feb-2017 - 00:00          30.00
NaN
46  16-Feb-2017 - 00:00  17-Feb-2017 - 00:00          36.92
NaN
47  17-Feb-2017 - 00:00  18-Feb-2017 - 00:00          38.03
NaN
48  18-Feb-2017 - 00:00  19-Feb-2017 - 00:00          29.63
NaN
49  19-Feb-2017 - 00:00  20-Feb-2017 - 00:00          34.19
NaN

    NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0         2.36         9.78        NaN          NaN         2.11
NaN
1         2.33         8.21        NaN          NaN         2.86
NaN
2        11.39        17.28        NaN          NaN         7.73
NaN
3         6.06        12.32        NaN          NaN         2.72
NaN
4         5.58        12.67        NaN          NaN         2.65
NaN
5         6.91        14.38        NaN          NaN         2.28
NaN
6         5.72         9.66        NaN          NaN         2.19
NaN
7         4.33        10.57        NaN          NaN         2.23
NaN
8         6.02        15.74        NaN          NaN         3.76
```

NaN						
9	6.96	16.31	NaN	NaN	2.71	
NaN						
10	6.57	18.45	NaN	NaN	2.75	
NaN						
11	22.63	26.31	NaN	NaN	2.57	
NaN						
12	32.73	38.26	NaN	NaN	2.65	
NaN						
13	24.83	27.27	NaN	NaN	2.48	
NaN						
14	16.29	20.86	NaN	NaN	3.54	
NaN						
15	4.18	14.54	NaN	NaN	3.23	
NaN						
16	4.08	20.86	NaN	NaN	4.49	
NaN						
17	4.76	20.39	NaN	NaN	3.52	
NaN						
18	2.71	15.88	NaN	NaN	2.55	
NaN						
19	2.86	18.64	NaN	NaN	4.17	
NaN						
20	2.37	15.49	NaN	NaN	2.53	
NaN						
21	2.47	15.81	NaN	NaN	2.99	
NaN						
22	3.12	15.40	NaN	NaN	4.70	
NaN						
23	2.83	14.93	NaN	NaN	3.77	
NaN						
24	2.37	12.49	NaN	NaN	2.63	
NaN						
25	2.44	12.68	NaN	NaN	2.60	
NaN						
26	2.53	12.44	NaN	NaN	2.46	
NaN						
27	2.59	11.63	NaN	NaN	2.35	
NaN						
28	6.07	13.59	NaN	NaN	2.90	
NaN						
29	6.07	18.00	NaN	NaN	2.77	
NaN						
30	6.47	17.91	NaN	NaN	3.97	
NaN						
31	3.63	16.19	NaN	NaN	2.27	
1.91						
32	6.89	18.19	NaN	NaN	2.41	
1.01						
33	6.19	18.27	NaN	NaN	3.09	

```
1.01
34        6.62        17.96        NaN          NaN          2.64
1.01
35        5.41        18.31        NaN          NaN          2.90
1.01
36        6.82        21.76        NaN          NaN          2.83
1.01
37        9.01        25.18        NaN          NaN          4.03
1.01
38        9.12        23.48        NaN          NaN          2.85
1.01
39        5.63        22.22        NaN          NaN          2.97
1.01
40        6.47        21.84        NaN          NaN          2.74
1.01
41       10.70        22.33        NaN          NaN          2.32
1.01
42        5.93        17.56        NaN          NaN          2.20
1.01
43        5.46        15.74        NaN          NaN          2.54
1.01
44        2.54        12.35        NaN          NaN          2.17
1.01
45        3.77        14.66        NaN          NaN          2.57
1.01
46        3.26        14.88        NaN          NaN          2.20
1.01
47        5.55        18.64        NaN          NaN          2.10
1.21
48        5.58        17.27        NaN          NaN          1.99
1.21
49        5.52        18.13        NaN          NaN          3.60
1.21

    Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0           24.63             0.52             2.95
1           20.49             0.13             2.01
2           13.04             0.45             3.52
3           19.42             0.65             3.98
4           25.89             0.60             3.53
5           25.16             0.74             3.80
6           35.32             0.36             3.20
7           31.22             0.34             1.94
8           24.32             0.51             3.27
9           21.71             0.55             3.88
10          24.67             0.74             3.79
11          30.78             0.74             3.52
12          30.94             6.60             4.35
13          44.53             0.23             0.77
14          45.14             1.00             2.82
```

```
15          50.70             0.44             1.41
16          52.13             0.55             2.52
17          50.51             0.88             2.55
18          63.89             0.00             1.10
19          62.39             0.26             2.05
20          55.96             0.00             1.59
21          62.37             0.00             2.37
22          54.17             0.42             2.69
23          65.30             0.31             2.15
24          71.66             0.00             0.84
25          51.52             0.00             0.77
26          24.48             0.00             1.33
27          16.77             0.00             1.06
28          15.22             0.21             1.67
29          21.93             0.41             2.19
30          23.68             1.05             3.39
31          33.67             0.06             2.74
32          27.56             0.51             2.93
33          33.36             0.54             3.29
34          38.83             0.47             2.02
35          32.93             0.65             0.78
36          32.11             0.84             0.54
37          35.07             1.32             1.34
38          40.37             1.50             2.12
39          39.35             1.32             1.89
40          32.65             1.34             1.90
41          29.16             1.34             1.90
42          34.39             1.41             1.96
43          34.28             1.90             2.33
44          52.83             2.23             2.59
45          49.97             1.94             2.37
46          67.63             1.67             2.16
47          47.98             1.34             1.90
48          39.44             1.34             1.90
49          32.65             1.39             1.95
```

This method is not very effective due to the large number of persisting NaN values even after the imputation, this is due to the reason that there are a lot of NaN values in a column before the first non-null value, which cannot be replace with any numerical.

```
df.info()
df7.info()
#to show the number of non null value which still remains after
imputation
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
```

```
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1841 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   NH3 (ug/m3)      306 non-null    float64
 8   SO2 (ug/m3)      1830 non-null   float64
 9   CO (mg/m3)       1687 non-null   float64
 10  Ozone (ug/m3)    1833 non-null   float64
 11  Benzene (ug/m3)  1859 non-null   float64
 12  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1885 non-null   float64
 3   PM10 (ug/m3)     309 non-null    float64
 4   NO (ug/m3)       1885 non-null   float64
 5   NO2 (ug/m3)      1885 non-null   float64
 6   NOx (ppb)        1609 non-null   float64
 7   NH3 (ug/m3)      309 non-null    float64
 8   SO2 (ug/m3)      1885 non-null   float64
 9   CO (mg/m3)       1854 non-null   float64
 10  Ozone (ug/m3)    1885 non-null   float64
 11  Benzene (ug/m3)  1885 non-null   float64
 12  Toluene (ug/m3)  1885 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB
```

Filling with next observation

This method would fill the missing values with first non-missing value that occurs after it. The imputation happens backwards and is continued until a nn value is encountered.

```
df.head(50) #dataset before handling the missing data
```

```
               From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0   01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1   02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2   03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
```

				NaN
3	04-Jan-2017 - 00:00	05-Jan-2017 - 00:00	33.61	NaN
4	05-Jan-2017 - 00:00	06-Jan-2017 - 00:00	129.38	NaN
5	06-Jan-2017 - 00:00	07-Jan-2017 - 00:00	64.52	NaN
6	07-Jan-2017 - 00:00	08-Jan-2017 - 00:00	45.01	NaN
7	08-Jan-2017 - 00:00	09-Jan-2017 - 00:00	32.71	NaN
8	09-Jan-2017 - 00:00	10-Jan-2017 - 00:00	33.66	NaN
9	10-Jan-2017 - 00:00	11-Jan-2017 - 00:00	34.82	NaN
10	11-Jan-2017 - 00:00	12-Jan-2017 - 00:00	38.33	NaN
11	12-Jan-2017 - 00:00	13-Jan-2017 - 00:00	42.32	NaN
12	13-Jan-2017 - 00:00	14-Jan-2017 - 00:00	103.51	NaN
13	14-Jan-2017 - 00:00	15-Jan-2017 - 00:00	33.96	NaN
14	15-Jan-2017 - 00:00	16-Jan-2017 - 00:00	54.93	NaN
15	16-Jan-2017 - 00:00	17-Jan-2017 - 00:00	46.01	NaN
16	17-Jan-2017 - 00:00	18-Jan-2017 - 00:00	47.76	NaN
17	18-Jan-2017 - 00:00	19-Jan-2017 - 00:00	36.59	NaN
18	19-Jan-2017 - 00:00	20-Jan-2017 - 00:00	42.82	NaN
19	20-Jan-2017 - 00:00	21-Jan-2017 - 00:00	68.26	NaN
20	21-Jan-2017 - 00:00	22-Jan-2017 - 00:00	38.96	NaN
21	22-Jan-2017 - 00:00	23-Jan-2017 - 00:00	26.28	NaN
22	23-Jan-2017 - 00:00	24-Jan-2017 - 00:00	33.59	NaN
23	24-Jan-2017 - 00:00	25-Jan-2017 - 00:00	29.59	NaN
24	25-Jan-2017 - 00:00	26-Jan-2017 - 00:00	40.49	NaN
25	26-Jan-2017 - 00:00	27-Jan-2017 - 00:00	29.95	NaN
26	27-Jan-2017 - 00:00	28-Jan-2017 - 00:00	18.93	NaN
27	28-Jan-2017 - 00:00	29-Jan-2017 - 00:00	7.64	

```
NaN
28  29-Jan-2017 - 00:00  30-Jan-2017 - 00:00           8.65
NaN
29  30-Jan-2017 - 00:00  31-Jan-2017 - 00:00          19.69
NaN
30  31-Jan-2017 - 00:00  01-Feb-2017 - 00:00          35.80
NaN
31  01-Feb-2017 - 00:00  02-Feb-2017 - 00:00          37.48
NaN
32  02-Feb-2017 - 00:00  03-Feb-2017 - 00:00          34.84
NaN
33  03-Feb-2017 - 00:00  04-Feb-2017 - 00:00          36.89
NaN
34  04-Feb-2017 - 00:00  05-Feb-2017 - 00:00          40.92
NaN
35  05-Feb-2017 - 00:00  06-Feb-2017 - 00:00          43.07
NaN
36  06-Feb-2017 - 00:00  07-Feb-2017 - 00:00          42.05
NaN
37  07-Feb-2017 - 00:00  08-Feb-2017 - 00:00          47.71
NaN
38  08-Feb-2017 - 00:00  09-Feb-2017 - 00:00          61.60
NaN
39  09-Feb-2017 - 00:00  10-Feb-2017 - 00:00          54.49
NaN
40  10-Feb-2017 - 00:00  11-Feb-2017 - 00:00          50.29
NaN
41  11-Feb-2017 - 00:00  12-Feb-2017 - 00:00          44.20
NaN
42  12-Feb-2017 - 00:00  13-Feb-2017 - 00:00          26.51
NaN
43  13-Feb-2017 - 00:00  14-Feb-2017 - 00:00          17.22
NaN
44  14-Feb-2017 - 00:00  15-Feb-2017 - 00:00          22.04
NaN
45  15-Feb-2017 - 00:00  16-Feb-2017 - 00:00          30.00
NaN
46  16-Feb-2017 - 00:00  17-Feb-2017 - 00:00          36.92
NaN
47  17-Feb-2017 - 00:00  18-Feb-2017 - 00:00          38.03
NaN
48  18-Feb-2017 - 00:00  19-Feb-2017 - 00:00          29.63
NaN
49  19-Feb-2017 - 00:00  20-Feb-2017 - 00:00          34.19
NaN

    NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0         2.36         9.78        NaN          NaN         2.11
NaN
```

1	2.33	8.21	NaN	NaN	2.86	NaN
2	11.39	17.28	NaN	NaN	7.73	NaN
3	6.06	12.32	NaN	NaN	2.72	NaN
4	5.58	12.67	NaN	NaN	2.65	NaN
5	6.91	14.38	NaN	NaN	2.28	NaN
6	5.72	9.66	NaN	NaN	2.19	NaN
7	4.33	10.57	NaN	NaN	2.23	NaN
8	6.02	15.74	NaN	NaN	3.76	NaN
9	6.96	16.31	NaN	NaN	2.71	NaN
10	6.57	18.45	NaN	NaN	2.75	NaN
11	22.63	26.31	NaN	NaN	2.57	NaN
12	32.73	38.26	NaN	NaN	2.65	NaN
13	24.83	27.27	NaN	NaN	2.48	NaN
14	16.29	20.86	NaN	NaN	3.54	NaN
15	4.18	14.54	NaN	NaN	3.23	NaN
16	4.08	20.86	NaN	NaN	4.49	NaN
17	4.76	20.39	NaN	NaN	3.52	NaN
18	2.71	15.88	NaN	NaN	2.55	NaN
19	2.86	18.64	NaN	NaN	4.17	NaN
20	2.37	15.49	NaN	NaN	2.53	NaN
21	2.47	15.81	NaN	NaN	2.99	NaN
22	3.12	15.40	NaN	NaN	4.70	NaN
23	2.83	14.93	NaN	NaN	3.77	NaN
24	2.37	12.49	NaN	NaN	2.63	NaN
25	2.44	12.68	NaN	NaN	2.60	NaN

26	2.53	12.44	NaN	NaN	2.46	NaN
27	2.59	11.63	NaN	NaN	2.35	NaN
28	6.07	13.59	NaN	NaN	2.90	NaN
29	6.07	18.00	NaN	NaN	2.77	NaN
30	6.47	17.91	NaN	NaN	3.97	NaN
31	3.63	16.19	NaN	NaN	2.27	1.91
32	6.89	18.19	NaN	NaN	2.41	1.01
33	6.19	18.27	NaN	NaN	3.09	NaN
34	6.62	17.96	NaN	NaN	2.64	NaN
35	5.41	18.31	NaN	NaN	2.90	NaN
36	6.82	21.76	NaN	NaN	2.83	NaN
37	9.01	25.18	NaN	NaN	4.03	NaN
38	9.12	23.48	NaN	NaN	2.85	NaN
39	5.63	22.22	NaN	NaN	2.97	NaN
40	6.47	21.84	NaN	NaN	2.74	NaN
41	10.70	22.33	NaN	NaN	2.32	NaN
42	5.93	17.56	NaN	NaN	2.20	NaN
43	5.46	15.74	NaN	NaN	2.54	NaN
44	2.54	12.35	NaN	NaN	2.17	NaN
45	3.77	14.66	NaN	NaN	2.57	NaN
46	3.26	14.88	NaN	NaN	2.20	NaN
47	5.55	18.64	NaN	NaN	2.10	1.21
48	5.58	17.27	NaN	NaN	1.99	1.21
49	5.52	18.13	NaN	NaN	3.60	NaN

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)

0	24.63	0.52	2.95
1	20.49	0.13	2.01
2	13.04	0.45	3.52
3	19.42	0.65	3.98
4	25.89	0.60	3.53
5	25.16	0.74	3.80
6	35.32	0.36	3.20
7	31.22	0.34	1.94
8	24.32	0.51	3.27
9	21.71	0.55	3.88
10	24.67	0.74	3.79
11	30.78	0.74	3.52
12	30.94	6.60	4.35
13	44.53	0.23	0.77
14	45.14	1.00	2.82
15	50.70	0.44	1.41
16	52.13	0.55	2.52
17	50.51	0.88	2.55
18	63.89	0.00	1.10
19	62.39	0.26	2.05
20	55.96	0.00	1.59
21	62.37	0.00	2.37
22	54.17	0.42	2.69
23	65.30	0.31	2.15
24	71.66	0.00	0.84
25	51.52	0.00	0.77
26	24.48	0.00	1.33
27	16.77	0.00	1.06
28	15.22	0.21	1.67
29	21.93	0.41	2.19
30	23.68	1.05	3.39
31	33.67	0.06	2.74
32	27.56	0.51	2.93
33	33.36	0.54	3.29
34	38.83	0.47	2.02
35	32.93	0.65	0.78
36	32.11	0.84	0.54
37	35.07	1.32	1.34
38	40.37	1.50	2.12
39	39.35	1.32	1.89
40	32.65	1.34	1.90
41	29.16	1.34	1.90
42	34.39	1.41	1.96
43	34.28	1.90	2.33
44	52.83	2.23	2.59
45	49.97	1.94	2.37
46	67.63	1.67	2.16
47	47.98	1.34	1.90
48	39.44	1.34	1.90
49	32.65	1.39	1.95

```
df8 = pd.read_excel (r'/content/Data 2017-2022.xlsx')
df8.bfill(inplace=True)
df8.head(50) #after imputation
```

```
              From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0   01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
60.71
1   02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
60.71
2   03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
60.71
3   04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
60.71
4   05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
60.71
5   06-Jan-2017 - 00:00  07-Jan-2017 - 00:00          64.52
60.71
6   07-Jan-2017 - 00:00  08-Jan-2017 - 00:00          45.01
60.71
7   08-Jan-2017 - 00:00  09-Jan-2017 - 00:00          32.71
60.71
8   09-Jan-2017 - 00:00  10-Jan-2017 - 00:00          33.66
60.71
9   10-Jan-2017 - 00:00  11-Jan-2017 - 00:00          34.82
60.71
10  11-Jan-2017 - 00:00  12-Jan-2017 - 00:00          38.33
60.71
11  12-Jan-2017 - 00:00  13-Jan-2017 - 00:00          42.32
60.71
12  13-Jan-2017 - 00:00  14-Jan-2017 - 00:00         103.51
60.71
13  14-Jan-2017 - 00:00  15-Jan-2017 - 00:00          33.96
60.71
14  15-Jan-2017 - 00:00  16-Jan-2017 - 00:00          54.93
60.71
15  16-Jan-2017 - 00:00  17-Jan-2017 - 00:00          46.01
60.71
16  17-Jan-2017 - 00:00  18-Jan-2017 - 00:00          47.76
60.71
17  18-Jan-2017 - 00:00  19-Jan-2017 - 00:00          36.59
60.71
18  19-Jan-2017 - 00:00  20-Jan-2017 - 00:00          42.82
60.71
19  20-Jan-2017 - 00:00  21-Jan-2017 - 00:00          68.26
60.71
20  21-Jan-2017 - 00:00  22-Jan-2017 - 00:00          38.96
60.71
21  22-Jan-2017 - 00:00  23-Jan-2017 - 00:00          26.28
60.71
```

22 23-Jan-2017 - 00:00 24-Jan-2017 - 00:00 33.59
60.71
23 24-Jan-2017 - 00:00 25-Jan-2017 - 00:00 29.59
60.71
24 25-Jan-2017 - 00:00 26-Jan-2017 - 00:00 40.49
60.71
25 26-Jan-2017 - 00:00 27-Jan-2017 - 00:00 29.95
60.71
26 27-Jan-2017 - 00:00 28-Jan-2017 - 00:00 18.93
60.71
27 28-Jan-2017 - 00:00 29-Jan-2017 - 00:00 7.64
60.71
28 29-Jan-2017 - 00:00 30-Jan-2017 - 00:00 8.65
60.71
29 30-Jan-2017 - 00:00 31-Jan-2017 - 00:00 19.69
60.71
30 31-Jan-2017 - 00:00 01-Feb-2017 - 00:00 35.80
60.71
31 01-Feb-2017 - 00:00 02-Feb-2017 - 00:00 37.48
60.71
32 02-Feb-2017 - 00:00 03-Feb-2017 - 00:00 34.84
60.71
33 03-Feb-2017 - 00:00 04-Feb-2017 - 00:00 36.89
60.71
34 04-Feb-2017 - 00:00 05-Feb-2017 - 00:00 40.92
60.71
35 05-Feb-2017 - 00:00 06-Feb-2017 - 00:00 43.07
60.71
36 06-Feb-2017 - 00:00 07-Feb-2017 - 00:00 42.05
60.71
37 07-Feb-2017 - 00:00 08-Feb-2017 - 00:00 47.71
60.71
38 08-Feb-2017 - 00:00 09-Feb-2017 - 00:00 61.60
60.71
39 09-Feb-2017 - 00:00 10-Feb-2017 - 00:00 54.49
60.71
40 10-Feb-2017 - 00:00 11-Feb-2017 - 00:00 50.29
60.71
41 11-Feb-2017 - 00:00 12-Feb-2017 - 00:00 44.20
60.71
42 12-Feb-2017 - 00:00 13-Feb-2017 - 00:00 26.51
60.71
43 13-Feb-2017 - 00:00 14-Feb-2017 - 00:00 17.22
60.71
44 14-Feb-2017 - 00:00 15-Feb-2017 - 00:00 22.04
60.71
45 15-Feb-2017 - 00:00 16-Feb-2017 - 00:00 30.00
60.71
46 16-Feb-2017 - 00:00 17-Feb-2017 - 00:00 36.92
60.71

```
47  17-Feb-2017 - 00:00  18-Feb-2017 - 00:00          38.03
60.71
48  18-Feb-2017 - 00:00  19-Feb-2017 - 00:00          29.63
60.71
49  19-Feb-2017 - 00:00  20-Feb-2017 - 00:00          34.19
60.71

    NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0         2.36         9.78      27.66         5.36         2.11
1.91
1         2.33         8.21      27.66         5.36         2.86
1.91
2        11.39        17.28      27.66         5.36         7.73
1.91
3         6.06        12.32      27.66         5.36         2.72
1.91
4         5.58        12.67      27.66         5.36         2.65
1.91
5         6.91        14.38      27.66         5.36         2.28
1.91
6         5.72         9.66      27.66         5.36         2.19
1.91
7         4.33        10.57      27.66         5.36         2.23
1.91
8         6.02        15.74      27.66         5.36         3.76
1.91
9         6.96        16.31      27.66         5.36         2.71
1.91
10        6.57        18.45      27.66         5.36         2.75
1.91
11       22.63        26.31      27.66         5.36         2.57
1.91
12       32.73        38.26      27.66         5.36         2.65
1.91
13       24.83        27.27      27.66         5.36         2.48
1.91
14       16.29        20.86      27.66         5.36         3.54
1.91
15        4.18        14.54      27.66         5.36         3.23
1.91
16        4.08        20.86      27.66         5.36         4.49
1.91
17        4.76        20.39      27.66         5.36         3.52
1.91
18        2.71        15.88      27.66         5.36         2.55
1.91
19        2.86        18.64      27.66         5.36         4.17
1.91
20        2.37        15.49      27.66         5.36         2.53
```

1.91					
21	2.47	15.81	27.66	5.36	2.99
1.91					
22	3.12	15.40	27.66	5.36	4.70
1.91					
23	2.83	14.93	27.66	5.36	3.77
1.91					
24	2.37	12.49	27.66	5.36	2.63
1.91					
25	2.44	12.68	27.66	5.36	2.60
1.91					
26	2.53	12.44	27.66	5.36	2.46
1.91					
27	2.59	11.63	27.66	5.36	2.35
1.91					
28	6.07	13.59	27.66	5.36	2.90
1.91					
29	6.07	18.00	27.66	5.36	2.77
1.91					
30	6.47	17.91	27.66	5.36	3.97
1.91					
31	3.63	16.19	27.66	5.36	2.27
1.91					
32	6.89	18.19	27.66	5.36	2.41
1.01					
33	6.19	18.27	27.66	5.36	3.09
1.21					
34	6.62	17.96	27.66	5.36	2.64
1.21					
35	5.41	18.31	27.66	5.36	2.90
1.21					
36	6.82	21.76	27.66	5.36	2.83
1.21					
37	9.01	25.18	27.66	5.36	4.03
1.21					
38	9.12	23.48	27.66	5.36	2.85
1.21					
39	5.63	22.22	27.66	5.36	2.97
1.21					
40	6.47	21.84	27.66	5.36	2.74
1.21					
41	10.70	22.33	27.66	5.36	2.32
1.21					
42	5.93	17.56	27.66	5.36	2.20
1.21					
43	5.46	15.74	27.66	5.36	2.54
1.21					
44	2.54	12.35	27.66	5.36	2.17
1.21					
45	3.77	14.66	27.66	5.36	2.57

1.21
46 3.26 14.88 27.66 5.36 2.20
1.21
47 5.55 18.64 27.66 5.36 2.10
1.21
48 5.58 17.27 27.66 5.36 1.99
1.21
49 5.52 18.13 27.66 5.36 3.60
1.16

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
0	24.63	0.52	2.95
1	20.49	0.13	2.01
2	13.04	0.45	3.52
3	19.42	0.65	3.98
4	25.89	0.60	3.53
5	25.16	0.74	3.80
6	35.32	0.36	3.20
7	31.22	0.34	1.94
8	24.32	0.51	3.27
9	21.71	0.55	3.88
10	24.67	0.74	3.79
11	30.78	0.74	3.52
12	30.94	6.60	4.35
13	44.53	0.23	0.77
14	45.14	1.00	2.82
15	50.70	0.44	1.41
16	52.13	0.55	2.52
17	50.51	0.88	2.55
18	63.89	0.00	1.10
19	62.39	0.26	2.05
20	55.96	0.00	1.59
21	62.37	0.00	2.37
22	54.17	0.42	2.69
23	65.30	0.31	2.15
24	71.66	0.00	0.84
25	51.52	0.00	0.77
26	24.48	0.00	1.33
27	16.77	0.00	1.06
28	15.22	0.21	1.67
29	21.93	0.41	2.19
30	23.68	1.05	3.39
31	33.67	0.06	2.74
32	27.56	0.51	2.93
33	33.36	0.54	3.29
34	38.83	0.47	2.02
35	32.93	0.65	0.78
36	32.11	0.84	0.54
37	35.07	1.32	1.34
38	40.37	1.50	2.12

```
39          39.35             1.32             1.89
40          32.65             1.34             1.90
41          29.16             1.34             1.90
42          34.39             1.41             1.96
43          34.28             1.90             2.33
44          52.83             2.23             2.59
45          49.97             1.94             2.37
46          67.63             1.67             2.16
47          47.98             1.34             1.90
48          39.44             1.34             1.90
49          32.65             1.39             1.95

df.info()
df8.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1841 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   NH3 (ug/m3)      306 non-null    float64
 8   SO2 (ug/m3)      1830 non-null   float64
 9   CO (mg/m3)       1687 non-null   float64
 10  Ozone (ug/m3)    1833 non-null   float64
 11  Benzene (ug/m3)  1859 non-null   float64
 12  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1885 non-null   float64
 3   PM10 (ug/m3)     1885 non-null   float64
 4   NO (ug/m3)       1885 non-null   float64
 5   NO2 (ug/m3)      1885 non-null   float64
 6   NOx (ppb)        1885 non-null   float64
 7   NH3 (ug/m3)      1885 non-null   float64
 8   SO2 (ug/m3)      1885 non-null   float64
 9   CO (mg/m3)       1885 non-null   float64
 10  Ozone (ug/m3)    1885 non-null   float64
```

```
 11  Benzene (ug/m3)  1885 non-null   float64
 12  Toluene (ug/m3)  1885 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB
```

As observed there are no null values in the dataset now (there are a total of 1885 nn values in all the colums) and this method is more effective than the previous as the null values right from the beginning of each column is imputed with some numerical.

Median imputation

Median imputation is a method in which the missing values are replaced with the median value of the entire feature column. When the data is skewed, it is good to consider using the median value for replacing the missing values.

```
df.head(50)
```

```
               From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0   01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1   02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2   03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3   04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4   05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN
5   06-Jan-2017 - 00:00  07-Jan-2017 - 00:00          64.52
NaN
6   07-Jan-2017 - 00:00  08-Jan-2017 - 00:00          45.01
NaN
7   08-Jan-2017 - 00:00  09-Jan-2017 - 00:00          32.71
NaN
8   09-Jan-2017 - 00:00  10-Jan-2017 - 00:00          33.66
NaN
9   10-Jan-2017 - 00:00  11-Jan-2017 - 00:00          34.82
NaN
10  11-Jan-2017 - 00:00  12-Jan-2017 - 00:00          38.33
NaN
11  12-Jan-2017 - 00:00  13-Jan-2017 - 00:00          42.32
NaN
12  13-Jan-2017 - 00:00  14-Jan-2017 - 00:00         103.51
NaN
13  14-Jan-2017 - 00:00  15-Jan-2017 - 00:00          33.96
NaN
14  15-Jan-2017 - 00:00  16-Jan-2017 - 00:00          54.93
NaN
```

15	16-Jan-2017 - 00:00	17-Jan-2017 - 00:00	46.01	NaN
16	17-Jan-2017 - 00:00	18-Jan-2017 - 00:00	47.76	NaN
17	18-Jan-2017 - 00:00	19-Jan-2017 - 00:00	36.59	NaN
18	19-Jan-2017 - 00:00	20-Jan-2017 - 00:00	42.82	NaN
19	20-Jan-2017 - 00:00	21-Jan-2017 - 00:00	68.26	NaN
20	21-Jan-2017 - 00:00	22-Jan-2017 - 00:00	38.96	NaN
21	22-Jan-2017 - 00:00	23-Jan-2017 - 00:00	26.28	NaN
22	23-Jan-2017 - 00:00	24-Jan-2017 - 00:00	33.59	NaN
23	24-Jan-2017 - 00:00	25-Jan-2017 - 00:00	29.59	NaN
24	25-Jan-2017 - 00:00	26-Jan-2017 - 00:00	40.49	NaN
25	26-Jan-2017 - 00:00	27-Jan-2017 - 00:00	29.95	NaN
26	27-Jan-2017 - 00:00	28-Jan-2017 - 00:00	18.93	NaN
27	28-Jan-2017 - 00:00	29-Jan-2017 - 00:00	7.64	NaN
28	29-Jan-2017 - 00:00	30-Jan-2017 - 00:00	8.65	NaN
29	30-Jan-2017 - 00:00	31-Jan-2017 - 00:00	19.69	NaN
30	31-Jan-2017 - 00:00	01-Feb-2017 - 00:00	35.80	NaN
31	01-Feb-2017 - 00:00	02-Feb-2017 - 00:00	37.48	NaN
32	02-Feb-2017 - 00:00	03-Feb-2017 - 00:00	34.84	NaN
33	03-Feb-2017 - 00:00	04-Feb-2017 - 00:00	36.89	NaN
34	04-Feb-2017 - 00:00	05-Feb-2017 - 00:00	40.92	NaN
35	05-Feb-2017 - 00:00	06-Feb-2017 - 00:00	43.07	NaN
36	06-Feb-2017 - 00:00	07-Feb-2017 - 00:00	42.05	NaN
37	07-Feb-2017 - 00:00	08-Feb-2017 - 00:00	47.71	NaN
38	08-Feb-2017 - 00:00	09-Feb-2017 - 00:00	61.60	NaN
39	09-Feb-2017 - 00:00	10-Feb-2017 - 00:00	54.49	NaN

```
40  10-Feb-2017 - 00:00  11-Feb-2017 - 00:00          50.29
NaN
41  11-Feb-2017 - 00:00  12-Feb-2017 - 00:00          44.20
NaN
42  12-Feb-2017 - 00:00  13-Feb-2017 - 00:00          26.51
NaN
43  13-Feb-2017 - 00:00  14-Feb-2017 - 00:00          17.22
NaN
44  14-Feb-2017 - 00:00  15-Feb-2017 - 00:00          22.04
NaN
45  15-Feb-2017 - 00:00  16-Feb-2017 - 00:00          30.00
NaN
46  16-Feb-2017 - 00:00  17-Feb-2017 - 00:00          36.92
NaN
47  17-Feb-2017 - 00:00  18-Feb-2017 - 00:00          38.03
NaN
48  18-Feb-2017 - 00:00  19-Feb-2017 - 00:00          29.63
NaN
49  19-Feb-2017 - 00:00  20-Feb-2017 - 00:00          34.19
NaN

    NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0         2.36         9.78        NaN          NaN         2.11
NaN
1         2.33         8.21        NaN          NaN         2.86
NaN
2        11.39        17.28        NaN          NaN         7.73
NaN
3         6.06        12.32        NaN          NaN         2.72
NaN
4         5.58        12.67        NaN          NaN         2.65
NaN
5         6.91        14.38        NaN          NaN         2.28
NaN
6         5.72         9.66        NaN          NaN         2.19
NaN
7         4.33        10.57        NaN          NaN         2.23
NaN
8         6.02        15.74        NaN          NaN         3.76
NaN
9         6.96        16.31        NaN          NaN         2.71
NaN
10        6.57        18.45        NaN          NaN         2.75
NaN
11       22.63        26.31        NaN          NaN         2.57
NaN
12       32.73        38.26        NaN          NaN         2.65
NaN
13       24.83        27.27        NaN          NaN         2.48
```

NaN
14 16.29 20.86 NaN NaN 3.54
NaN
15 4.18 14.54 NaN NaN 3.23
NaN
16 4.08 20.86 NaN NaN 4.49
NaN
17 4.76 20.39 NaN NaN 3.52
NaN
18 2.71 15.88 NaN NaN 2.55
NaN
19 2.86 18.64 NaN NaN 4.17
NaN
20 2.37 15.49 NaN NaN 2.53
NaN
21 2.47 15.81 NaN NaN 2.99
NaN
22 3.12 15.40 NaN NaN 4.70
NaN
23 2.83 14.93 NaN NaN 3.77
NaN
24 2.37 12.49 NaN NaN 2.63
NaN
25 2.44 12.68 NaN NaN 2.60
NaN
26 2.53 12.44 NaN NaN 2.46
NaN
27 2.59 11.63 NaN NaN 2.35
NaN
28 6.07 13.59 NaN NaN 2.90
NaN
29 6.07 18.00 NaN NaN 2.77
NaN
30 6.47 17.91 NaN NaN 3.97
NaN
31 3.63 16.19 NaN NaN 2.27
1.91
32 6.89 18.19 NaN NaN 2.41
1.01
33 6.19 18.27 NaN NaN 3.09
NaN
34 6.62 17.96 NaN NaN 2.64
NaN
35 5.41 18.31 NaN NaN 2.90
NaN
36 6.82 21.76 NaN NaN 2.83
NaN
37 9.01 25.18 NaN NaN 4.03
NaN
38 9.12 23.48 NaN NaN 2.85

NaN
39 5.63 22.22 NaN NaN 2.97
NaN
40 6.47 21.84 NaN NaN 2.74
NaN
41 10.70 22.33 NaN NaN 2.32
NaN
42 5.93 17.56 NaN NaN 2.20
NaN
43 5.46 15.74 NaN NaN 2.54
NaN
44 2.54 12.35 NaN NaN 2.17
NaN
45 3.77 14.66 NaN NaN 2.57
NaN
46 3.26 14.88 NaN NaN 2.20
NaN
47 5.55 18.64 NaN NaN 2.10
1.21
48 5.58 17.27 NaN NaN 1.99
1.21
49 5.52 18.13 NaN NaN 3.60
NaN

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
0	24.63	0.52	2.95
1	20.49	0.13	2.01
2	13.04	0.45	3.52
3	19.42	0.65	3.98
4	25.89	0.60	3.53
5	25.16	0.74	3.80
6	35.32	0.36	3.20
7	31.22	0.34	1.94
8	24.32	0.51	3.27
9	21.71	0.55	3.88
10	24.67	0.74	3.79
11	30.78	0.74	3.52
12	30.94	6.60	4.35
13	44.53	0.23	0.77
14	45.14	1.00	2.82
15	50.70	0.44	1.41
16	52.13	0.55	2.52
17	50.51	0.88	2.55
18	63.89	0.00	1.10
19	62.39	0.26	2.05
20	55.96	0.00	1.59
21	62.37	0.00	2.37
22	54.17	0.42	2.69
23	65.30	0.31	2.15
24	71.66	0.00	0.84

```
25          51.52             0.00             0.77
26          24.48             0.00             1.33
27          16.77             0.00             1.06
28          15.22             0.21             1.67
29          21.93             0.41             2.19
30          23.68             1.05             3.39
31          33.67             0.06             2.74
32          27.56             0.51             2.93
33          33.36             0.54             3.29
34          38.83             0.47             2.02
35          32.93             0.65             0.78
36          32.11             0.84             0.54
37          35.07             1.32             1.34
38          40.37             1.50             2.12
39          39.35             1.32             1.89
40          32.65             1.34             1.90
41          29.16             1.34             1.90
42          34.39             1.41             1.96
43          34.28             1.90             2.33
44          52.83             2.23             2.59
45          49.97             1.94             2.37
46          67.63             1.67             2.16
47          47.98             1.34             1.90
48          39.44             1.34             1.90
49          32.65             1.39             1.95
```

```
df9 = pd.read_excel (r'/content/Data 2017-2022.xlsx')
c=['PM2.5 (ug/m3)','PM10 (ug/m3)','NO (ug/m3)','NO2 (ug/m3)','NOx
(ppb)','NH3 (ug/m3)','SO2 (ug/m3)','CO (mg/m3)','Ozone
(ug/m3)','Benzene (ug/m3)','Toluene (ug/m3)']
df9[c] = df9[c].fillna(df9[c].median())
df9.head(50) #all the missing values in the columns are replaced withe
the median values of the same column
```

```
              From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0   01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
67.25
1   02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
67.25
2   03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
67.25
3   04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
67.25
4   05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
67.25
5   06-Jan-2017 - 00:00  07-Jan-2017 - 00:00          64.52
67.25
6   07-Jan-2017 - 00:00  08-Jan-2017 - 00:00          45.01
67.25
7   08-Jan-2017 - 00:00  09-Jan-2017 - 00:00          32.71
```

67.25
8 09-Jan-2017 - 00:00 10-Jan-2017 - 00:00 33.66
67.25
9 10-Jan-2017 - 00:00 11-Jan-2017 - 00:00 34.82
67.25
10 11-Jan-2017 - 00:00 12-Jan-2017 - 00:00 38.33
67.25
11 12-Jan-2017 - 00:00 13-Jan-2017 - 00:00 42.32
67.25
12 13-Jan-2017 - 00:00 14-Jan-2017 - 00:00 103.51
67.25
13 14-Jan-2017 - 00:00 15-Jan-2017 - 00:00 33.96
67.25
14 15-Jan-2017 - 00:00 16-Jan-2017 - 00:00 54.93
67.25
15 16-Jan-2017 - 00:00 17-Jan-2017 - 00:00 46.01
67.25
16 17-Jan-2017 - 00:00 18-Jan-2017 - 00:00 47.76
67.25
17 18-Jan-2017 - 00:00 19-Jan-2017 - 00:00 36.59
67.25
18 19-Jan-2017 - 00:00 20-Jan-2017 - 00:00 42.82
67.25
19 20-Jan-2017 - 00:00 21-Jan-2017 - 00:00 68.26
67.25
20 21-Jan-2017 - 00:00 22-Jan-2017 - 00:00 38.96
67.25
21 22-Jan-2017 - 00:00 23-Jan-2017 - 00:00 26.28
67.25
22 23-Jan-2017 - 00:00 24-Jan-2017 - 00:00 33.59
67.25
23 24-Jan-2017 - 00:00 25-Jan-2017 - 00:00 29.59
67.25
24 25-Jan-2017 - 00:00 26-Jan-2017 - 00:00 40.49
67.25
25 26-Jan-2017 - 00:00 27-Jan-2017 - 00:00 29.95
67.25
26 27-Jan-2017 - 00:00 28-Jan-2017 - 00:00 18.93
67.25
27 28-Jan-2017 - 00:00 29-Jan-2017 - 00:00 7.64
67.25
28 29-Jan-2017 - 00:00 30-Jan-2017 - 00:00 8.65
67.25
29 30-Jan-2017 - 00:00 31-Jan-2017 - 00:00 19.69
67.25
30 31-Jan-2017 - 00:00 01-Feb-2017 - 00:00 35.80
67.25
31 01-Feb-2017 - 00:00 02-Feb-2017 - 00:00 37.48
67.25
32 02-Feb-2017 - 00:00 03-Feb-2017 - 00:00 34.84

67.25
33 03-Feb-2017 - 00:00 04-Feb-2017 - 00:00   36.89
67.25
34 04-Feb-2017 - 00:00 05-Feb-2017 - 00:00   40.92
67.25
35 05-Feb-2017 - 00:00 06-Feb-2017 - 00:00   43.07
67.25
36 06-Feb-2017 - 00:00 07-Feb-2017 - 00:00   42.05
67.25
37 07-Feb-2017 - 00:00 08-Feb-2017 - 00:00   47.71
67.25
38 08-Feb-2017 - 00:00 09-Feb-2017 - 00:00   61.60
67.25
39 09-Feb-2017 - 00:00 10-Feb-2017 - 00:00   54.49
67.25
40 10-Feb-2017 - 00:00 11-Feb-2017 - 00:00   50.29
67.25
41 11-Feb-2017 - 00:00 12-Feb-2017 - 00:00   44.20
67.25
42 12-Feb-2017 - 00:00 13-Feb-2017 - 00:00   26.51
67.25
43 13-Feb-2017 - 00:00 14-Feb-2017 - 00:00   17.22
67.25
44 14-Feb-2017 - 00:00 15-Feb-2017 - 00:00   22.04
67.25
45 15-Feb-2017 - 00:00 16-Feb-2017 - 00:00   30.00
67.25
46 16-Feb-2017 - 00:00 17-Feb-2017 - 00:00   36.92
67.25
47 17-Feb-2017 - 00:00 18-Feb-2017 - 00:00   38.03
67.25
48 18-Feb-2017 - 00:00 19-Feb-2017 - 00:00   29.63
67.25
49 19-Feb-2017 - 00:00 20-Feb-2017 - 00:00   34.19
67.25

    NO (ug/m3) NO2 (ug/m3) NOx (ppb) NH3 (ug/m3) SO2 (ug/m3) CO
(mg/m3) \
0   2.36   9.78   14.615   14.455   2.11
0.74
1   2.33   8.21   14.615   14.455   2.86
0.74
2   11.39   17.28   14.615   14.455   7.73
0.74
3   6.06   12.32   14.615   14.455   2.72
0.74
4   5.58   12.67   14.615   14.455   2.65
0.74
5   6.91   14.38   14.615   14.455   2.28
0.74

6	5.72	9.66	14.615	14.455	2.19
0.74					
7	4.33	10.57	14.615	14.455	2.23
0.74					
8	6.02	15.74	14.615	14.455	3.76
0.74					
9	6.96	16.31	14.615	14.455	2.71
0.74					
10	6.57	18.45	14.615	14.455	2.75
0.74					
11	22.63	26.31	14.615	14.455	2.57
0.74					
12	32.73	38.26	14.615	14.455	2.65
0.74					
13	24.83	27.27	14.615	14.455	2.48
0.74					
14	16.29	20.86	14.615	14.455	3.54
0.74					
15	4.18	14.54	14.615	14.455	3.23
0.74					
16	4.08	20.86	14.615	14.455	4.49
0.74					
17	4.76	20.39	14.615	14.455	3.52
0.74					
18	2.71	15.88	14.615	14.455	2.55
0.74					
19	2.86	18.64	14.615	14.455	4.17
0.74					
20	2.37	15.49	14.615	14.455	2.53
0.74					
21	2.47	15.81	14.615	14.455	2.99
0.74					
22	3.12	15.40	14.615	14.455	4.70
0.74					
23	2.83	14.93	14.615	14.455	3.77
0.74					
24	2.37	12.49	14.615	14.455	2.63
0.74					
25	2.44	12.68	14.615	14.455	2.60
0.74					
26	2.53	12.44	14.615	14.455	2.46
0.74					
27	2.59	11.63	14.615	14.455	2.35
0.74					
28	6.07	13.59	14.615	14.455	2.90
0.74					
29	6.07	18.00	14.615	14.455	2.77
0.74					
30	6.47	17.91	14.615	14.455	3.97
0.74					

31	3.63	16.19	14.615	14.455	2.27	1.91
32	6.89	18.19	14.615	14.455	2.41	1.01
33	6.19	18.27	14.615	14.455	3.09	0.74
34	6.62	17.96	14.615	14.455	2.64	0.74
35	5.41	18.31	14.615	14.455	2.90	0.74
36	6.82	21.76	14.615	14.455	2.83	0.74
37	9.01	25.18	14.615	14.455	4.03	0.74
38	9.12	23.48	14.615	14.455	2.85	0.74
39	5.63	22.22	14.615	14.455	2.97	0.74
40	6.47	21.84	14.615	14.455	2.74	0.74
41	10.70	22.33	14.615	14.455	2.32	0.74
42	5.93	17.56	14.615	14.455	2.20	0.74
43	5.46	15.74	14.615	14.455	2.54	0.74
44	2.54	12.35	14.615	14.455	2.17	0.74
45	3.77	14.66	14.615	14.455	2.57	0.74
46	3.26	14.88	14.615	14.455	2.20	0.74
47	5.55	18.64	14.615	14.455	2.10	1.21
48	5.58	17.27	14.615	14.455	1.99	1.21
49	5.52	18.13	14.615	14.455	3.60	0.74

	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
0	24.63	0.52	2.95
1	20.49	0.13	2.01
2	13.04	0.45	3.52
3	19.42	0.65	3.98
4	25.89	0.60	3.53
5	25.16	0.74	3.80
6	35.32	0.36	3.20
7	31.22	0.34	1.94
8	24.32	0.51	3.27
9	21.71	0.55	3.88

```
10          24.67             0.74             3.79
11          30.78             0.74             3.52
12          30.94             6.60             4.35
13          44.53             0.23             0.77
14          45.14             1.00             2.82
15          50.70             0.44             1.41
16          52.13             0.55             2.52
17          50.51             0.88             2.55
18          63.89             0.00             1.10
19          62.39             0.26             2.05
20          55.96             0.00             1.59
21          62.37             0.00             2.37
22          54.17             0.42             2.69
23          65.30             0.31             2.15
24          71.66             0.00             0.84
25          51.52             0.00             0.77
26          24.48             0.00             1.33
27          16.77             0.00             1.06
28          15.22             0.21             1.67
29          21.93             0.41             2.19
30          23.68             1.05             3.39
31          33.67             0.06             2.74
32          27.56             0.51             2.93
33          33.36             0.54             3.29
34          38.83             0.47             2.02
35          32.93             0.65             0.78
36          32.11             0.84             0.54
37          35.07             1.32             1.34
38          40.37             1.50             2.12
39          39.35             1.32             1.89
40          32.65             1.34             1.90
41          29.16             1.34             1.90
42          34.39             1.41             1.96
43          34.28             1.90             2.33
44          52.83             2.23             2.59
45          49.97             1.94             2.37
46          67.63             1.67             2.16
47          47.98             1.34             1.90
48          39.44             1.34             1.90
49          32.65             1.39             1.95
```

```
df.info()
df9.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
```

```
 2   PM2.5 (ug/m3)    1841 non-null   float64
 3   PM10 (ug/m3)     306 non-null    float64
 4   NO (ug/m3)       1844 non-null   float64
 5   NO2 (ug/m3)      1843 non-null   float64
 6   NOx (ppb)        1598 non-null   float64
 7   NH3 (ug/m3)      306 non-null    float64
 8   SO2 (ug/m3)      1830 non-null   float64
 9   CO (mg/m3)       1687 non-null   float64
 10  Ozone (ug/m3)    1833 non-null   float64
 11  Benzene (ug/m3)  1859 non-null   float64
 12  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 13 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   From Date        1885 non-null   object
 1   To Date          1885 non-null   object
 2   PM2.5 (ug/m3)    1885 non-null   float64
 3   PM10 (ug/m3)     1885 non-null   float64
 4   NO (ug/m3)       1885 non-null   float64
 5   NO2 (ug/m3)      1885 non-null   float64
 6   NOx (ppb)        1885 non-null   float64
 7   NH3 (ug/m3)      1885 non-null   float64
 8   SO2 (ug/m3)      1885 non-null   float64
 9   CO (mg/m3)       1885 non-null   float64
 10  Ozone (ug/m3)    1885 non-null   float64
 11  Benzene (ug/m3)  1885 non-null   float64
 12  Toluene (ug/m3)  1885 non-null   float64
dtypes: float64(11), object(2)
memory usage: 191.6+ KB

df9.describe() #this method is more precise than the mean imputation

       PM2.5 (ug/m3)  PM10 (ug/m3)   NO (ug/m3)  NO2 (ug/m3)    NOx
(ppb)  \
count    1885.000000   1885.000000  1885.000000  1885.000000
1885.000000
mean       30.425263     65.772875     6.920477    12.528886
15.811684
std        20.057418     10.959726     5.553900     8.611886
8.115541
min         0.410000     21.600000     0.010000     0.020000
0.000000
25%        16.850000     67.250000     3.330000     6.630000
10.530000
50%        27.280000     67.250000     5.460000    11.280000
14.615000
75%        39.010000     67.250000     9.210000    16.310000
```

```
19.280000
max       278.970000    371.610000    98.620000   134.760000
106.740000

       NH3 (ug/m3)  SO2 (ug/m3)   CO (mg/m3)  Ozone (ug/m3)  Benzene
(ug/m3)  \
count  1885.000000  1885.000000  1885.000000    1885.000000
1885.000000
mean     14.100618     6.502992     0.818599      27.500546
0.574976
std       2.376704     4.991580     1.392886      16.595455
2.214523
min       5.360000     0.090000     0.000000       0.100000
0.000000
25%      14.455000     3.920000     0.620000      16.930000
0.000000
50%      14.455000     4.880000     0.740000      24.680000
0.000000
75%      14.455000     7.070000     0.880000      34.180000
0.340000
max      33.680000    37.180000    48.020000     183.990000
46.230000

       Toluene (ug/m3)
count      1885.000000
mean          1.916032
std           4.354631
min           0.000000
25%           0.000000
50%           0.190000
75%           2.730000
max         121.150000
```

OUTLIER ANALYSIS

An outlier is an observation that appears to deviate markedly from other observations in the sample. Otherwise, Outliers are extreme values that stand out greatly from the overall pattern of values in a dataset or graph.

Outlier analysis is critical in analyzing the data for at least two reasons:

> The outliers may negatively bias the entire outcome of an analysis. Detect points that are considered “abnormal,” or which don't fit a particular pattern.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
```

```
df17 = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')
df17.head()
```

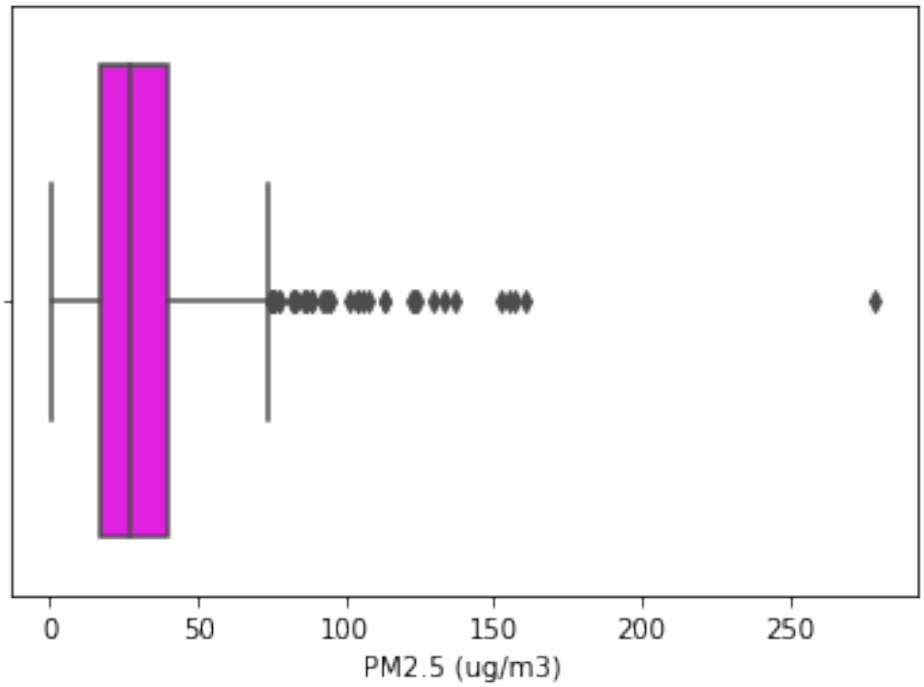
```
             From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0  01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1  02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2  03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3  04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4  05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN

   NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0        2.36         9.78        NaN          NaN         2.11
NaN
1        2.33         8.21        NaN          NaN         2.86
NaN
2       11.39        17.28        NaN          NaN         7.73
NaN
3        6.06        12.32        NaN          NaN         2.72
NaN
4        5.58        12.67        NaN          NaN         2.65
NaN

   Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0          24.63             0.52             2.95
1          20.49             0.13             2.01
2          13.04             0.45             3.52
3          19.42             0.65             3.98
4          25.89             0.60             3.53
```

UNIVARIATE VISUALIZATION

1. Box plot
2. Histogram
3. Scatter plot

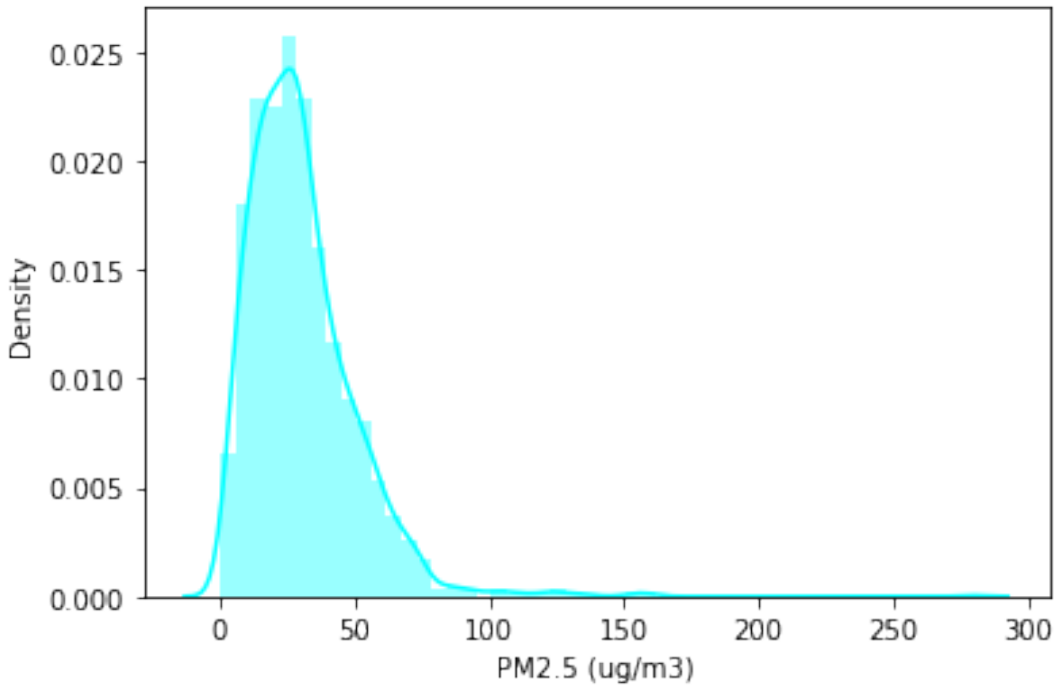
```
warnings.filterwarnings('ignore')
sns.boxplot(df17['PM2.5 (ug/m3)'],color='magenta')

# we can clearly see that data points above 65 are considered to be
outliers in PM2.5 column
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad83a6dd90>
```



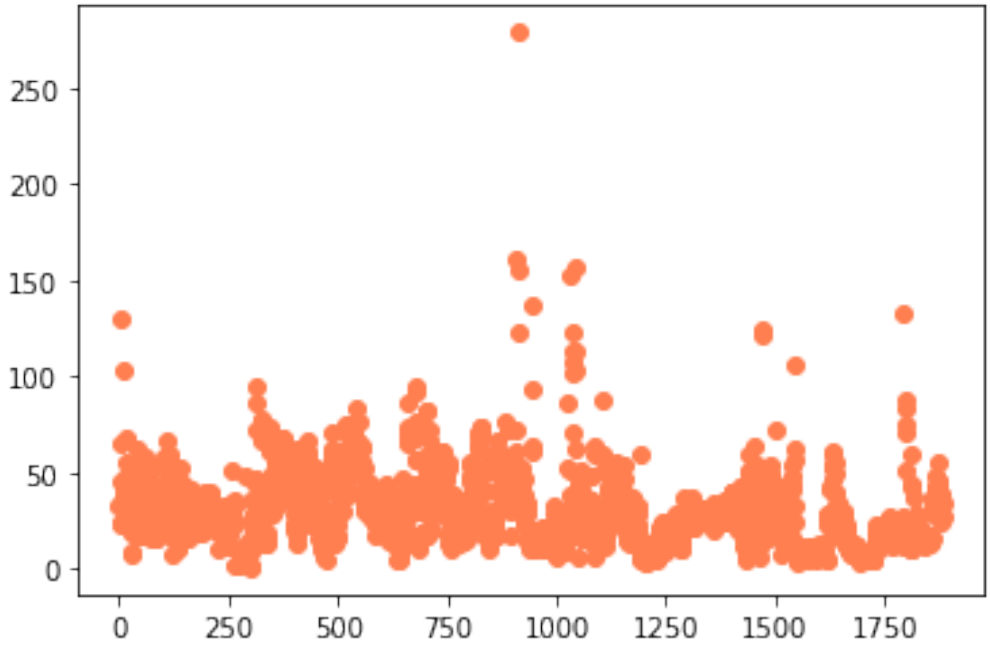

```
warnings.filterwarnings('ignore')
sns.distplot(df17['PM2.5 (ug/m3)'],color='cyan')

#This plots helps us to identify the nature of distribution. The PM
2.5 is highly skewed and diviates from mean.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad82d0b110>
```



```
plt.scatter(df17.index,df17['PM2.5 (ug/m3)'],c='coral')
```

```
<matplotlib.collections.PathCollection at 0x7fad82d381d0>
```

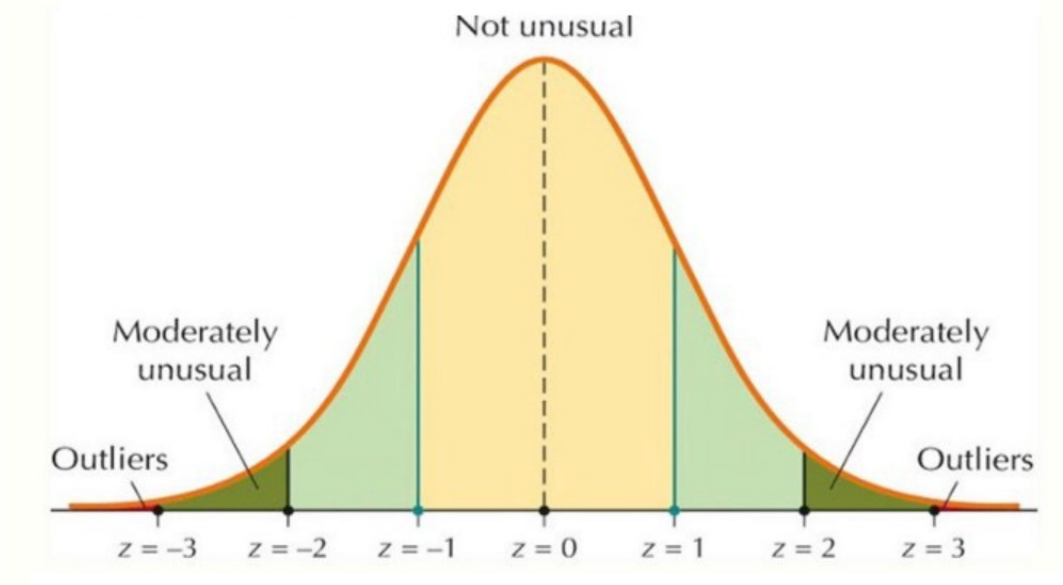


METHOD 1

Z-score treatment

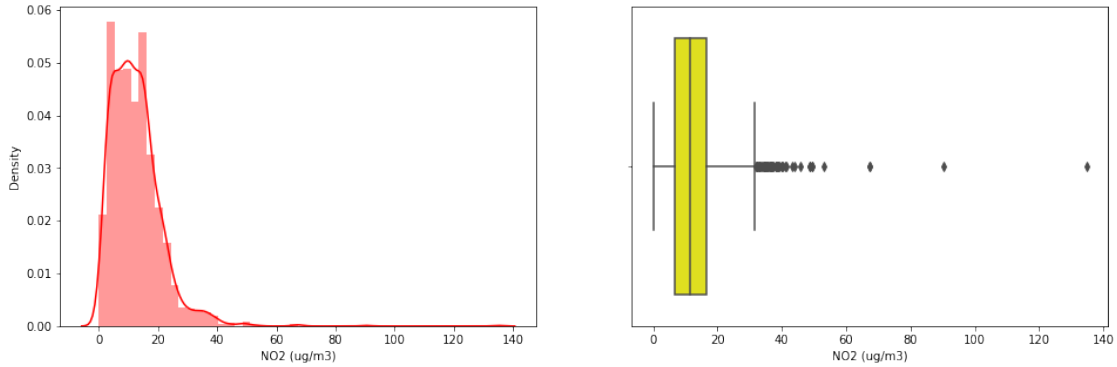
Assumption: Data normally distributed

If the z score of a data point is more than 3, it indicates that the data point is quite different from the other data points. Such a data point can be a outlier.



```
#Column used for Z scroe outlier detection: NO2
df10 = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')

plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(df10['NO2 (ug/m3)'],color='red')
plt.subplot(1,2,2)
sns.boxplot(df10['NO2 (ug/m3)'],color='yellow')
#this is the histogram and box plot before oultier detection and
handling. The data is positively skewed and contains ouliers.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad82b4b590>
```




```
#any value above the upper limit and below lower limit are considered
to be outliers.
#Upper Limit=mean+ 3*standard deviation
#Lower Limit=Mean -3*standard deviation

upper_limit = df10['NO2 (ug/m3)'].mean() + 3*df10['NO2 (ug/m3)'].std()
lower_limit = df10['NO2 (ug/m3)'].mean() - 3*df10['NO2 (ug/m3)'].std()
print(upper_limit,lower_limit)
```

```
38.679623230940415 -13.564929796322787
```

```
#rows with outliers in NO2 column
df10[(df10['NO2 (ug/m3)'] > upper_limit) | (df10['NO2 (ug/m3)'] <
lower_limit)]
```

```
                From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
180   30-Jun-2017 - 00:00  01-Jul-2017 - 00:00          31.75
NaN
249   07-Sep-2017 - 00:00  08-Sep-2017 - 00:00          19.21
NaN
250   08-Sep-2017 - 00:00  09-Sep-2017 - 00:00          22.40
NaN
318   15-Nov-2017 - 00:00  16-Nov-2017 - 00:00          37.34
NaN
341   08-Dec-2017 - 00:00  09-Dec-2017 - 00:00          40.15
```

NaN
450   27-Mar-2018 - 00:00 28-Mar-2018 - 00:00   36.68
NaN
451   28-Mar-2018 - 00:00 29-Mar-2018 - 00:00   26.03
NaN
452   29-Mar-2018 - 00:00 30-Mar-2018 - 00:00   26.92
NaN
485   01-May-2018 - 00:00 02-May-2018 - 00:00   60.14
NaN
486   02-May-2018 - 00:00 03-May-2018 - 00:00   70.31
NaN
489   05-May-2018 - 00:00 06-May-2018 - 00:00   41.37
NaN
490   06-May-2018 - 00:00 07-May-2018 - 00:00   41.88
NaN
491   07-May-2018 - 00:00 08-May-2018 - 00:00   50.53
NaN
492   08-May-2018 - 00:00 09-May-2018 - 00:00   24.55
NaN
493   09-May-2018 - 00:00 10-May-2018 - 00:00   13.16
NaN
907   27-Jun-2019 - 00:00 28-Jun-2019 - 00:00   45.49
NaN
940   30-Jul-2019 - 00:00 31-Jul-2019 - 00:00   14.59
NaN
944   03-Aug-2019 - 00:00 04-Aug-2019 - 00:00   93.28
NaN
1031 29-Oct-2019 - 00:00 30-Oct-2019 - 00:00   19.56
NaN

      NO (ug/m3) NO2 (ug/m3) NOx (ppb) NH3 (ug/m3) SO2 (ug/m3) \
180   27.48   67.25   NaN   NaN   2.89
249   16.52   67.18   NaN   NaN   3.19
250   7.39   39.90   NaN   NaN   3.54
318   7.71   39.89   24.58   NaN   19.52
341   8.72   40.23   25.51   NaN   12.88
450   4.77   38.92   16.36   NaN   5.31
451   7.38   43.96   20.62   NaN   4.19
452   4.10   41.06   17.44   NaN   4.44
485   27.83   134.76   85.59   NaN   3.61
486   25.57   90.25   62.20   NaN   4.42
489   27.44   49.53   43.73   NaN   4.21
490   29.45   53.20   47.02   NaN   4.70
491   26.82   48.70   42.86   NaN   32.59
492   25.54   45.68   40.43   NaN   7.09
493   27.68   48.75   43.53   NaN   6.81
907   10.31   41.40   51.68   NaN   3.78
940   8.77   43.29   52.05   NaN   4.23
944   9.13   39.21   48.27   NaN   4.35
1031   71.75   49.36   106.74   NaN   10.83

```
      CO (mg/m3)  Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
180          NaN          27.53             1.11             3.06
249         2.09          22.56             0.05             2.47
250         2.06          14.14             0.03             1.93
318          NaN          23.73             0.42             3.51
341         0.00          46.45             2.27             5.31
450         0.88           8.53             1.52             1.58
451         1.05           8.64             0.85             0.68
452         0.60          16.80             0.52             0.03
485         0.74          42.12             0.74             1.18
486         0.66          57.81             1.36             2.22
489         0.72          24.28             1.56             2.49
490         0.57          30.71             1.45             2.35
491         0.61          45.60             1.38             2.26
492         0.71          23.26             1.49             2.40
493         1.21          30.99             1.79             2.79
907         0.84          19.44             0.00             6.53
940         0.79          38.56             1.74             3.30
944         0.74          30.13             2.37             5.00
1031        0.57          26.30             0.00             2.04
```

```
#Rows excluding outliers values in NO2 column
new_df = df10[(df10['NO2 (ug/m3)'] < upper_limit) & (df10['NO2
(ug/m3)'] > lower_limit)]
new_df.head()
```

```
             From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0  01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1  02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2  03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3  04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4  05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN

   NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  CO
(mg/m3)  \
0        2.36         9.78        NaN          NaN         2.11
NaN
1        2.33         8.21        NaN          NaN         2.86
NaN
2       11.39        17.28        NaN          NaN         7.73
NaN
3        6.06        12.32        NaN          NaN         2.72
NaN
```

```
4        5.58        12.67        NaN          NaN         2.65
NaN
```

```
   Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0          24.63             0.52             2.95
1          20.49             0.13             2.01
2          13.04             0.45             3.52
3          19.42             0.65             3.98
4          25.89             0.60             3.53
```

```
#CAPPING --> In this technique, we cap our outliers data and make the
limit (we replace the oultier values with upper limit and lower limit
respectively)
new_df_cap = df10.copy()
new_df_cap['NO2 (ug/m3)'] = np.where( new_df_cap['NO2 (ug/m3)'] >
upper_limit,upper_limit,
                                     np.where(new_df_cap['NO2
(ug/m3)'] < lower_limit, lower_limit,new_df_cap['NO2 (ug/m3)']))

df10['NO2 (ug/m3)'].describe()
```

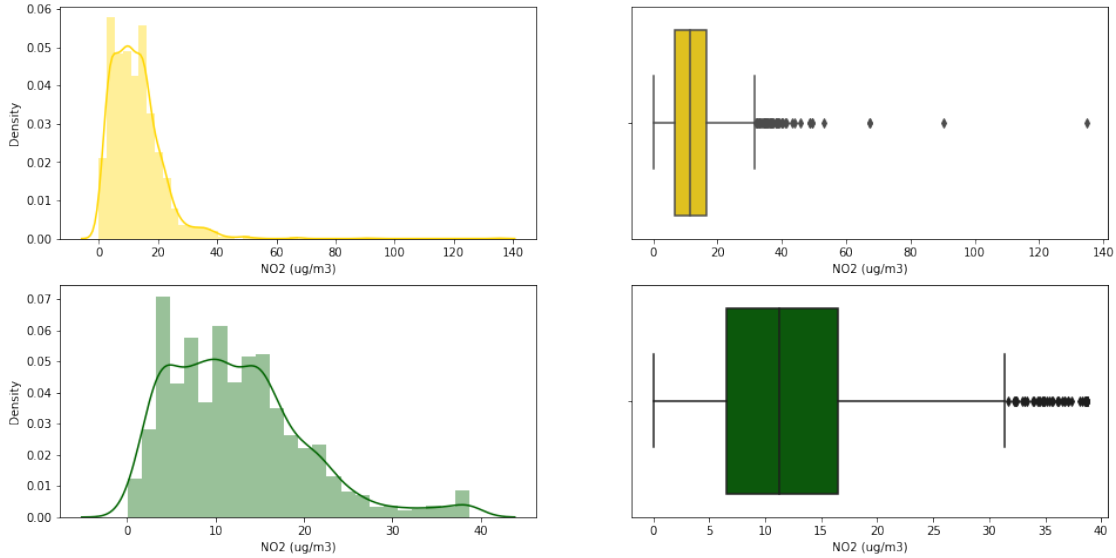
```
count    1843.000000
mean       12.557347
std         8.707426
min         0.020000
25%         6.510000
50%        11.280000
75%        16.470000
max       134.760000
Name: NO2 (ug/m3), dtype: float64
```

```
new_df_cap['NO2 (ug/m3)'].describe()
#It is evident the maximum value has come nearer to the mean of the
data. and there are slight variations in statistical parameters after
handling outliers.
```

```
count    1843.000000
mean       12.401293
std         7.738514
min         0.020000
25%         6.510000
50%        11.280000
75%        16.470000
max        38.679623
Name: NO2 (ug/m3), dtype: float64
```

```
import warnings
warnings.filterwarnings('ignore')
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df10['NO2 (ug/m3)'],color='gold' )
```

```
plt.subplot(2,2,2)
sns.boxplot(df10['NO2 (ug/m3)'],color='gold' )
plt.subplot(2,2,3)
sns.distplot(new_df_cap['NO2 (ug/m3)'],color='darkgreen')
plt.subplot(2,2,4)
sns.boxplot(new_df_cap['NO2 (ug/m3)'],color='darkgreen')
plt.show()

# from histogram we can see that after handling outliers using capping
technique, the data points have become normally distributed.
# The box plot before and after removal of ouliers, the range of
values have reduced  from 0-140 to 0-40.
```



METHOD 2

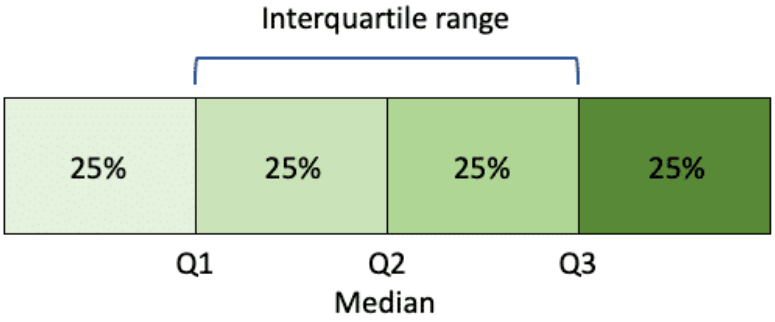
IQR based filtering

For Skewed data distribution.

Ouliers are:

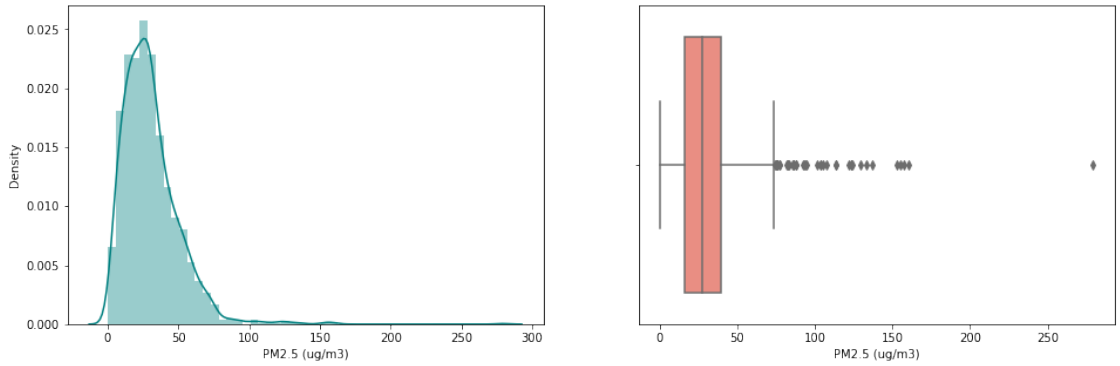
greater than 75th percentile + 1.5 IQR

less than the 25th percentile – 1.5 IQR




```
df11 = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')

plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(df11['PM2.5 (ug/m3)'],color='teal')
plt.subplot(1,2,2)
sns.boxplot(df11['PM2.5 (ug/m3)'],color='salmon')
# The data is positively skewed and above 80 all are considered to be
ouliers in PM 2.5 column.
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad80564f90>
```




```
#Inter quatile range is an statistical parameter which helps in
identifying the outliers in the dataset.
# Q1+1.5*IQR and Q3-1.5*IQR are considered as outlier in this method,
where Q1 and Q2 are lower quartile (25%) and upper quartile(75%).

Q1 = df11['PM2.5 (ug/m3)'].quantile(0.25)
Q3 = df11['PM2.5 (ug/m3)'].quantile(0.75)
IQR=Q3-Q1

upper_limit = Q1 + 1.5 * IQR
lower_limit = Q3 - 1.5 * IQR
print(upper_limit)
print(lower_limit)
```

```
51.025
5.045000000000002

#rows with outliers in PM 2.5 column
df11[(df11['PM2.5 (ug/m3)'] > upper_limit) | (df11['PM2.5 (ug/m3)'] <
lower_limit)]

                From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
4     05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN
5     06-Jan-2017 - 00:00  07-Jan-2017 - 00:00          64.52
NaN
12    13-Jan-2017 - 00:00  14-Jan-2017 - 00:00         103.51
NaN
14    15-Jan-2017 - 00:00  16-Jan-2017 - 00:00          54.93
NaN
19    20-Jan-2017 - 00:00  21-Jan-2017 - 00:00          68.26
NaN
...                   ...                   ...            ...
...
1798  04-Dec-2021 - 00:00  05-Dec-2021 - 00:00          82.82
24.72
1799  05-Dec-2021 - 00:00  06-Dec-2021 - 00:00          88.01
24.56
1801  07-Dec-2021 - 00:00  08-Dec-2021 - 00:00          74.48
82.86
1812  18-Dec-2021 - 00:00  19-Dec-2021 - 00:00          59.56
43.90
1876  20-Feb-2022 - 00:00  21-Feb-2022 - 00:00          55.46
100.33

      NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  \
4           5.58        12.67        NaN          NaN         2.65
5           6.91        14.38        NaN          NaN         2.28
12         32.73        38.26        NaN          NaN         2.65
14         16.29        20.86        NaN          NaN         3.54
19          2.86        18.64        NaN          NaN         4.17
...          ...          ...        ...          ...          ...
1798       13.23        14.48      27.72        18.35         9.35
1799       14.49        14.46      28.94        18.53         9.39
1801       15.32        14.44      29.76        18.71         9.41
1812       12.05        14.57      23.76        17.74         9.38
1876        4.36        15.48      19.85        17.17        19.95

      CO (mg/m3)  Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
4            NaN          25.89             0.60             3.53
5            NaN          25.16             0.74             3.80
12           NaN          30.94             6.60             4.35
14           NaN          45.14             1.00             2.82
```

```
19           NaN          62.39             0.26             2.05
...          ...            ...              ...              ...
1798        0.82          39.26             0.00             0.00
1799        0.82          23.70             0.00             0.00
1801        0.87           9.26             0.00             0.00
1812        0.87          31.00             0.00             0.00
1876        0.94         100.45             0.00             0.00

[279 rows x 13 columns]
```

```
#dataset excluding the rows which have outliers in PM 2.5 column
df11[(df11['PM2.5 (ug/m3)'] < upper_limit) & (df11['PM2.5 (ug/m3)'] >
lower_limit)]
```

```
                From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0     01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1     02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
NaN
2     03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3     04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
6     07-Jan-2017 - 00:00  08-Jan-2017 - 00:00          45.01
NaN
...                   ...                  ...            ...
...
1880  24-Feb-2022 - 00:00  25-Feb-2022 - 00:00          32.05
64.31
1881  25-Feb-2022 - 00:00  26-Feb-2022 - 00:00          38.95
74.92
1882  26-Feb-2022 - 00:00  27-Feb-2022 - 00:00          38.40
74.07
1883  27-Feb-2022 - 00:00  28-Feb-2022 - 00:00          27.51
57.33
1884  28-Feb-2022 - 00:00  01-Mar-2022 - 00:00          34.58
68.20

      NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  \
0           2.36         9.78        NaN          NaN         2.11
1           2.33         8.21        NaN          NaN         2.86
2          11.39        17.28        NaN          NaN         7.73
3           6.06        12.32        NaN          NaN         2.72
6           5.72         9.66        NaN          NaN         2.19
...          ...          ...        ...          ...          ...
1880        6.21        11.89      18.10        18.37        20.30
1881       11.21        14.10      25.32        27.52        20.64
1882         NaN          NaN        NaN          NaN        21.26
1883         NaN          NaN        NaN          NaN        21.77
```

```
1884        4.09        13.32      17.40        17.40        18.81

      CO (mg/m3)  Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0            NaN          24.63             0.52             2.95
1            NaN          20.49             0.13             2.01
2            NaN          13.04             0.45             3.52
3            NaN          19.42             0.65             3.98
6            NaN          35.32             0.36             3.20
...          ...            ...              ...              ...
1880        1.22         103.20             0.00             0.00
1881        1.13          46.97             0.00             0.00
1882        0.95          63.83             0.00             0.00
1883        1.05          84.68             0.00             0.00
1884        1.41          42.84             2.08             2.17

[1562 rows x 13 columns]
```

```
#capping the oultiers with the maximum and minimum
new_df_cap = df11.copy()
new_df_cap['PM2.5 (ug/m3)'] = np.where(new_df_cap['PM2.5 (ug/m3)'] >
upper_limit,upper_limit,
    np.where(new_df_cap['PM2.5 (ug/m3)'] <
lower_limit,lower_limit,new_df_cap['PM2.5 (ug/m3)']))

df11['PM2.5 (ug/m3)'].describe()
```

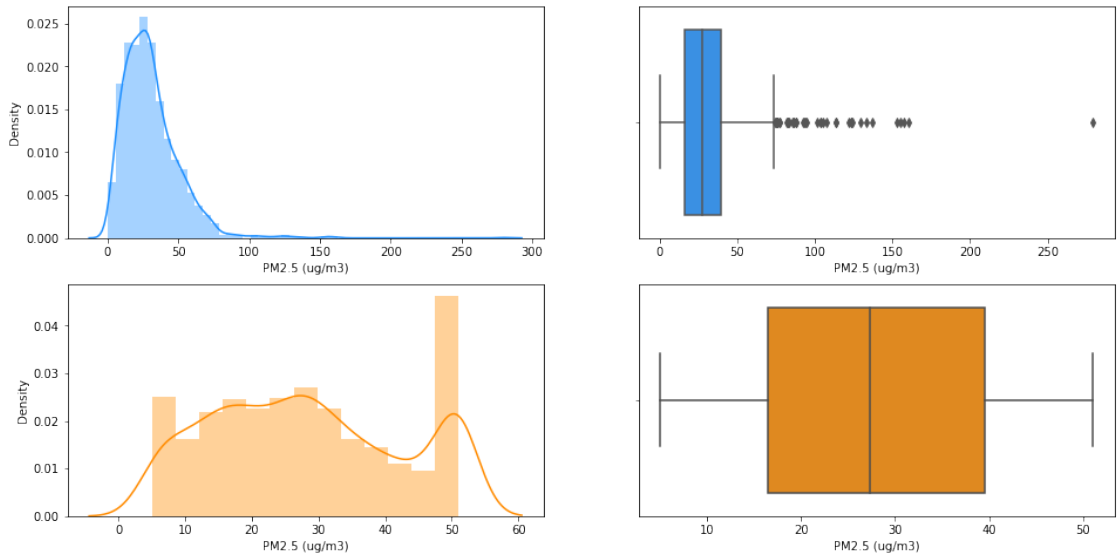
```
count    1841.000000
mean       30.500435
std        20.289850
min         0.410000
25%        16.540000
50%        27.280000
75%        39.530000
max       278.970000
Name: PM2.5 (ug/m3), dtype: float64
```

```
new_df_cap['PM2.5 (ug/m3)'].describe()
#The maximum value is deviated towards the maen unlike before capping.
```

```
count    1841.000000
mean       28.310193
std        14.220360
min         5.045000
25%        16.540000
50%        27.280000
75%        39.530000
max        51.025000
Name: PM2.5 (ug/m3), dtype: float64
```

```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
```

```
sns.distplot(df11['PM2.5 (ug/m3)'],color='dodgerblue' )
plt.subplot(2,2,2)
sns.boxplot(df11['PM2.5 (ug/m3)'],color='dodgerblue' )
plt.subplot(2,2,3)
sns.distplot(new_df_cap['PM2.5 (ug/m3)'],color='darkorange')
plt.subplot(2,2,4)
sns.boxplot(new_df_cap['PM2.5 (ug/m3)'],color='darkorange')
plt.show()

# data distribution have change from positively skewed to normal
distribution and ouliers are completely removed.
```



METHOD 3

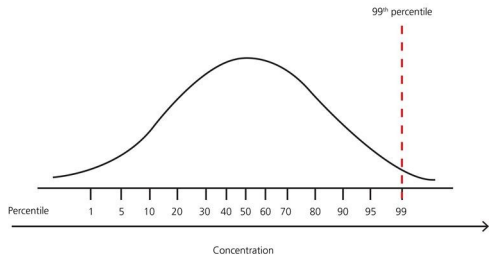
Percentile

– This technique works by setting a particular threshold value(user defined, or domain specific).

– While capping, we use a method is known as Winsorization.

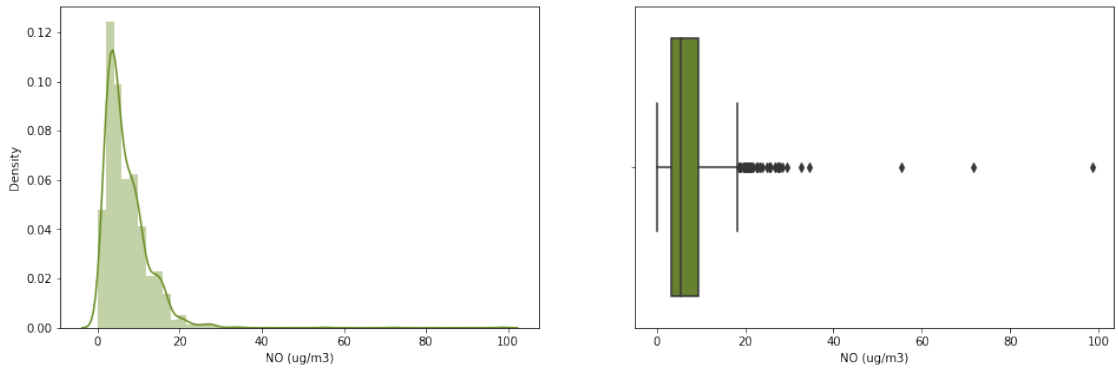
– Symmetry is maintained on both sides means if remove 1% from the right then in the left we also drop by 1%.

--For Other distributions: Use percentile-based approach.



```
df12 = pd.read_excel (r'/content/Data 2017-2022.xlsx',
sheet_name='Chennai')

plt.figure(figsize=(16,5))
plt.subplot(1,2,1)
sns.distplot(df12['NO (ug/m3)'],color='olivedrab')
plt.subplot(1,2,2)
sns.boxplot(df12['NO (ug/m3)'],color='olivedrab')

#NO values are skewed and outliers are present
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fad8051d7d0>
```




```
upper_limit = df12['NO (ug/m3)'].quantile(0.90)
lower_limit = df12['NO (ug/m3)'].quantile(0.10)
print(upper_limit,lower_limit)

# unlike inter quartile range, the percentage value which is
considered to be outlier is not fixed.
# But same percent of data points are detected as ouliers in both
extremes.
```

```
13.881000000000002 2.07

#rows with outliers
df12[(df12['NO (ug/m3)'] > upper_limit) | (df12['NO (ug/m3)'] <
lower_limit)]
```

```
                From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
11    12-Jan-2017 - 00:00  13-Jan-2017 - 00:00          42.32
NaN
12    13-Jan-2017 - 00:00  14-Jan-2017 - 00:00         103.51
NaN
13    14-Jan-2017 - 00:00  15-Jan-2017 - 00:00          33.96
NaN
14    15-Jan-2017 - 00:00  16-Jan-2017 - 00:00          54.93
NaN
54    24-Feb-2017 - 00:00  25-Feb-2017 - 00:00          21.08
NaN
```

```
...                   ...                   ...            ...
...
1820  26-Dec-2021 - 00:00  27-Dec-2021 - 00:00          18.97
33.59
1821  27-Dec-2021 - 00:00  28-Dec-2021 - 00:00          17.11
30.60
1822  28-Dec-2021 - 00:00  29-Dec-2021 - 00:00          18.40
32.68
1826  01-Jan-2022 - 00:00  02-Jan-2022 - 00:00          16.89
30.25
1853  28-Jan-2022 - 00:00  29-Jan-2022 - 00:00          14.52
26.41

      NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  \
11         22.63        26.31        NaN          NaN         2.57
12         32.73        38.26        NaN          NaN         2.65
13         24.83        27.27        NaN          NaN         2.48
14         16.29        20.86        NaN          NaN         3.54
54         26.66        37.35        NaN          NaN        22.53
...          ...          ...        ...          ...          ...
1820       16.36        20.26      33.89        21.71         9.41
1821       23.86        19.33      40.42        22.65         9.41
1822       16.55        19.39      35.94        21.86         9.39
1826       14.07        14.47      28.54        18.48         9.39
1853        2.03         5.11       7.13         8.41        13.75

      CO (mg/m3)  Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
11           NaN          30.78             0.74             3.52
12           NaN          30.94             6.60             4.35
13           NaN          44.53             0.23             0.77
14           NaN          45.14             1.00             2.82
54           NaN          34.30             1.64             2.09
...          ...            ...              ...              ...
1820        0.84          25.79             0.00             0.00
1821        0.76          24.77             0.00             0.00
1822        0.78          21.34             0.00             0.00
1826        0.78          36.92             0.00             0.00
1853        0.77          47.45             0.00             0.00

[369 rows x 13 columns]
```

```
#rows excluding outliers
df12[(df12['NO (ug/m3)'] <= upper_limit) & (df12['NO (ug/m3)'] >=
lower_limit)]
```

```
                From Date              To Date  PM2.5 (ug/m3)  PM10
(ug/m3)  \
0     01-Jan-2017 - 00:00  02-Jan-2017 - 00:00          32.61
NaN
1     02-Jan-2017 - 00:00  03-Jan-2017 - 00:00          22.93
```

```
NaN
2     03-Jan-2017 - 00:00  04-Jan-2017 - 00:00          24.19
NaN
3     04-Jan-2017 - 00:00  05-Jan-2017 - 00:00          33.61
NaN
4     05-Jan-2017 - 00:00  06-Jan-2017 - 00:00         129.38
NaN
...                   ...                  ...            ...
...
1878  22-Feb-2022 - 00:00  23-Feb-2022 - 00:00          31.25
63.08
1879  23-Feb-2022 - 00:00  24-Feb-2022 - 00:00          24.57
52.81
1880  24-Feb-2022 - 00:00  25-Feb-2022 - 00:00          32.05
64.31
1881  25-Feb-2022 - 00:00  26-Feb-2022 - 00:00          38.95
74.92
1884  28-Feb-2022 - 00:00  01-Mar-2022 - 00:00          34.58
68.20

      NO (ug/m3)  NO2 (ug/m3)  NOx (ppb)  NH3 (ug/m3)  SO2 (ug/m3)  \
0           2.36         9.78        NaN          NaN         2.11
1           2.33         8.21        NaN          NaN         2.86
2          11.39        17.28        NaN          NaN         7.73
3           6.06        12.32        NaN          NaN         2.72
4           5.58        12.67        NaN          NaN         2.65
...          ...          ...        ...          ...          ...
1878        5.02         9.97      14.98        14.92        20.20
1879        4.44         9.82      14.25        13.56        20.15
1880        6.21        11.89      18.10        18.37        20.30
1881       11.21        14.10      25.32        27.52        20.64
1884        4.09        13.32      17.40        17.40        18.81

      CO (mg/m3)  Ozone (ug/m3)  Benzene (ug/m3)  Toluene (ug/m3)
0            NaN          24.63             0.52             2.95
1            NaN          20.49             0.13             2.01
2            NaN          13.04             0.45             3.52
3            NaN          19.42             0.65             3.98
4            NaN          25.89             0.60             3.53
...          ...            ...              ...              ...
1878        1.18         167.65             0.00             0.00
1879        1.33         162.59             0.00             0.00
1880        1.22         103.20             0.00             0.00
1881        1.13          46.97             0.00             0.00
1884        1.41          42.84             2.08             2.17

[1475 rows x 13 columns]
```

```
#capping to handle outliers
new_df_cap = df12.copy()
```

```
new_df_cap['NO (ug/m3)'] = np.where(new_df_cap['NO (ug/m3)'] >
upper_limit,upper_limit,
    np.where(new_df_cap['NO (ug/m3)'] <
lower_limit,lower_limit,new_df_cap['NO (ug/m3)']))
```

```
df12['NO (ug/m3)'].describe()
```

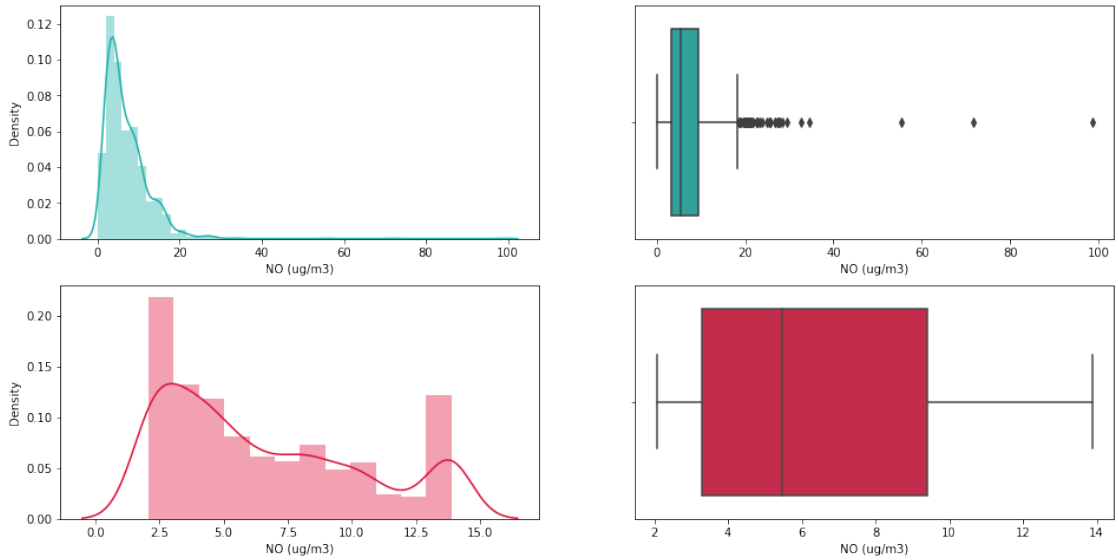
```
count    1844.000000
mean        6.952950
std         5.611016
min         0.010000
25%         3.295000
50%         5.460000
75%         9.400000
max        98.620000
Name: NO (ug/m3), dtype: float64
```

```
new_df_cap['NO (ug/m3)'].describe()
#removal of outliers will affect all statistical parameters
considerably.
```

```
count    1844.000000
mean        6.557823
std         3.856485
min         2.070000
25%         3.295000
50%         5.460000
75%         9.400000
max        13.881000
Name: NO (ug/m3), dtype: float64
```

```
plt.figure(figsize=(16,8))
plt.subplot(2,2,1)
sns.distplot(df12['NO (ug/m3)'],color='lightseagreen' )
plt.subplot(2,2,2)
sns.boxplot(df12['NO (ug/m3)'],color='lightseagreen')
plt.subplot(2,2,3)
sns.distplot(new_df_cap['NO (ug/m3)'],color='crimson')
plt.subplot(2,2,4)
sns.boxplot(new_df_cap['NO (ug/m3)'],color='crimson')
plt.show()
```

```
#graphically representation gives a clear view of how outlier analysis
can improve the distribution of data points.
```



There are many more advanced ways to handle outliers. But among these 3 methods, IQR seems to be the most effective one, though it changes from one analysis to analyis.

Outliers, unless they are data entry errors, are always an important part of a data set.

Figuring out why they are important is challenging and needs lot of analytical and lo. Then you have to figure out what to do about them.

REFERENCES :- https://www.analyticsvidhya.com/blog/2021/05/feature-engineering-how-to-detect-and-remove-outliers-with-python-code/

https://www.geeksforgeeks.org/detect-and-remove-the-outliers-using-python/

https://github.com/atulpatelDS/Youtube/blob/main/Data_Cleaning/Hands-on%20Handling%20missing%20value%20with%20List%20%26%20Pairwise%20Deletion%20with%20Python%20-%20Data%20Cleaning%20Tutorial%206.ipynb

https://www.analyticsvidhya.com/blog/2021/10/a-complete-guide-to-dealing-with-missing-values-in-python/

https://towardsdatascience.com/missing-data-and-imputation-89e9889268c8

https://www.geeksforgeeks.org/python-pandas-dataframe-ffill/https://www.geeksforgeeks.org/python-pandas-dataframe-ffill/


THANK YOU

LINEAR REGRESSION IMPUTATION

With regression imputation the information of other variables is used to predict the missing values in a variable by using a regression model. Commonly, first the regression model is estimated in the observed data and subsequently using the regression weights the missing values are predicted and replaced.

In [1]:
```
#importing libraries
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
```

In [2]:
```
df = pd.read_excel (r'Data 2017-2022.xlsx', sheet_name='Chennai')
```

In [3]:
```
df.sample(5)
```

Out[3]:

	PM2.5 (ug/m3)	PM10 (ug/m3)	NO (ug/m3)	NO2 (ug/m3)	NOx (ppb)	NH3 (ug/m3)	SO2 (ug/m3)	CO(mg/m3)	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
1153	27.41	NaN	9.10	3.20	11.72	NaN	5.86	0.29	37.80	0.01	0.00
1139	55.18	NaN	11.07	10.19	21.07	NaN	7.09	1.10	53.32	0.87	0.00
528	65.12	NaN	19.70	24.90	25.75	NaN	4.60	0.68	25.59	0.00	3.94
1127	29.34	NaN	9.31	4.73	13.39	NaN	7.31	0.97	23.58	0.11	0.00
1755	24.08	69.24	5.40	9.09	14.35	17.36	5.37	0.70	9.69	0.00	0.00

In [4]:
```
df.shape
```

Out[4]:
```
(1885, 11)
```

In [5]:
```
df.info()
```

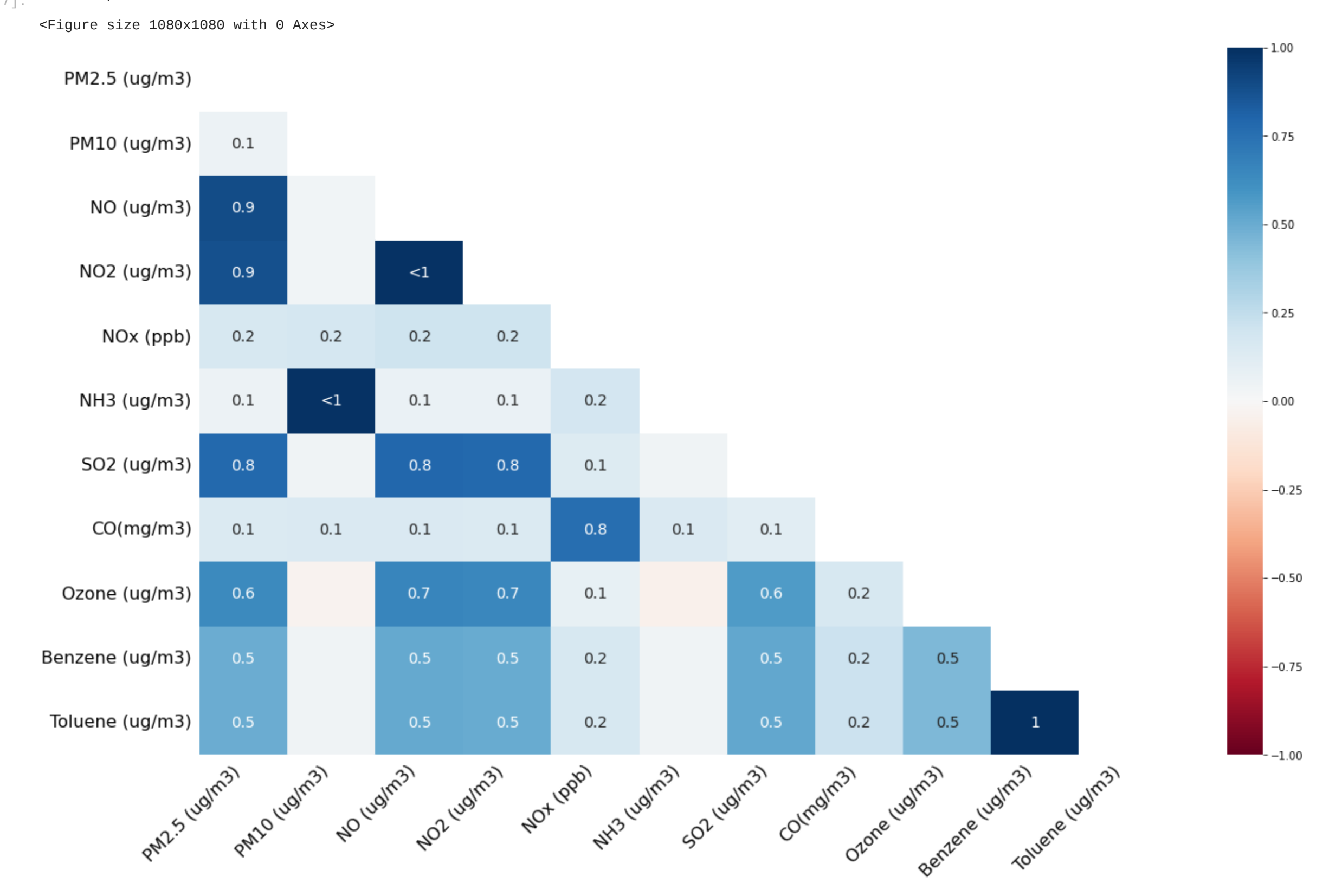
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1885 entries, 0 to 1884
Data columns (total 11 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   PM2.5 (ug/m3)    1841 non-null   float64
 1   PM10 (ug/m3)     306 non-null    float64
 2   NO (ug/m3)       1844 non-null   float64
 3   NO2 (ug/m3)      1843 non-null   float64
 4   NOx (ppb)        1598 non-null   float64
 5   NH3 (ug/m3)      306 non-null    float64
 6   SO2 (ug/m3)      1830 non-null   float64
 7   CO(mg/m3)        1687 non-null   float64
 8   Ozone (ug/m3)    1833 non-null   float64
 9   Benzene (ug/m3)  1859 non-null   float64
 10  Toluene (ug/m3)  1859 non-null   float64
dtypes: float64(11)
memory usage: 162.1 KB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
PM2.5 (ug/m3)        44
PM10 (ug/m3)       1579
NO (ug/m3)           41
NO2 (ug/m3)          42
NOx (ppb)           287
NH3 (ug/m3)        1579
SO2 (ug/m3)          55
CO(mg/m3)           198
Ozone (ug/m3)        52
Benzene (ug/m3)      26
Toluene (ug/m3)      26
dtype: int64
```

In [7]:
```
#helps in visualizing the correlation between all the columns
# Both NO and NO2 has strong positive correlation with PM2.5

import missingno as msno
plt.figure(figsize=(15,15))
msno.heatmap(df)
```

Out[7]:
```
<AxesSubplot:>
<Figure size 1080x1080 with 0 Axes>
```



1.Separating Null values

In [8]:
```
#test data contains only the rows in which CO column has NULL values.
test_data=df[df["CO(mg/m3)"].isna()]
```

In [9]:
```
test_data
```

Out[9]:

	PM2.5 (ug/m3)	PM10 (ug/m3)	NO (ug/m3)	NO2 (ug/m3)	NOx (ppb)	NH3 (ug/m3)	SO2 (ug/m3)	CO(mg/m3)	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
0	32.61	NaN	2.36	9.78	NaN	NaN	2.11	NaN	24.63	0.52	2.95
1	22.93	NaN	2.33	8.21	NaN	NaN	2.86	NaN	20.49	0.13	2.01
2	24.19	NaN	11.39	17.28	NaN	NaN	7.73	NaN	13.04	0.45	3.52
3	33.61	NaN	6.06	12.32	NaN	NaN	2.72	NaN	19.42	0.65	3.98
4	129.38	NaN	5.58	12.67	NaN	NaN	2.65	NaN	25.89	0.60	3.53
...	...	...	...	...	...	...	...	...	...	...	...
1071	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1072	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1082	35.22	NaN	6.74	3.61	9.51	NaN	4.27	NaN	31.88	4.14	0.00
1083	48.23	NaN	7.42	3.24	9.79	NaN	4.48	NaN	38.56	11.22	0.26
1088	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

198 rows × 11 columns

2.Droping the null values from df

In [10]:
```
#dropping all the rows with null values(in any column) from the original datset
df.dropna(inplace=True)
```

In [11]:
```
df
```

Out[11]:

	PM2.5 (ug/m3)	PM10 (ug/m3)	NO (ug/m3)	NO2 (ug/m3)	NOx (ppb)	NH3 (ug/m3)	SO2 (ug/m3)	CO(mg/m3)	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
1576	7.34	60.71	1.77	3.74	5.51	5.36	8.00	0.50	12.44	0.00	0.00
1577	5.08	56.41	1.81	3.74	5.54	5.36	6.61	0.51	27.50	0.00	0.00
1578	5.84	56.20	1.70	3.73	5.43	5.36	5.89	0.57	19.75	0.00	0.00
1580	8.28	56.20	1.72	3.74	5.46	5.36	5.72	0.53	25.36	0.00	0.00
1584	5.40	56.20	1.76	3.75	5.51	5.36	5.37	0.54	14.95	0.00	0.00
...	...	...	...	...	...	...	...	...	...	...	...
1878	31.25	63.08	5.02	9.97	14.98	14.92	20.20	1.18	167.65	0.00	0.00
1879	24.57	52.81	4.44	9.82	14.25	13.56	20.15	1.33	162.59	0.00	0.00
1880	32.05	64.31	6.21	11.89	18.10	18.37	20.30	1.22	103.20	0.00	0.00
1881	38.95	74.92	11.21	14.10	25.32	27.52	20.64	1.13	46.97	0.00	0.00
1884	34.58	68.20	4.09	13.32	17.40	17.40	18.81	1.41	42.84	2.08	2.17

286 rows × 11 columns

In [12]:
```
#all null values are dropped
df.isnull().sum()
```

Out[12]:
```
PM2.5 (ug/m3)      0
PM10 (ug/m3)       0
NO (ug/m3)         0
NO2 (ug/m3)        0
NOx (ppb)          0
NH3 (ug/m3)        0
SO2 (ug/m3)        0
CO(mg/m3)          0
Ozone (ug/m3)      0
Benzene (ug/m3)    0
Toluene (ug/m3)    0
dtype: int64
```

In [13]:
```
#shape has drastically reduced from (1885,11) to (286,11)
df.shape
```

Out[13]:
```
(286, 11)
```

3.Create "x_train" and "y_train" from df

In [14]:
```
#y_train is the column which we are choosing for imputation without null values
# y_train contains completely filled data(column)
#Train data will have only rows with non null values
y_train=df["CO(mg/m3)"]
```

In [15]:
```
y_train
```

Out[15]:
```
1576    0.50
1577    0.51
1578    0.57
1580    0.53
1584    0.54
        ...
1878    1.18
1879    1.33
1880    1.22
1881    1.13
1884    1.41
Name: CO(mg/m3), Length: 286, dtype: float64
```

In [16]:
```
y_train.shape
```

Out[16]:
```
(286,)
```

In [17]:
```
#x_train --> Filled dataframe(no null values) except df['CO (mg/m3)'] (y_train column) features.
x_train=df.drop("CO(mg/m3)",axis=1)
```

In [18]:
```
x_train
```

Out[18]:

	PM2.5 (ug/m3)	PM10 (ug/m3)	NO (ug/m3)	NO2 (ug/m3)	NOx (ppb)	NH3 (ug/m3)	SO2 (ug/m3)	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
1576	7.34	60.71	1.77	3.74	5.51	5.36	8.00	12.44	0.00	0.00
1577	5.08	56.41	1.81	3.74	5.54	5.36	6.61	27.50	0.00	0.00
1578	5.84	56.20	1.70	3.73	5.43	5.36	5.89	19.75	0.00	0.00
1580	8.28	56.20	1.72	3.74	5.46	5.36	5.72	25.36	0.00	0.00
1584	5.40	56.20	1.76	3.75	5.51	5.36	5.37	14.95	0.00	0.00
...	...	...	...	...	...	...	...	...	...	...
1878	31.25	63.08	5.02	9.97	14.98	14.92	20.20	167.65	0.00	0.00
1879	24.57	52.81	4.44	9.82	14.25	13.56	20.15	162.59	0.00	0.00
1880	32.05	64.31	6.21	11.89	18.10	18.37	20.30	103.20	0.00	0.00
1881	38.95	74.92	11.21	14.10	25.32	27.52	20.64	46.97	0.00	0.00
1884	34.58	68.20	4.09	13.32	17.40	17.40	18.81	42.84	2.08	2.17

286 rows × 10 columns

In [19]:
```
x_train.shape
```

Out[19]:
```
(286, 10)
```

4.Build the model

In [20]:
```
#simple regression model using inbuilt libraries in sklearn
from sklearn.linear_model import LinearRegression
lr=LinearRegression()
```

In [21]:
```
#train the model on train data set x_train, y_train
lr.fit(x_train,y_train)
```

Out[21]:
```
LinearRegression()
```

5.Create the x_test from test data

In [22]:
```
#removing CO Column from test_data
#the whole column will have only null values
x_test=test_data.drop("CO(mg/m3)",axis=1)
```

In [23]:
```
x_test
```

Out[23]:

	PM2.5 (ug/m3)	PM10 (ug/m3)	NO (ug/m3)	NO2 (ug/m3)	NOx (ppb)	NH3 (ug/m3)	SO2 (ug/m3)	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
0	32.61	NaN	2.36	9.78	NaN	NaN	2.11	24.63	0.52	2.95
1	22.93	NaN	2.33	8.21	NaN	NaN	2.86	20.49	0.13	2.01
2	24.19	NaN	11.39	17.28	NaN	NaN	7.73	13.04	0.45	3.52
3	33.61	NaN	6.06	12.32	NaN	NaN	2.72	19.42	0.65	3.98
4	129.38	NaN	5.58	12.67	NaN	NaN	2.65	25.89	0.60	3.53
...	...	...	...	...	...	...	...	...	...	...
1071	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1072	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1082	35.22	NaN	6.74	3.61	9.51	NaN	4.27	31.88	4.14	0.00
1083	48.23	NaN	7.42	3.24	9.79	NaN	4.48	38.56	11.22	0.26
1088	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

198 rows × 10 columns

In [24]:
```
x_test.shape
```

Out[24]:
```
(198, 10)
```

In [25]:
```
#removing columns with huge missing data (other than CO)
x_test=df.drop(labels='PM10 (ug/m3)', axis=1)
x_test=df.drop(labels='NH3 (ug/m3)', axis=1)
```

In [26]:
```
#mean imputation for other columns(to get completely filled data)
x=['PM2.5 (ug/m3)','NO (ug/m3)','NO2 (ug/m3)','NOx (ppb)','SO2 (ug/m3)','Ozone (ug/m3)','Benzene (ug/m3)','Toluene (ug/m3)']
x_test[x] = x_test[x].fillna(x_test[x].mean())
```

In [27]:
```
#final X_test data
x_test
```

Out[27]:

	PM2.5 (ug/m3)	PM10 (ug/m3)	NO (ug/m3)	NO2 (ug/m3)	NOx (ppb)	SO2 (ug/m3)	CO(mg/m3)	Ozone (ug/m3)	Benzene (ug/m3)	Toluene (ug/m3)
1576	7.34	60.71	1.77	3.74	5.51	8.00	0.50	12.44	0.00	0.00
1577	5.08	56.41	1.81	3.74	5.54	6.61	0.51	27.50	0.00	0.00
1578	5.84	56.20	1.70	3.73	5.43	5.89	0.57	19.75	0.00	0.00
1580	8.28	56.20	1.72	3.74	5.46	5.72	0.53	25.36	0.00	0.00
1584	5.40	56.20	1.76	3.75	5.51	5.37	0.54	14.95	0.00	0.00
...	...	...	...	...	...	...	...	...	...	...
1878	31.25	63.08	5.02	9.97	14.98	20.20	1.18	167.65	0.00	0.00
1879	24.57	52.81	4.44	9.82	14.25	20.15	1.33	162.59	0.00	0.00
1880	32.05	64.31	6.21	11.89	18.10	20.30	1.22	103.20	0.00	0.00
1881	38.95	74.92	11.21	14.10	25.32	20.64	1.13	46.97	0.00	0.00
1884	34.58	68.20	4.09	13.32	17.40	18.81	1.41	42.84	2.08	2.17

286 rows × 10 columns

6.Apply the model on X_test and predicting the missing values

In [28]:
```
#predict missing values in CO
y_pred=lr.predict(x_test)
```

```
C:\Users\Murali\anaconda3\lib\site-packages\sklearn\base.py:493: FutureWarning: The feature names should match those that were passed during fit. Starting version 1.2, an error wil
l be raised.
Feature names unseen at fit time:
- CO(mg/m3)
Feature names seen at fit time, yet now missing:
- NH3 (ug/m3)

  warnings.warn(message, FutureWarning)
```

In [29]:
```
y_pred
```

Out[29]:
```
array([0.59228563, 0.61962348, 0.62705292, 0.63774318, 0.62932662,
       0.64499524, 0.63190067, 0.63662446, 0.64085806, 0.63112603,
       0.66193982, 0.63432952, 0.6250001 , 0.61588959, 0.63541253,
       0.65988706, 0.64634404, 0.6324636 , 0.65890025, 0.61271557,
       0.63625866, 0.64555923, 0.65381844, 0.6059577 , 0.60001191,
       0.66206673, 0.69105188, 0.65075541, 0.64649899, 0.64660843,
       0.67292409, 0.68017852, 0.685563  , 0.69145025, 0.6956958 ,
       0.66071735, 0.67175487, 0.66056909, 0.69386773, 0.67807293,
       0.70354965, 0.72718457, 0.71713292, 0.71177313, 0.72836997,
       0.71564759, 0.69366327, 0.68948267, 0.71429937, 0.68812516,
       0.66224837, 0.67582786, 0.67419213, 0.64782902, 0.63380382,
       0.64063845, 0.66990378, 0.62601466, 0.65992033, 0.65061993,
       0.66360482, 0.60947098, 0.58858985, 0.58633064, 0.56861086,
       0.56702375, 0.56123403, 0.58457803, 0.56986624, 0.57259753,
       0.57166757, 0.56497215, 0.58015565, 0.56769309, 0.56911783,
       0.56218718, 0.57161908, 0.54129584, 0.56989725, 0.56344829,
       0.57258998, 0.62921369, 0.73022643, 0.78714637, 0.79030559,
       0.69412242, 0.68798278, 0.65876106, 0.67313302, 0.67752268,
       0.66468178, 0.70016527, 0.69283155, 0.6899738 , 0.67222971,
       0.65947704, 0.66541113, 0.66665383, 0.67459358, 0.67898217,
       0.70306086, 0.67150521, 0.6849362 , 0.66357579, 0.69802801,
       0.68192827, 0.66741474, 0.68215241, 0.66303789, 0.66010201,
       0.66780663, 0.69072277, 0.67443669, 0.67774774, 0.68541922,
       0.66768133, 0.6531067 , 0.71422736, 0.77152578, 0.76512787,
       0.75690584, 0.73868954, 0.74475102, 0.76296776, 0.72368957,
       0.785243  , 0.74760146, 0.71775699, 0.72639743, 0.70486148,
       0.73765223, 0.72679441, 0.83350304, 0.74956892, 0.73310297,
       0.71908346, 0.7338085 , 0.72586447, 0.74761556, 0.77919157,
       0.79431546, 0.74531671, 0.74285525, 0.74212471, 0.71678162,
       0.72952827, 0.78563538, 0.75933413, 0.72230601, 0.78343442,
       0.75532554, 0.73343087, 0.75334508, 0.7600209 , 0.74168451,
       0.75273629, 0.69539106, 0.71781781, 0.70365729, 0.71964509,
       0.81901071, 0.80165038, 0.73646458, 0.71819528, 0.75067636,
       0.79682627, 0.75802258, 0.78172673, 0.75740647, 0.791109  ,
       0.75488768, 0.7217279 , 0.70881339, 0.80207054, 0.88315102,
       0.82847624, 0.76712955, 0.77072128, 0.57318296, 0.49568106,
       0.7451032 , 0.69928489, 0.72391702, 1.08306301, 0.8928276 ,
       0.73477416, 0.88650758, 0.87986634, 0.88063784, 0.84382784,
       0.85657152, 0.90938751, 0.84874723, 0.90987007, 0.91939003,
       0.87014873, 0.85307269, 0.97259413, 0.78528041, 0.86595344,
       0.83286882, 1.01267904, 0.83654114, 0.93808168, 0.96194497,
       0.94276975, 0.86350447, 0.89296874, 0.82524174, 0.81848902,
       0.81705371, 0.81582667, 0.82115034, 0.81987842, 0.82353856,
       0.81254696, 0.80351282, 0.78558278, 0.93263829, 0.8712541 ,
       0.83911844, 0.83569963, 0.83051115, 0.83022039, 0.89882398,
       0.95165272, 0.99423786, 0.86736811, 0.95470693, 0.93603636,
       0.8569465 , 0.84801477, 0.82857527, 0.85773052, 0.73554856,
       0.63938144, 0.63970605, 0.634049  , 0.62816029, 0.63061425,
       0.63232597, 0.60121669, 0.57083018, 0.56681616, 0.57786824,
       0.57674705, 0.56173981, 0.59148088, 0.59360357, 0.58997044,
       0.59943047, 0.59233343, 0.5837935 , 0.57823732, 0.58180757,
       0.57682188, 0.60087936, 0.6090365 , 0.60188971, 0.5964905 ,
       0.59255419, 0.58005114, 0.57647086, 0.60306195, 0.63878237,
       0.70870395, 0.81379321, 0.82386605, 0.85976926, 1.01729138,
       0.88561119, 1.00027378, 0.8563058 , 0.86013643, 0.89775106,
       0.97919073, 1.00239402, 0.80132994, 0.82251149, 0.9478614 ,
       0.85178601, 0.76586966, 0.75547156, 0.76623338, 0.74190453,
       1.11402078])
```

In [30]:
```
y_pred.shape
```

Out[30]:
```
(286,)
```

7.Replace the missing values with predicted values

In [31]:
```
#replace all null values with y_pred array
test_data.loc[test_data.CO(mg/m3).isnull(),'CO(mg/m3)']=y_pred
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_33580/1261623517.py in <module>
      1 #replace all null values with y_pred array
----> 2 test_data.loc[test_data.CO(mg/m3).isnull(),'CO(mg/m3)']=y_pred

~\anaconda3\lib\site-packages\pandas\core\generic.py in __getattr__(self, name)
   5485         ):
   5486             return self[name]
-> 5487         return object.__getattribute__(self, name)
   5488
   5489     def __setattr__(self, name: str, value) -> None:

AttributeError: 'DataFrame' object has no attribute 'CO'
```

In [ ]:
```
#this dataframe is the complete data after imputation of missing values using linear regression
test_data
```

Thank you!!!!