

AIR Q ASSESSMENT TN

Phase 3: Development part-1

Dataset Link: <https://tn.data.gov.in/resource/location-wise-daily-ambient-air-quality-tamil-nadu-year-2014>

Loading Dataset:

Load your dataset into a pandas DataFrame, as we discussed in the previous response.

```
import pandas as pd

Import matplotlib.pyplot as plt

Import seaborn as sns

df=pd.read_csv("C:\\\\Air Q TN.csv")
```

Preprocessing:

```
print(df.head())
```

This will display the first few rows of the dataset, including column names and some sample data.

Output:

	Stn	Code	Sampling Date	State	City/Town/Village/Area	\
0	38	01-02-14	Tamil Nadu		Chennai	
1	38	01-07-14	Tamil Nadu		Chennai	
2	38	21-01-14	Tamil Nadu		Chennai	
3	38	23-01-14	Tamil Nadu		Chennai	
4	38	28-01-14	Tamil Nadu		Chennai	

	Location of Monitoring Station	\
0	Kathivakkam, Municipal Kalyana Mandapam, Chennai	
1	Kathivakkam, Municipal Kalyana Mandapam, Chennai	
2	Kathivakkam, Municipal Kalyana Mandapam, Chennai	
3	Kathivakkam, Municipal Kalyana Mandapam, Chennai	

4 Kathivakkam, Municipal Kalyana Mandapam, Chennai

```
Agency Type of Location SO2 NO2 \
0 Tamilnadu State Pollution Control Board Industrial Area 11.0 17.0
1 Tamilnadu State Pollution Control Board Industrial Area 13.0 17.0
2 Tamilnadu State Pollution Control Board Industrial Area 12.0 18.0
3 Tamilnadu State Pollution Control Board Industrial Area 15.0 16.0
4 Tamilnadu State Pollution Control Board Industrial Area 13.0 14.0
```

```
RSPM/PM10 PM 2.5
0 55.0 NaN
1 45.0 NaN
2 50.0 NaN
3 46.0 NaN
4 42.0 NaN
```

```
print(df.info())
```

This will show you the data types of each column and how many non-null entries there are.

Output:

```
<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
None
```

```
print(df.describe())
```

This will provide statistics like mean, standard deviation, minimum, maximum, etc., for numerical columns.

Output:

	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

```
print(df.isnull().sum())
```

Output:

	Stn Code	
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	

```
print(df['SO2'].mean())
```

Output:

```
11.503138075313808
```

```
print(df['SO2'].median())
```

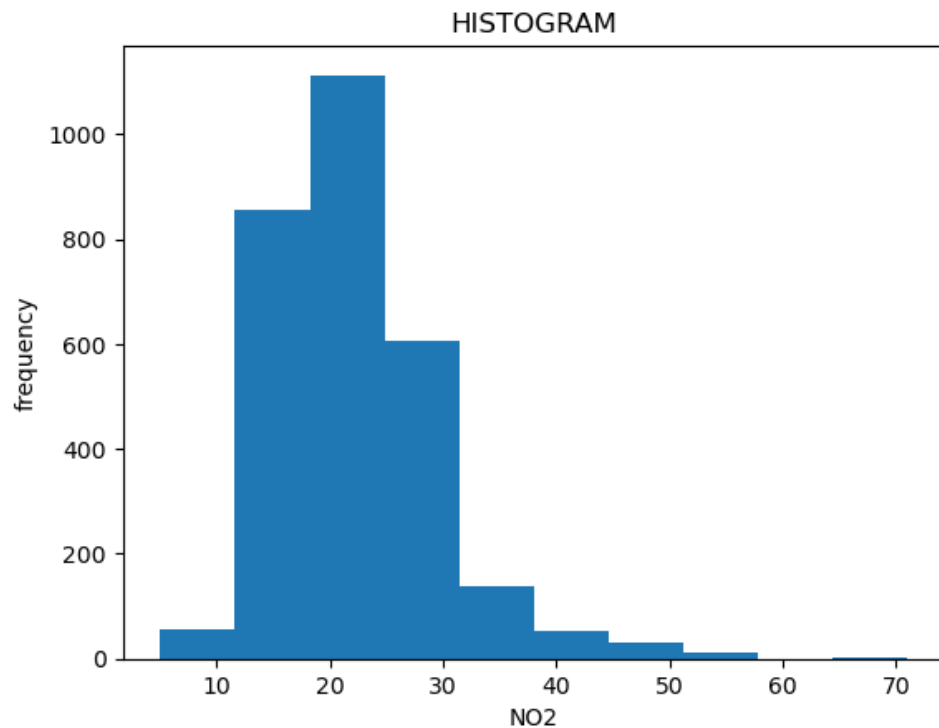
Output:

```
12.0
```

Histogram:

```
plt.title("HISTOGRAM")
plt.hist(df["NO2"])
plt.xlabel("NO2")
plt.ylabel("frequency")
plt.show()
```

Output:



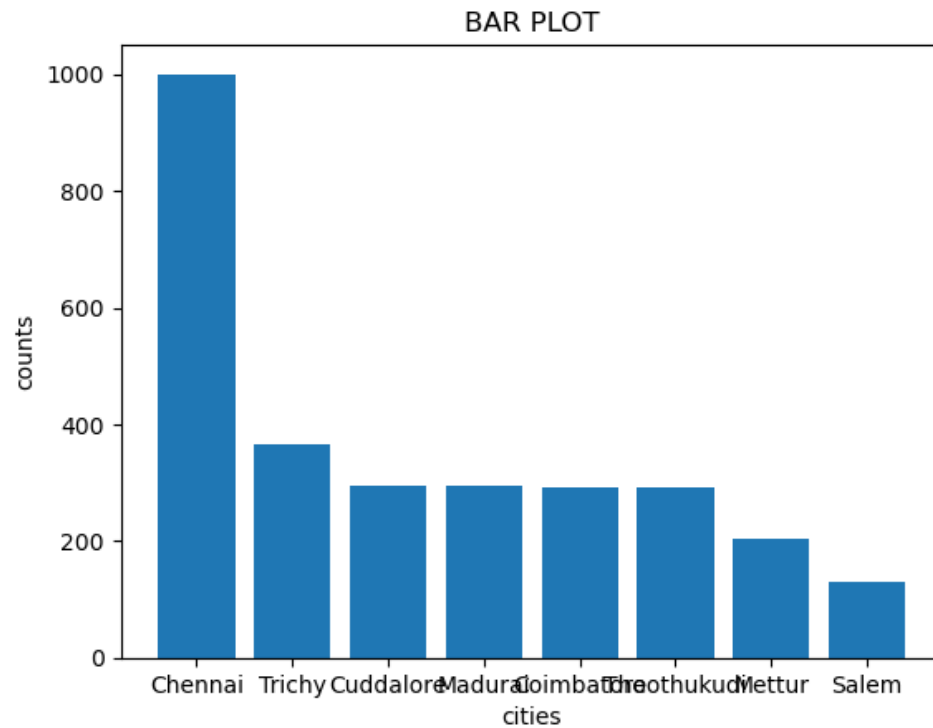
Bar Chart:

```
plt.title("BAR PLOT")

x=df["City/Town/Village/Area"].value_counts().nlargest(10)

plt.bar(x.keys(),x.values)
plt.xlabel("cities")
plt.ylabel("counts")
Plt.show()
```

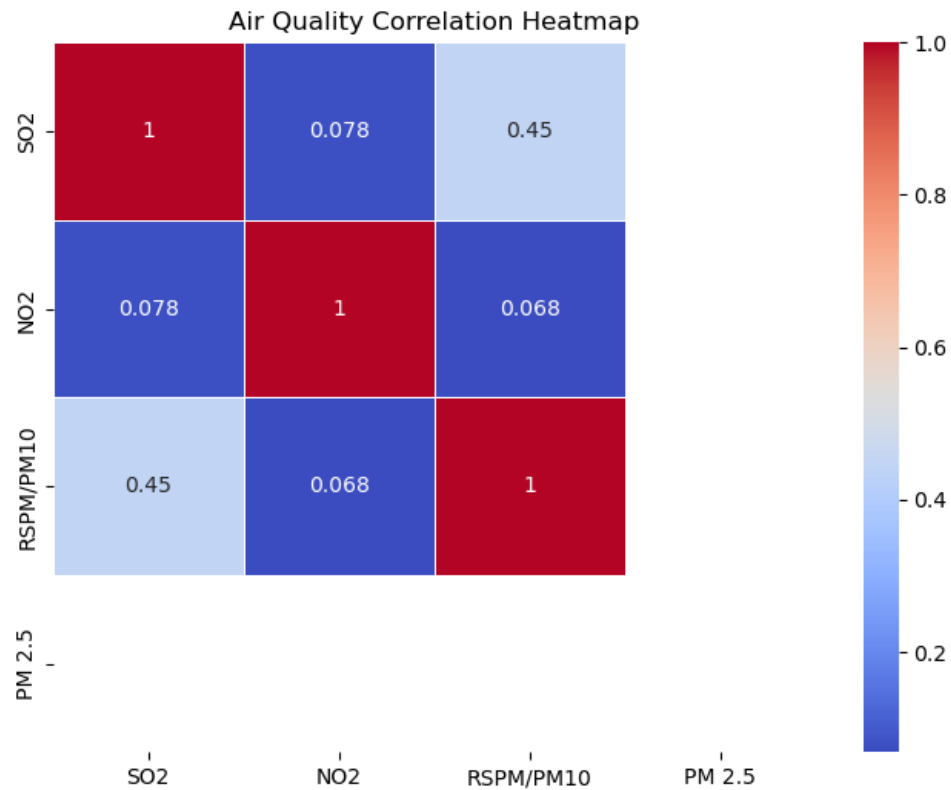
Output:



Heatmap:

```
numeric_columns = ['SO2', 'NO2', 'RSPM/PM10', 'PM 2.5']  
correlation_matrix = df[numeric_columns].corr()  
plt.figure(figsize=(8, 6))  
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',  
linewidths=0.5)  
plt.title('Air Quality Correlation Heatmap')  
plt.show()
```

Output:



Documentation:

It's essential to document EDA process, including the visualizations, insights, and any preprocessing steps. This documentation will help us and others understand the analysis later.