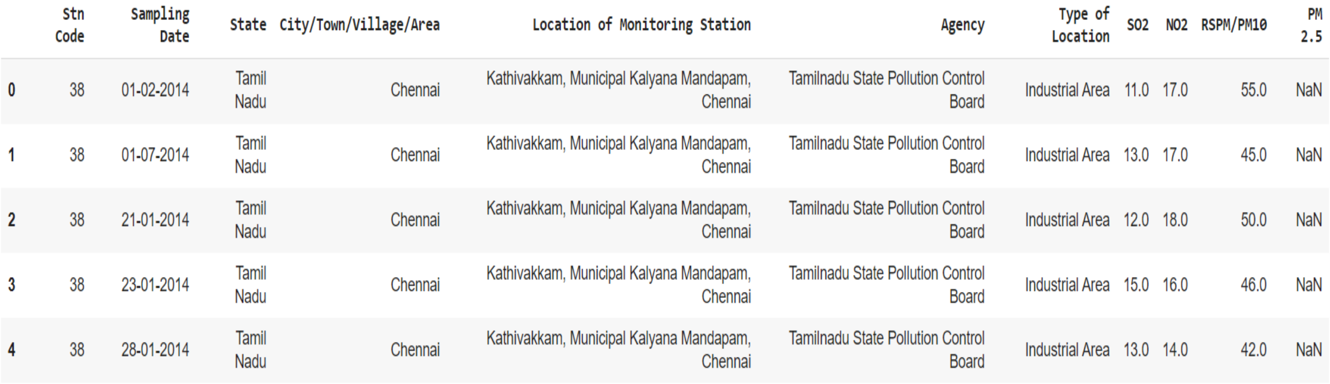
DAC Phase 3:
Problem Statement: Air Quality Assessment of TamilNadu

Loading and Pre-processing of data:

```
from google.colab import drive
drive.mount('/content/drive')
```

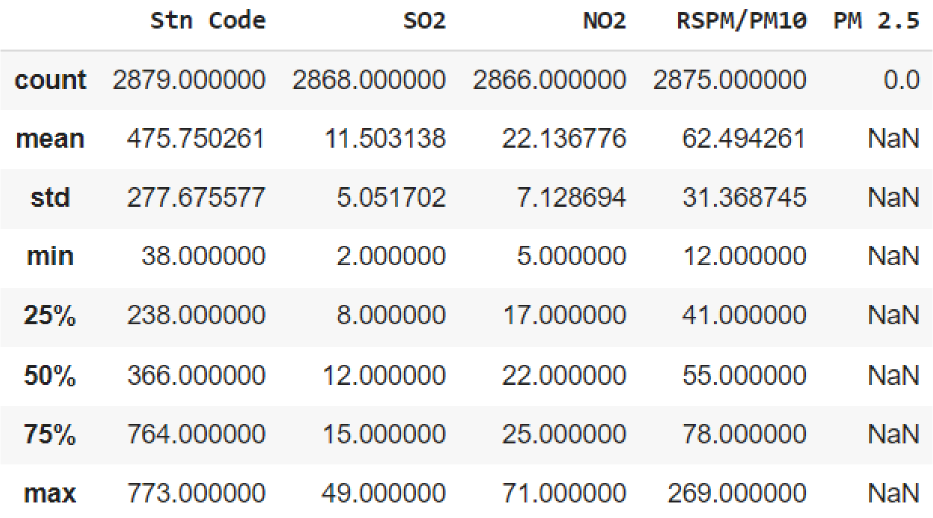
Loading data

```
import pandas as pd
import numpy as np
data = pd.read_csv('/content/drive/MyDrive/datasets/datasets/Air_quality.csv')
data.head(5)
```



	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
0	38	01-02-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	11.0	17.0	55.0	NaN
1	38	01-07-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	17.0	45.0	NaN
2	38	21-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	12.0	18.0	50.0	NaN
3	38	23-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	15.0	16.0	46.0	NaN
4	38	28-01-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial Area	13.0	14.0	42.0	NaN

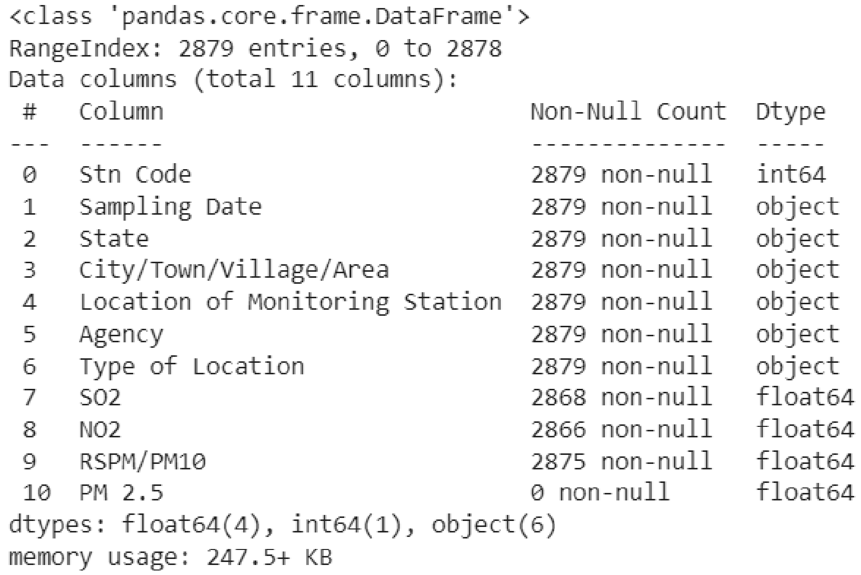
```
data.describe()
```



	Stn Code	SO2	NO2	RSPM/PM10	PM 2.5
count	2879.000000	2868.000000	2866.000000	2875.000000	0.0
mean	475.750261	11.503138	22.136776	62.494261	NaN
std	277.675577	5.051702	7.128694	31.368745	NaN
min	38.000000	2.000000	5.000000	12.000000	NaN
25%	238.000000	8.000000	17.000000	41.000000	NaN
50%	366.000000	12.000000	22.000000	55.000000	NaN
75%	764.000000	15.000000	25.000000	78.000000	NaN
max	773.000000	49.000000	71.000000	269.000000	NaN

This command is used to view the brief summary of the dataset. We can see the mathematical parameters such as percentiles, standard deviation , mean, minimum and maximum values and count of each column.

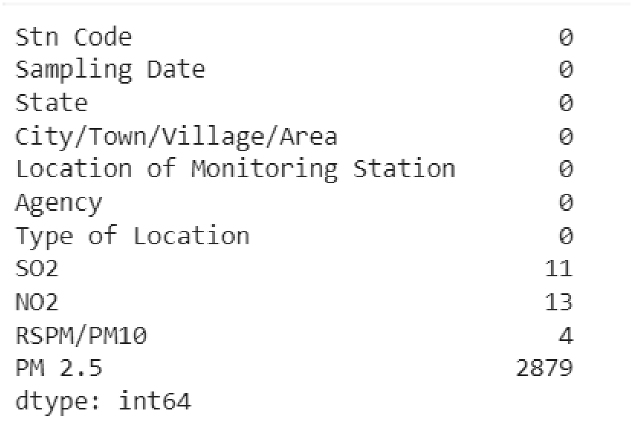
data.info()



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2879 entries, 0 to 2878
Data columns (total 11 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   Stn Code                       2879 non-null   int64  
 1   Sampling Date                  2879 non-null   object 
 2   State                          2879 non-null   object 
 3   City/Town/Village/Area         2879 non-null   object 
 4   Location of Monitoring Station  2879 non-null   object 
 5   Agency                         2879 non-null   object 
 6   Type of Location               2879 non-null   object 
 7   SO2                            2868 non-null   float64
 8   NO2                            2866 non-null   float64
 9   RSPM/PM10                      2875 non-null   float64
 10  PM 2.5                         0 non-null      float64
dtypes: float64(4), int64(1), object(6)
memory usage: 247.5+ KB
```

Info command is used check the datatype of every column and the count of each column. The difference between the describe() and info() is that describe command will give the mathematical parameters but info command will not give the mathematical parameters such as mean and standard deviation

data.isna().sum()



```
Stn Code                          0
Sampling Date                     0
State                             0
City/Town/Village/Area            0
Location of Monitoring Station    0
Agency                            0
Type of Location                  0
SO2                              11
NO2                              13
RSPM/PM10                         4
PM 2.5                         2879
dtype: int64
```

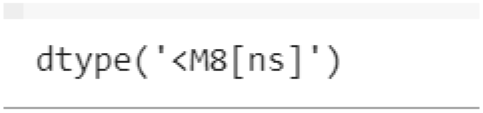
The above command is used to check for null values in each column. We can see that there are null values in the columns such as SO2,NO2,RSPM. It is very necessary to take action to clear the null values in the data set

```
mean_so2 = data['SO2'].mean()
data['SO2'] = data['SO2'].fillna(mean_so2)

mean_no2 = data['NO2'].mean()
data['NO2'] = data['NO2'].fillna(mean_no2)

mean_rspm = data['RSPM/PM10'].mean()
data['RSPM/PM10'] = data['RSPM/PM10'].fillna(mean_rspm)
data.drop('PM 2.5',axis=1,inplace=True)
```

Here fillna() method is used to fill the null values by the mean of the particular column

Converting the date column to date format from object

```
data['Sampling Date'] = pd.to_datetime(data['Sampling Date'])
data['Sampling Date'].dtype
```



```
dtype('<M8[ns]')
```

Initially the data type of the ‘Sampling Date’ column was object this will not be suitable to train a model or analyse the data set , so the data type of the column is converted to pandas date and time using pandas.to_datetime()

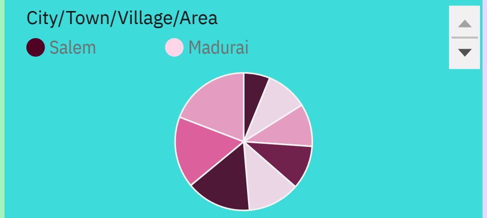
Tab 1

SO2

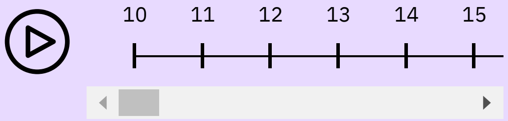
34

SO2

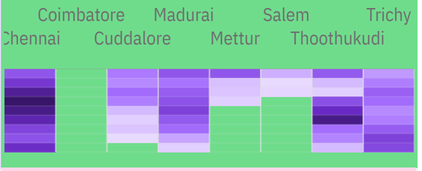
RSPM/PM10 by City/Town/Village/Area



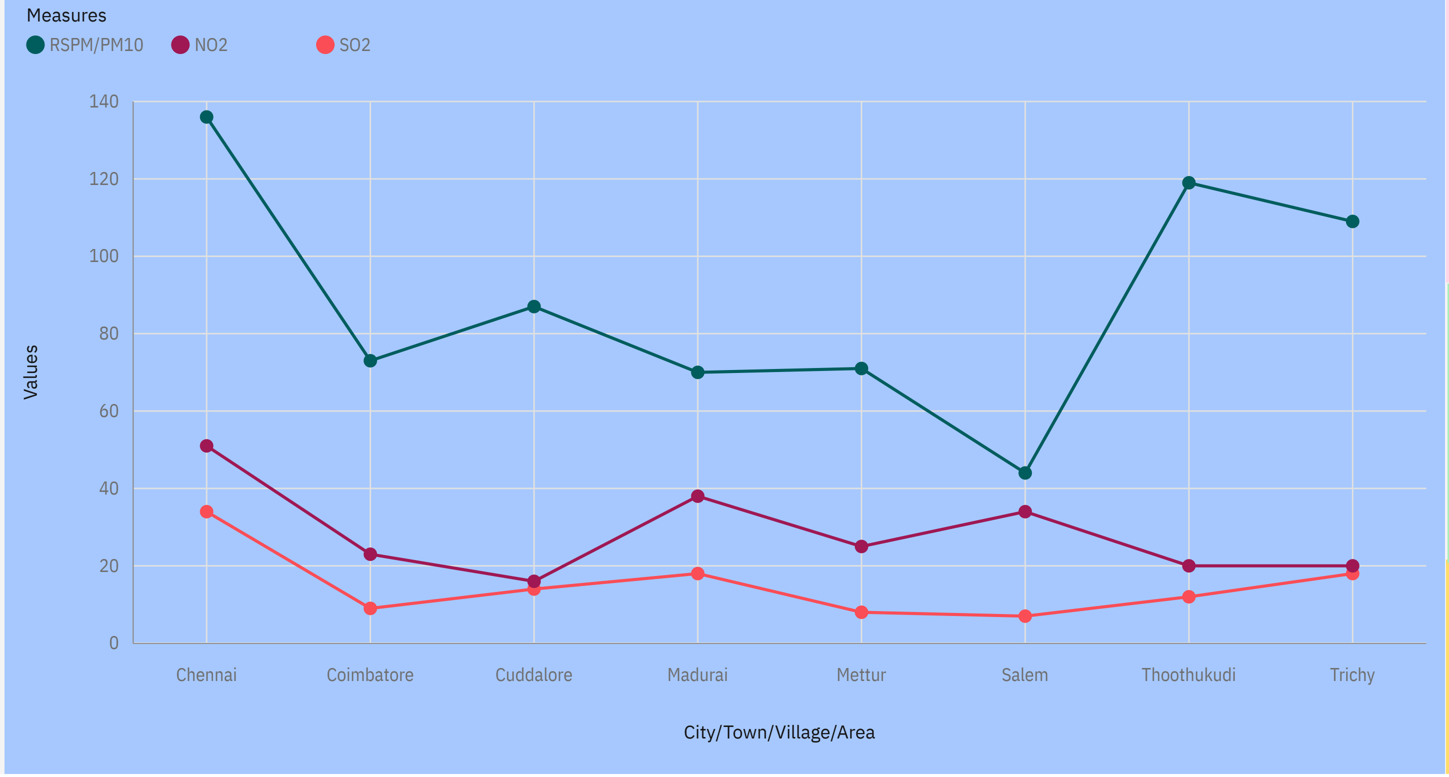
NO2



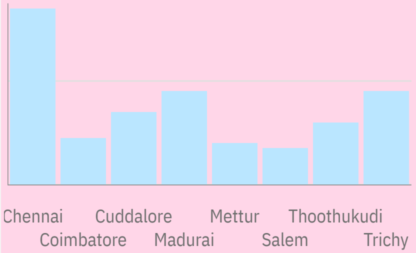
RSPM/PM10 by SO2 and City/Town/Village/Area



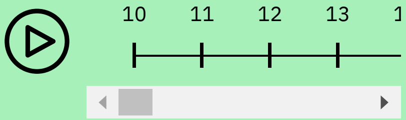
RSPM/PM10, NO2 and SO2 by City/Town/Village/Area



SO2 by City/Town/Village/Area



SO2



NO2

54

NO2

Insights:

1. Chennai has the highest RSPM/PM10 at 654, out of which SO2 13 contributed the most at 59.
2. 4 has a RSPM/PM10 of 61 for Coimbatore.
3. From 2014-01-30 to 2014-01-31, 10's RSPM/PM10 increased by 300%.
4. Chennai has the highest SO2 due to Stn Code 161.
5. Chennai is the most frequently occurring category of City/Town/Village/Area with a count of 1000 items with RSPM/PM10 values (34.7 % of the total).
6. The total number of results for RSPM/PM10, across all City/Town/Village/Area, is nearly three thousand.