DATE : 01.11.2023

TEAM ID : 719

PROJECT TITLE : Air Quality Analysis in Tamilnadu

Import Libraries

In [121]:

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

Loading Dataset

In [122]:

```
data = pd.read_csv('cpcb_dly_aq_tamil_nadu-2014.csv')
```

Data Exploration

In [123]:

```
data
```

Out [123]:

	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
...	...	...	...	...	...	...	...	...	...	...	...
2874	773	12-03-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.0	102.0	NaN
2875	773	12-10-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.0	91.0	NaN
2876	773	17-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.0	100.0	NaN

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	PM 2.5
2877	773	24-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.0	95.0	NaN
2878	773	31-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.0	94.0	NaN

2879 rows × 11 columns

In [124]:

```
# Drop PM2.5
new_data = data.drop(['PM 2.5'], axis=1)
```

In [125]:

```
#mean SO2,NO2,RSPM/PM10
mean_SO2 = new_data['SO2'].mean()
mean_NO2 = new_data['NO2'].mean()
mean_RSPM_PM10 = new_data['RSPM/PM10'].mean()
```

In [126]:

```
# Assuming your DataFrame is named 'air_quality_data'
average=new_data.groupby(['Location of Monitoring Station','City/Town/Village/Area', 'Type of Location'])[['
```

In [127]:

```
average.mean()
```

Out [127]:

```
SO2          11.548218
NO2          21.889282
RSPM/PM10    62.605155
dtype: float64
```

In [128]:

```
new_data['SO2'].fillna(value=mean_SO2,inplace=True)
new_data['NO2'].fillna(value=mean_NO2,inplace=True)
new_data['RSPM/PM10'].fillna(value=mean_RSPM_PM10,inplace=True)
```

In [129]:

```
new_data
```

Out [129]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10
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
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
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
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
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
...	...	...	...	...	...	...	...	...	...	...
2874	773	12-03-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	18.0	102.0
2875	773	12-10-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	12.0	14.0	91.0
2876	773	17-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	19.0	22.0	100.0
2877	773	24-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	15.0	17.0	95.0

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10
2878	773	31-12-2014	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential, Rural and other Areas	14.0	16.0	94.0

2879 rows × 10 columns

In [130]:

```
new_data.describe()
```

Out [130]:

	Stn Code	SO2	NO2	RSPM/PM10
count	2879.000000	2879.000000	2879.000000	2879.000000
mean	475.750261	11.503138	22.136776	62.494261
std	277.675577	5.042039	7.112576	31.346938
min	38.000000	2.000000	5.000000	12.000000
25%	238.000000	8.000000	17.000000	41.000000
50%	366.000000	12.000000	22.000000	55.000000
75%	764.000000	15.000000	25.000000	78.000000
max	773.000000	49.000000	71.000000	269.000000

In [131]:

```
new_data.head()
```

Out [131]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10
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
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
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
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
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

Data Pre-proccessing

In [132]:

```
cleandata=new_data.isnull().sum()
```

In [133]:

```
cleandata
```

Out [133]:

```
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                               0
NO2                               0
RSPM/PM10                         0
dtype: int64
```

In [134]:

```
new_data.describe()
```

Out [134]:

	Stn Code	SO2	NO2	RSPM/PM10
count	2879.000000	2879.000000	2879.000000	2879.000000
mean	475.750261	11.503138	22.136776	62.494261
std	277.675577	5.042039	7.112576	31.346938
min	38.000000	2.000000	5.000000	12.000000
25%	238.000000	8.000000	17.000000	41.000000
50%	366.000000	12.000000	22.000000	55.000000
75%	764.000000	15.000000	25.000000	78.000000
max	773.000000	49.000000	71.000000	269.000000

In [135]:
```
new_data.columns
```

Out [135]:
```
Index(['Stn Code', 'Sampling Date', 'State', 'City/Town/Village/Area',
       'Location of Monitoring Station', 'Agency', 'Type of Location', 'SO2',
       'NO2', 'RSPM/PM10'],
      dtype='object')
```

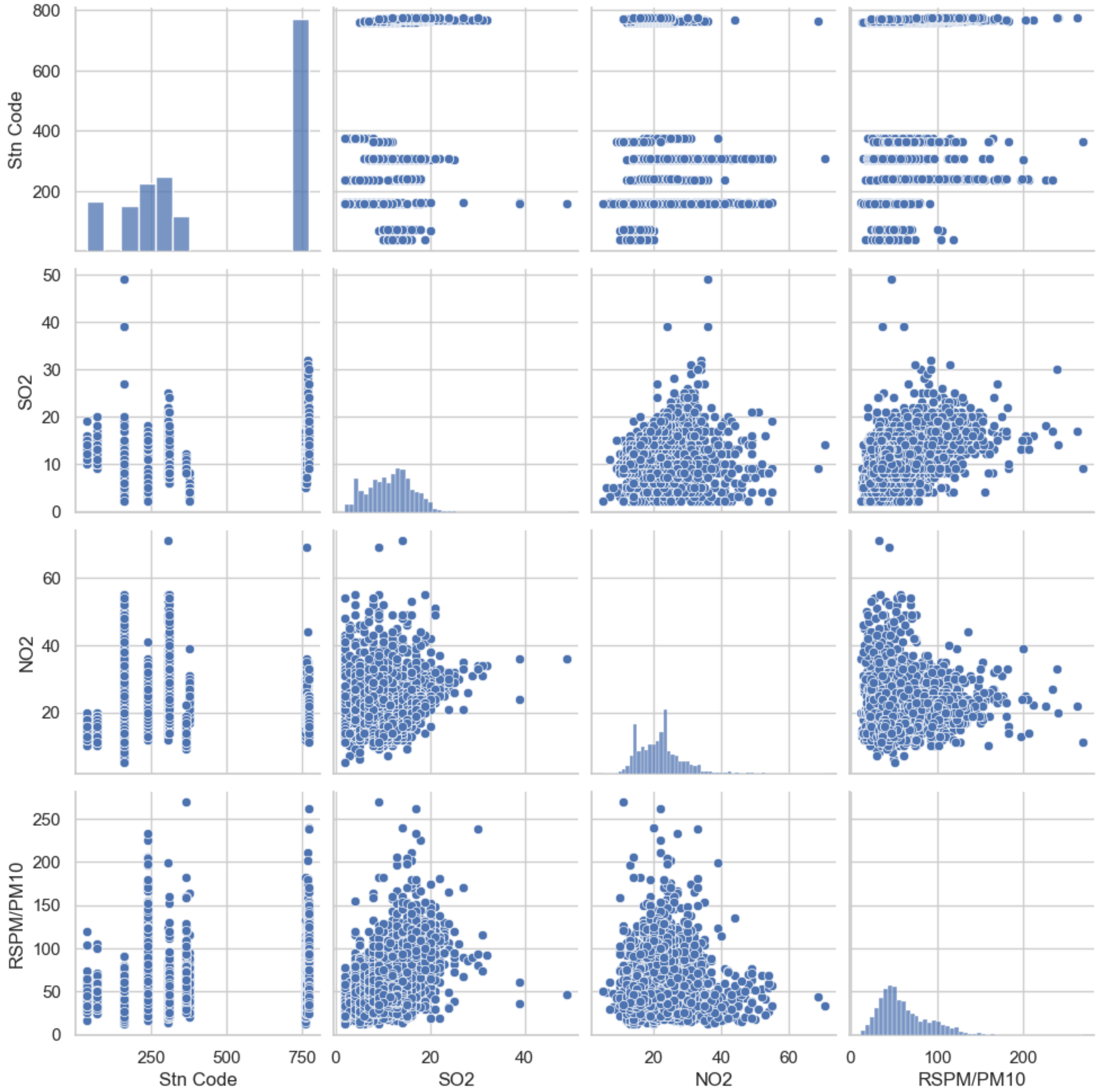
In [136]:
```
new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2879 entries, 0 to 2878
Data columns (total 10 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                             2879 non-null   float64
 8   NO2                             2879 non-null   float64
 9   RSPM/PM10                       2879 non-null   float64
dtypes: float64(3), int64(1), object(6)
memory usage: 225.1+ KB
```

In [137]:
```
sns.pairplot(data=new_data)
```

```
C:\ProgramData\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```

Out [137]:
```
<seaborn.axisgrid.PairGrid at 0x17277cc3090>
```

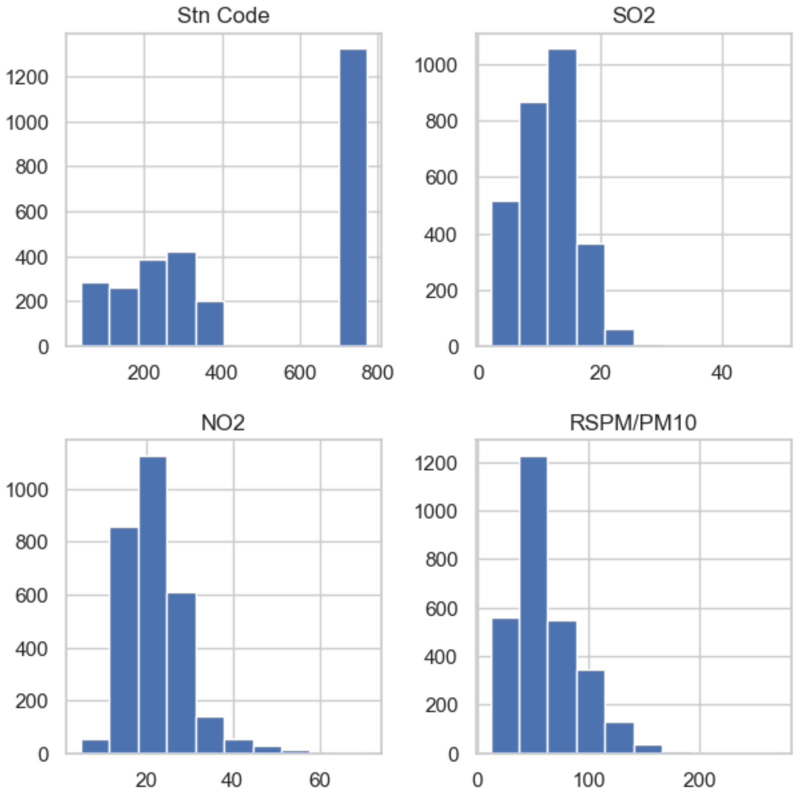


In [138]:
```
new_data.isnull().sum()
```

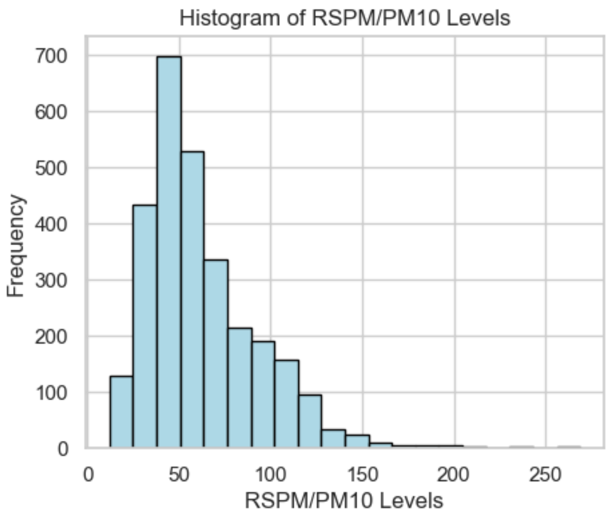
Out [138]:
```
Stn Code                          0
Sampling Date                     0
State                             0
City/Town/Village/Area            0
Location of Monitoring Station    0
Agency                            0
Type of Location                  0
SO2                               0
NO2                               0
RSPM/PM10                         0
dtype: int64
```

In [139]:
```python
new_data.hist(figsize=(7,7))
```

Out [139]:
```
array([[<Axes: title={'center': 'Stn Code'}>,
        <Axes: title={'center': 'SO2'}>],
       [<Axes: title={'center': 'NO2'}>,
        <Axes: title={'center': 'RSPM/PM10'}>]], dtype=object)
```

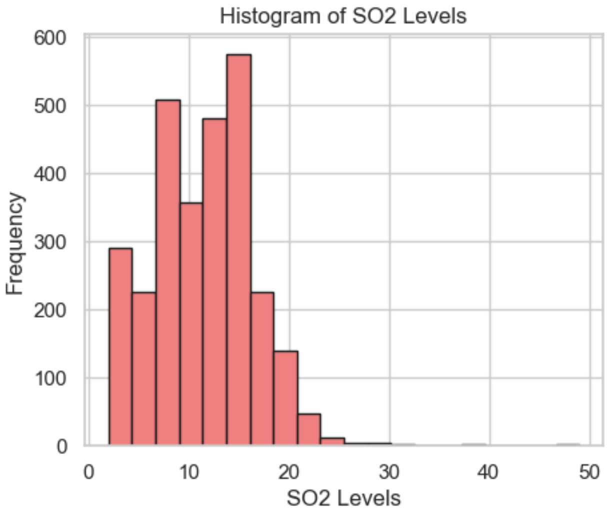


In [140]:
```python
# Histogram for RSPM/PM10

rspm_pm10_data = new_data['RSPM/PM10']
plt.figure(figsize=(5,4))  # Set the figure size
plt.hist(rspm_pm10_data, bins=20, edgecolor='black', color='lightblue')
plt.xlabel('RSPM/PM10 Levels')
plt.ylabel('Frequency')
plt.title('Histogram of RSPM/PM10 Levels')
plt.grid(True)
plt.show()
```



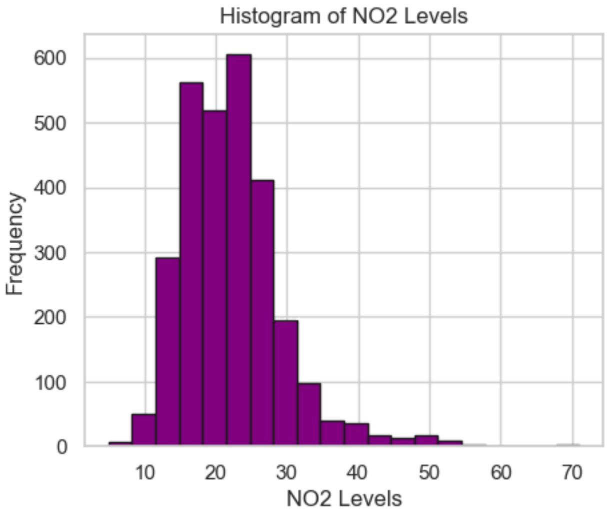
In [141]:

```
# Histogram for SO2

SO2_data = new_data['SO2']
plt.figure(figsize=(5, 4))
plt.hist(SO2_data, bins=20, edgecolor='k', color='lightcoral')
plt.xlabel('SO2 Levels')
plt.ylabel('Frequency')
plt.title('Histogram of SO2 Levels')
plt.grid(True)
plt.show()
```



In [142]:

```
# Histogram for NO2

NO2_data = new_data['NO2']
plt.figure(figsize=(5, 4))
plt.hist(NO2_data, bins=20, edgecolor='k', color='purple')
plt.xlabel('NO2 Levels')
plt.ylabel('Frequency')
plt.title('Histogram of NO2 Levels')
plt.grid(True)
plt.show()
```

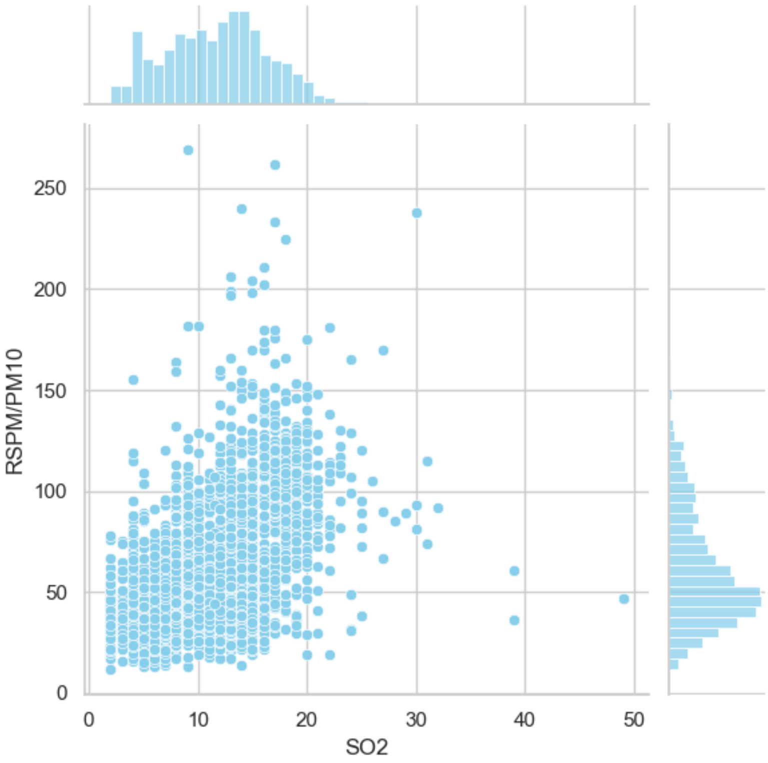


In [143]:

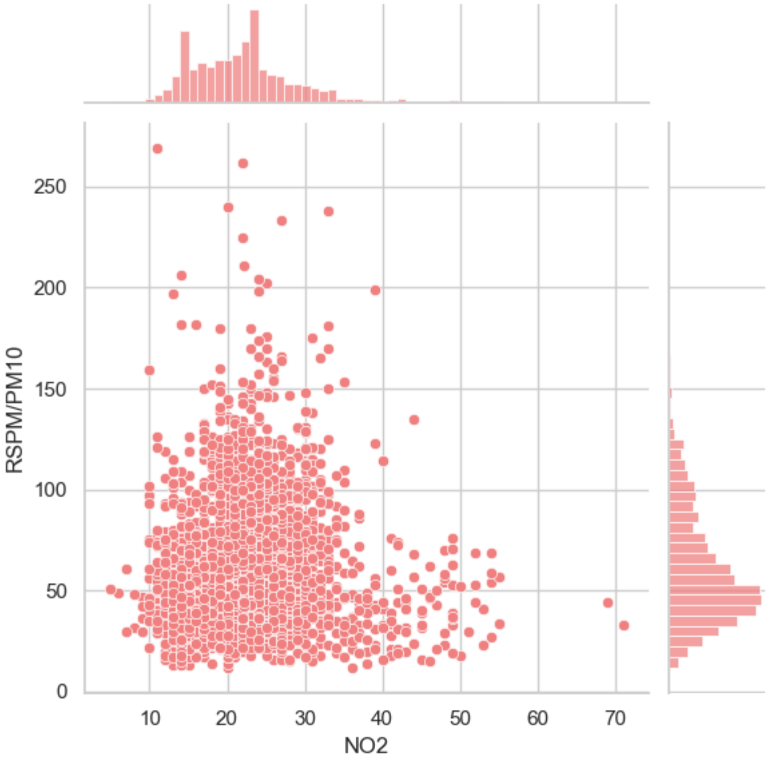
```
# Create a jointplot for SO2 vs. RSPM/PM10

sns.set(style="whitegrid")
plt.figure(figsize=(2,1))
sns.jointplot(x='SO2', y='RSPM/PM10', data=new_data, kind='scatter', color='skyblue')


# Create a jointplot for NO2 vs. RSPM/PM10
sns.set(style="whitegrid")
plt.figure(figsize=(2,1))
```

```
sns.jointplot(x='NO2', y='RSPM/PM10', data=new_data, kind='scatter', color='lightcoral')
plt.show()
```

```
<Figure size 200x100 with 0 Axes>
```

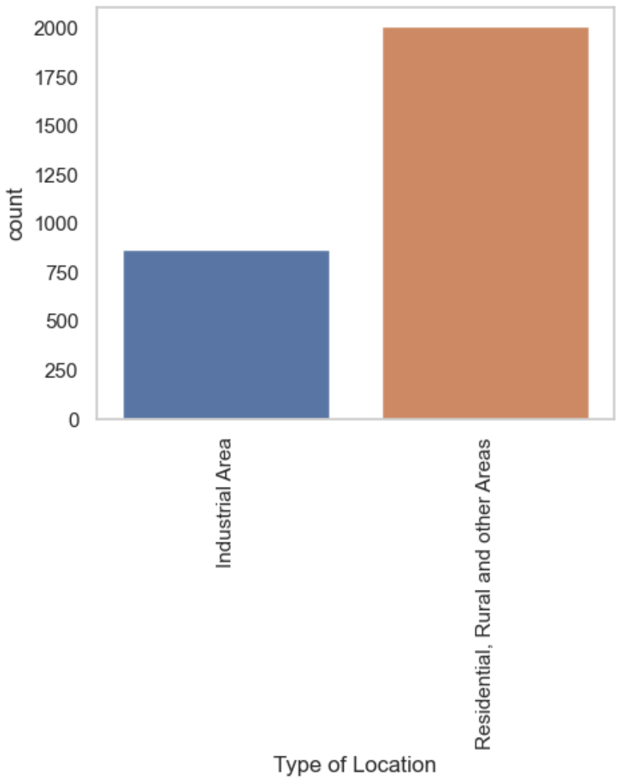


```
<Figure size 200x100 with 0 Axes>
```



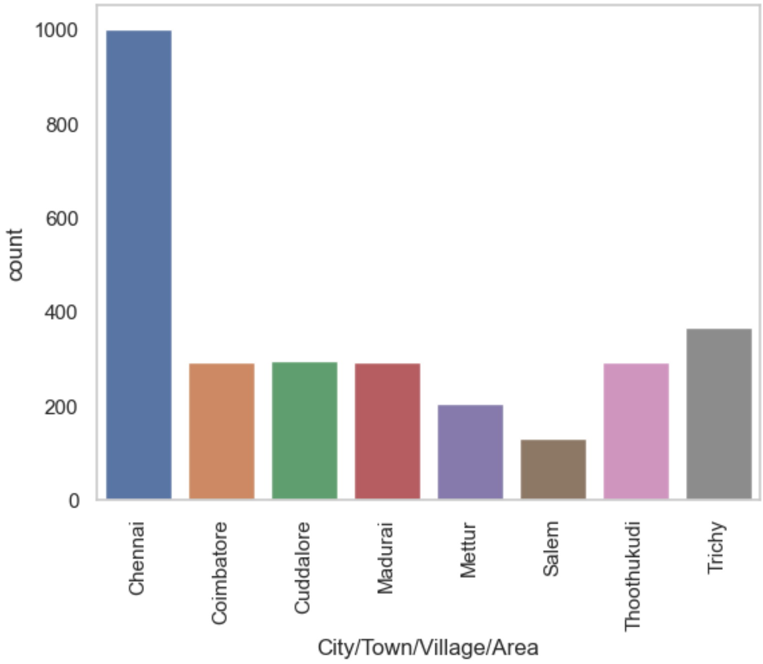
In [144]:

```
plt.figure(figsize=(5,4))
typ=sns.countplot(x ="Type of Location",data = new_data)
typ.set_xticklabels(typ.get_xticklabels(), rotation=90);
plt.grid(False)
```



In [145]:

```
datacount =sns.countplot(x ="City/Town/Village/Area",data = new_data);
datacount.set_xticklabels(datacount.get_xticklabels(), rotation=90);
plt.grid(False)
```

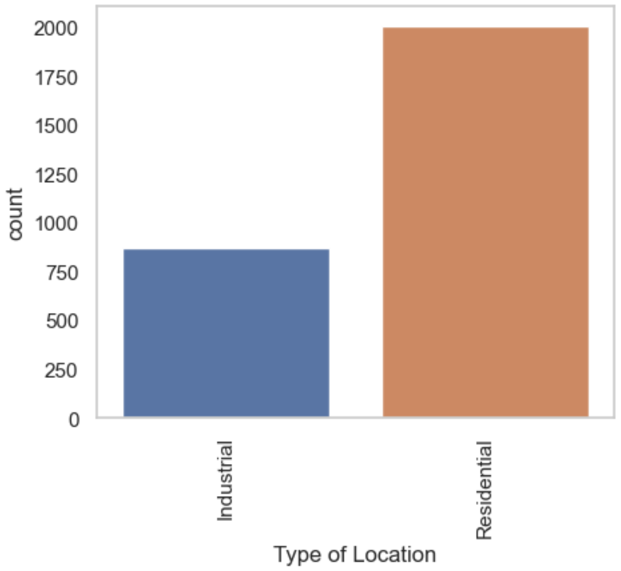


In [146]:

```
new_data['Type of Location'].replace("Industrial Areas","Industrial",inplace=True)
new_data['Type of Location'].replace("Industrial Area","Industrial",inplace=True)
new_data['Type of Location'].replace("Residential and others","Residential",inplace=True)
new_data['Type of Location'].replace("Residential, Rural and other Areas","Residential",inplace=True)
```

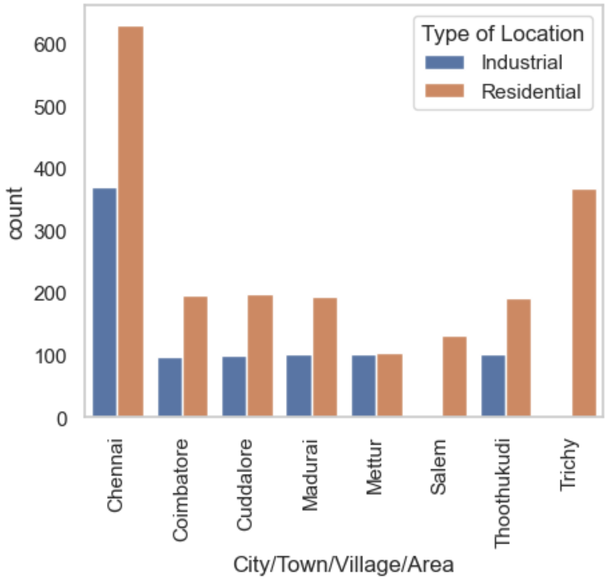
In [147]:

```
plt.figure(figsize=(5,4))
typ=sns.countplot(x ="Type of Location",data = new_data)
typ.set_xticklabels(typ.get_xticklabels(), rotation=90);
plt.grid(False)
```



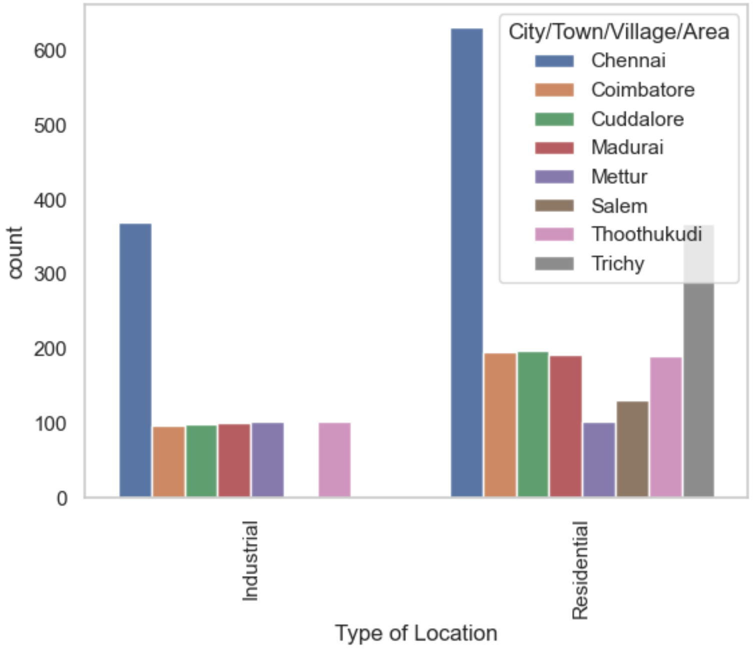
In [148]:

```
plt.figure(figsize=(5,4))
datacount_ty =sns.countplot(x ="City/Town/Village/Area",hue = 'Type of Location',data = new_data);
datacount_ty.set_xticklabels(datacount_ty.get_xticklabels(), rotation=90);
plt.grid(False)
plt.show()
```



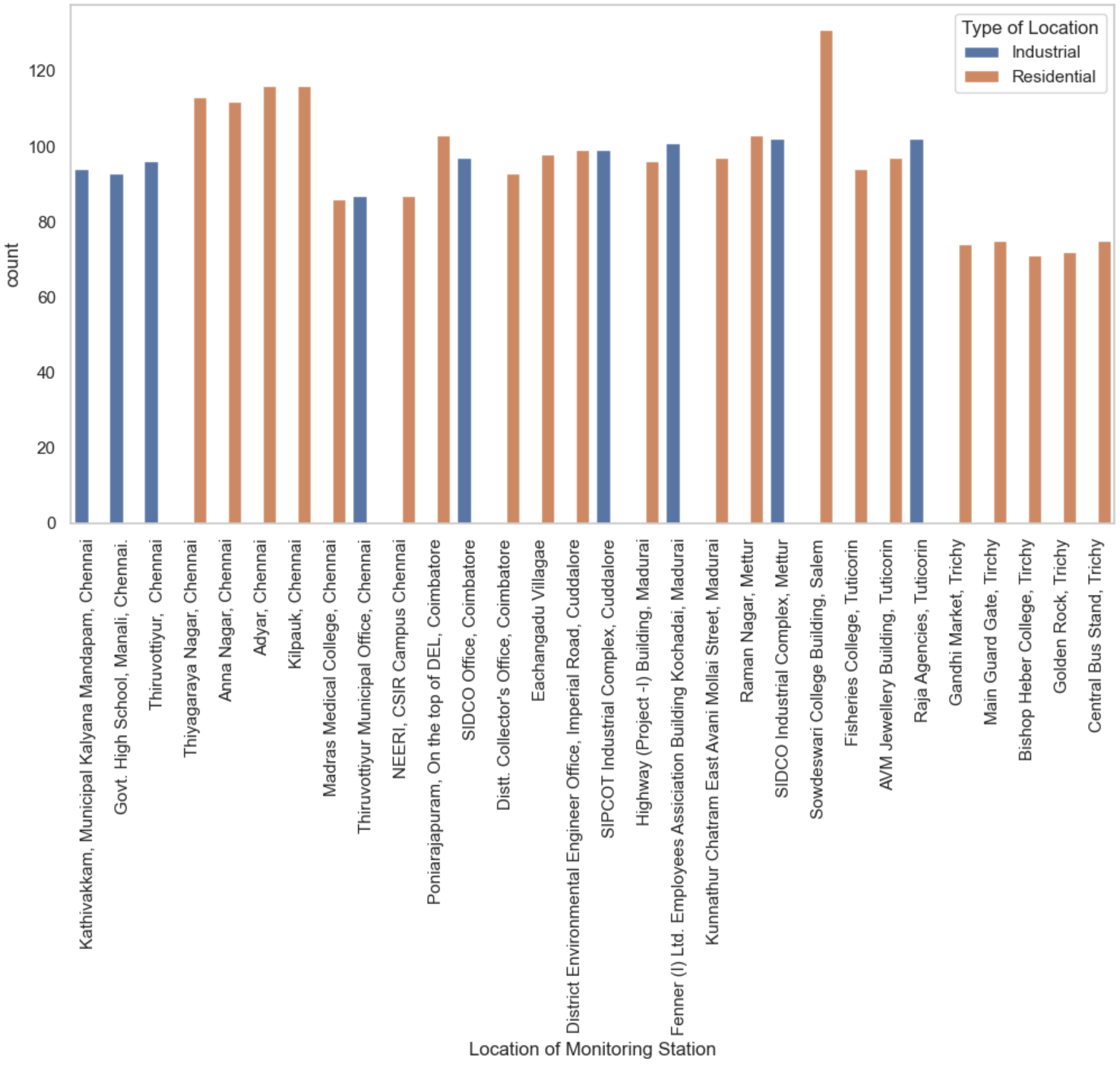
In [149]:

```
datacount_ty =sns.countplot(x ="Type of Location",hue = 'City/Town/Village/Area',data = new_data);
datacount_ty.set_xticklabels(datacount_ty.get_xticklabels(), rotation=90);
plt.grid(False)
```



In [150]:

```
plt.figure(figsize=(12,6))
datacount_ty =sns.countplot(x ="Location of Monitoring Station",hue = 'Type of Location',data = new_data);
datacount_ty.set_xticklabels(datacount_ty.get_xticklabels(), rotation=90);
plt.grid(False)
plt.show()
```



In [151]:

```
#br graph using plotly
import plotly.express as px
px.bar(new_data.head(500), x = 'SO2', y = 'Location of Monitoring Station',
       color = 'Location of Monitoring Station', height = 500, hover_data = ['SO2', 'Location of Monitoring
```

In [152]:

```
#br graph using plotly
import plotly.express as px
px.bar(new_data.head(500), x = 'NO2', y = 'Location of Monitoring Station',
       color = 'Location of Monitoring Station', height = 500, hover_data = ['NO2', 'Location of Monitoring
```

In [153]:

```
#br graph using plotly
import plotly.express as px
px.bar(new_data.head(500), x = 'RSPM/PM10', y = 'Location of Monitoring Station',
       color = 'Location of Monitoring Station', height = 500, hover_data = ['RSPM/PM10', 'Location of Monit
```

In [154]:

```
#Time series
px.bar(new_data.head(30), x = 'SO2', y = 'RSPM/PM10',
       color = 'SO2',orientation ='h',  height = 500,
       hover_data = ['SO2', 'RSPM/PM10'])
```

In [155]:

```
#Time series
px.bar(new_data.head(30), x = 'NO2', y = 'RSPM/PM10',
       color = 'NO2',orientation ='h',  height = 500,
       hover_data = ['NO2', 'RSPM/PM10'])
```

In [156]:

```
px.bar(new_data.sort_values(by='RSPM/PM10').tail(15), x = 'RSPM/PM10', y = 'Sampling Date',
       color = 'RSPM/PM10',title="Time Series Comparison of RSPM/PM10 Levels", orientation ='h',  height = 5
       hover_data = ['Sampling Date', 'RSPM/PM10'])
```

In [157]:

```
px.bar(new_data.sort_values(by='SO2').tail(15), x = 'SO2', y = 'Sampling Date',
       color = 'SO2',title="Time Series Comparison of SO2 Levels", orientation ='h',  height = 500,
       hover_data = ['Sampling Date', 'SO2'])
```

In [158]:

```
px.bar(new_data.sort_values(by='NO2').tail(15), x = 'SO2', y = 'Sampling Date',
       color = 'NO2',title="Time Series Comparison of NO2 Levels", orientation ='h',  height = 500,
       hover_data = ['Sampling Date', 'NO2'])
```

In [159]:

```
fig=plt.figure(figsize=(45,30))
fig=px.pie(new_data,values='SO2', names='City/Town/Village/Area'
            ,title="SO2 Emissions in City/Town/Village/Area",
           color_discrete_sequence=px.colors.sequential.RdBu,hole=.4)
fig.update_traces(textposition='inside')
fig.update_layout(uniformtext_minsize=12, uniformtext_mode='hide')
```

```
<Figure size 4500x3000 with 0 Axes>
```

In [160]:

```
fig=plt.figure(figsize=(45,30))
fig=px.pie(new_data,values='NO2', names='City/Town/Village/Area'
            ,title="NO2 Emissions in City/Town/Village/Area",
           color_discrete_sequence=px.colors.sequential.RdBu,hole=.4)
```

```
fig.update_traces(textposition='inside')
fig.update_layout(uniformtext_minsize=12, uniformtext_mode='hide')
```

```
<Figure size 4500x3000 with 0 Axes>
```

In [161]:

```
fig=plt.figure(figsize=(45,30))
fig=px.pie(new_data,values='RSPM/PM10', names='City/Town/Village/Area'
            ,title="RSPM/PM10 Emissions in City/Town/Village/Area",
           color_discrete_sequence=px.colors.sequential.RdBu,hole=.4)
fig.update_traces(textposition='inside')
fig.update_layout(uniformtext_minsize=12, uniformtext_mode='hide')
```

```
<Figure size 4500x3000 with 0 Axes>
```

In [162]:

```
new_data.head()
```

Out [162]:

	Stn Code	Sampling Date	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10
0	38	01-02-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial	11.0	17.0	55.0
1	38	01-07-2014	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial	13.0	17.0	45.0

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

In [163]:

```
loc = pd.pivot_table(new_data, values=['SO2','NO2','RSPM/PM10'],index='City/Town/Village/Area') # Aggfunc: c
loc
```

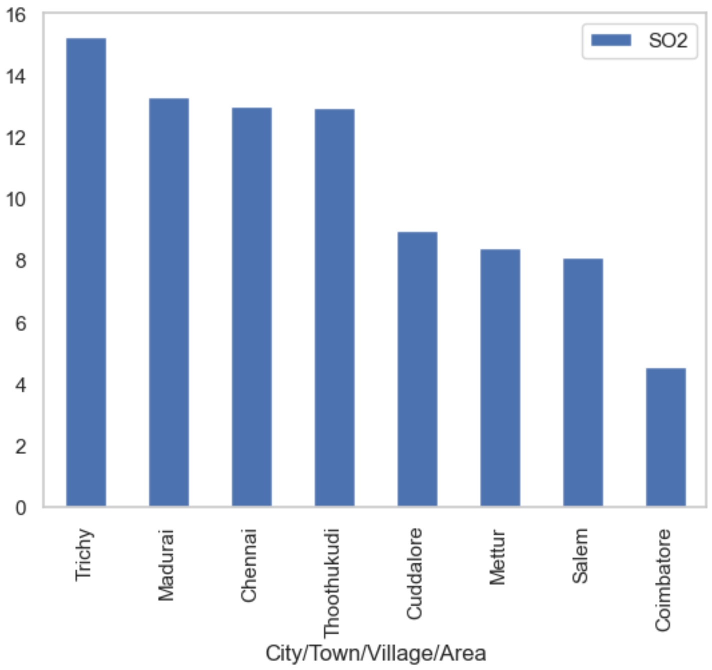
Out [163]:

	NO2	RSPM/PM10	SO2
City/Town/Village/Area			
Chennai	22.088684	58.998000	13.009509
Coimbatore	25.314460	49.353184	4.564857
Cuddalore	19.727276	61.881757	8.983129
Madurai	25.768707	45.724490	13.319728
Mettur	23.185366	52.721951	8.429268
Salem	28.664122	62.954198	8.114504
Thoothukudi	18.536770	83.387352	12.979544
Trichy	18.723189	85.054496	15.262968

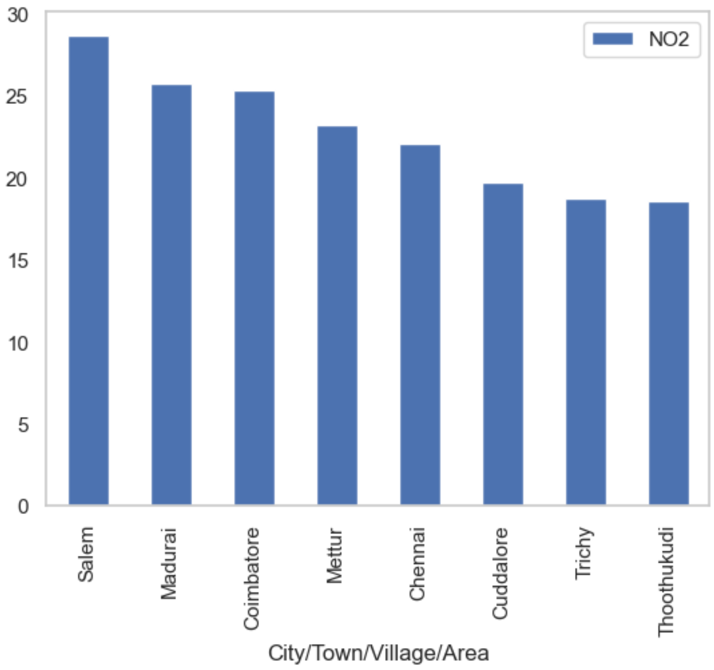
In [164]:

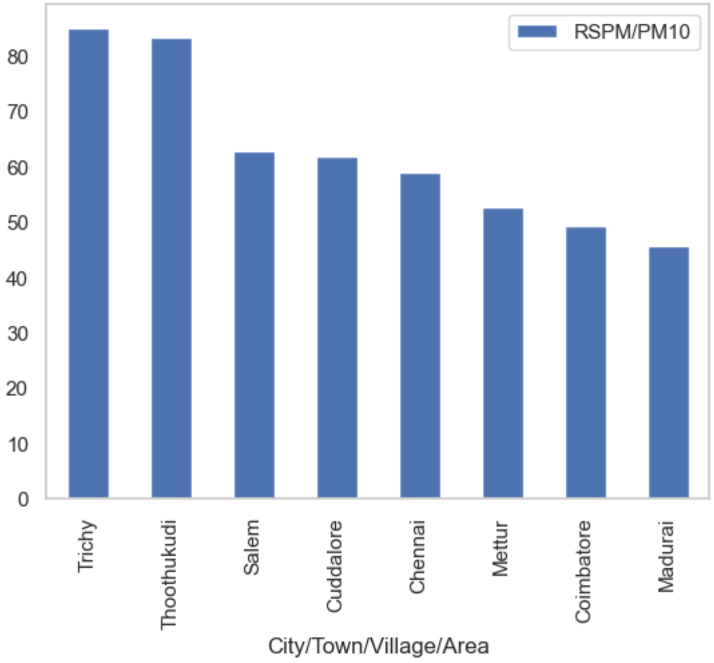
```
plt.figure(figsize=(1,1))
maxSO2 = loc.sort_values(by='SO2',ascending=False)
maxSO2.loc[:,['SO2']].head(10).plot(kind='bar');
plt.grid(False)
plt.show()

plt.figure(figsize=(1,1))
maxNO2 = loc.sort_values(by='NO2',ascending=False,)
maxNO2.loc[:,['NO2']].head(10).plot(kind='bar');
plt.grid(False)
plt.show()

maxRSPM_PM10 = loc.sort_values(by='RSPM/PM10',ascending=False);
maxRSPM_PM10.loc[:,['RSPM/PM10']].head(10).plot(kind='bar');
plt.grid(False)
plt.show()
```

```
<Figure size 100x100 with 0 Axes>
```



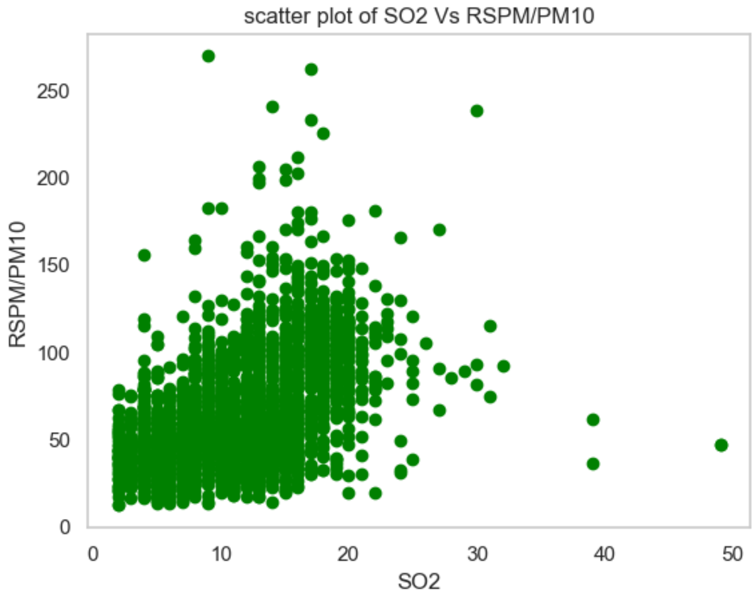
```
<Figure size 100x100 with 0 Axes>
```

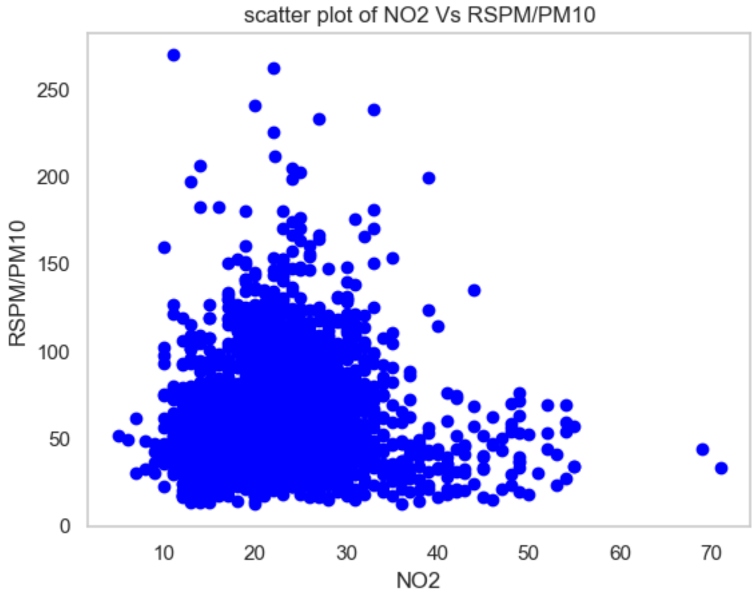




In [165]:

```
fig=plt.figure()
plt.scatter(new_data['SO2'],new_data['RSPM/PM10'], color ='green')
plt.xlabel("SO2")
plt.ylabel("RSPM/PM10")
plt.title("scatter plot of SO2 Vs RSPM/PM10")
plt.grid(False)
plt.show()

fig=plt.figure()
plt.scatter(new_data['NO2'],new_data['RSPM/PM10'], color ='blue')
plt.xlabel("NO2")
plt.ylabel("RSPM/PM10")
plt.title("scatter plot of NO2 Vs RSPM/PM10")
plt.grid(False)
plt.show()
```





In [166]:

```
# Convert the string to float

one_hot_encoded_data = pd.get_dummies(data=new_data, columns=['Sampling Date'])
```

In [167]:

```
one_hot_encoded_data
```

Out [167]:

	Stn Code	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	Sampling Date_01-02-2014	...	Sampling Date_30-08-2014	Sam Dat 09
0	38	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial	11.0	17.0	55.0	True	...	False	Fals
1	38	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial	13.0	17.0	45.0	False	...	False	Fals
2	38	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial	12.0	18.0	50.0	False	...	False	Fals
3	38	Tamil Nadu	Chennai	Kathivakkam, Municipal Kalyana Mandapam, Chennai	Tamilnadu State Pollution Control Board	Industrial	15.0	16.0	46.0	False	...	False	Fals
4	38	Tamil Nadu	Chennai	Kathivakkam, Municipal	Tamilnadu State	Industrial	13.0	14.0	42.0	False	...	False	Fals

	Stn Code	State	City/Town/Village/Area	Location of Monitoring Station	Agency	Type of Location	SO2	NO2	RSPM/PM10	Sampling Date_01-02-2014	...	Sampling Date_30-08-2014	Sam Dat 09
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2874	773	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential	15.0	18.0	102.0	False	...	False	Fals
2875	773	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential	12.0	14.0	91.0	False	...	False	Fals
2876	773	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential	19.0	22.0	100.0	False	...	False	Fals
2877	773	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential	15.0	17.0	95.0	False	...	False	Fals
2878	773	Tamil Nadu	Trichy	Central Bus Stand, Trichy	Tamilnadu State Pollution Control Board	Residential	14.0	16.0	94.0	False	...	False	Fals

2879 rows × 311 columns

In [168]:

```
# Convert the string to float

one_hot_encoded_data = pd.get_dummies(data=new_data, columns=['Sampling Date','State','City/Town/Village/Are
```

In [169]:

```
one_hot_encoded_data
```

Out [169]:

	Stn Code	SO2	NO2	RSPM/PM10	Sampling Date_01-02-2014	Sampling Date_01-03-2014	Sampling Date_01-04-2014	Sampling Date_01-06-2014	Sampling Date_01-07-2014	Sampling Date_01-08-2014	...	Location of Monitoring Station_SIDCO Office, Coimbatore	Location o Monitorin Station_SIPCO Industri Comple Cuddalor
0	38	11.0	17.0	55.0	True	False	False	False	False	False	...	False	False
1	38	13.0	17.0	45.0	False	False	False	False	True	False	...	False	False
2	38	12.0	18.0	50.0	False	False	False	False	False	False	...	False	False
3	38	15.0	16.0	46.0	False	False	False	False	False	False	...	False	False
4	38	13.0	14.0	42.0	False	False	False	False	False	False	...	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
2874	773	15.0	18.0	102.0	False	False	False	False	False	False	...	False	False
2875	773	12.0	14.0	91.0	False	False	False	False	False	False	...	False	False
2876	773	19.0	22.0	100.0	False	False	False	False	False	False	...	False	False
2877	773	15.0	17.0	95.0	False	False	False	False	False	False	...	False	False
2878	773	14.0	16.0	94.0	False	False	False	False	False	False	...	False	False

2879 rows × 349 columns

In [170]:

```
one_hot_encoded_data.corr()
```

Out [170]:

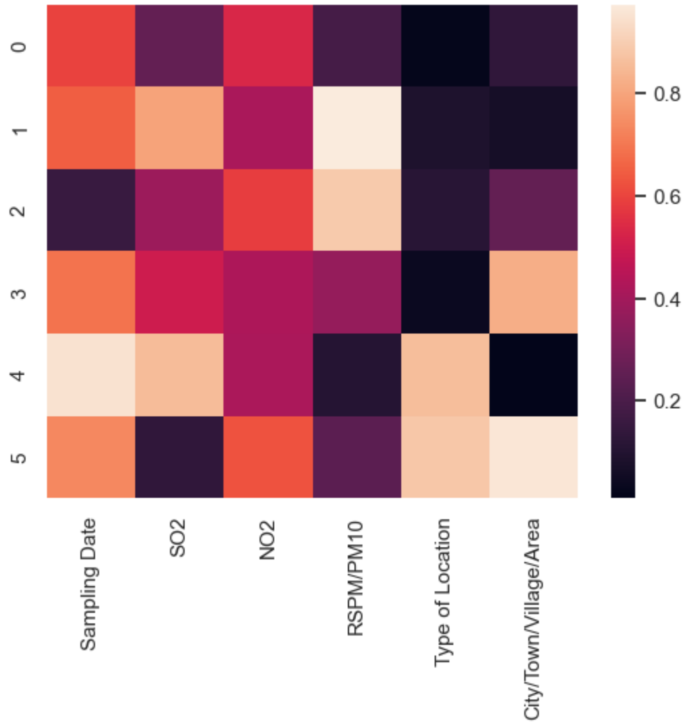
	Stn Code	SO2	NO2	RSPM/PM10	Sampling Date_01-02-2014	Sampling Date_01-03-2014	Sampling Date_01-04-2014	Sampling Date_01-06-2014	Sampling Date_01-07-2014	Sampling Date_01-08-2014	.
Stn Code	1.000000	0.263054	-0.043161	0.336019	0.005126	0.000471	-0.007457	0.010641	-0.003000	-0.004777	.
SO2	0.263054	1.000000	0.078234	0.444140	0.009454	-0.000556	-0.018521	0.015039	0.009721	-0.001342	.
NO2	-0.043161	0.078234	1.000000	0.067921	-0.024942	0.019103	-0.004651	-0.001191	0.042416	-0.010294	.
RSPM/PM10	0.336019	0.444140	0.067921	1.000000	0.008544	0.013980	0.017561	-0.000078	0.024559	0.021694	.
Sampling Date_01-02-2014	0.005126	0.009454	-0.024942	0.008544	1.000000	-0.004171	-0.002310	-0.003835	-0.004171	-0.003269	.

	Stn Code	SO2	NO2	RSPM/PM10	Sampling Date_01-02-2014	Sampling Date_01-03-2014	Sampling Date_01-04-2014	Sampling Date_01-06-2014	Sampling Date_01-07-2014	Sampling Date_01-08-2014	.
...	...	...	...	...	...	...	...	...	...	...	.
Location of Monitoring Station_Thiyagaraya Nagar, Chennai	0.211311	0.294549	0.172215	0.256885	-0.012518	0.013066	-0.007539	-0.012518	0.013066	-0.010669	.
Agency_National Environmental Engineering Research Institute	-0.358340	-0.265946	0.189082	-0.246745	-0.019513	0.014930	-0.011752	-0.019513	0.033005	-0.016632	.
Agency_Tamilnadu State Pollution Control Board	0.358340	0.265946	-0.189082	0.246745	0.019513	-0.014930	0.011752	0.019513	-0.033005	0.016632	.
Type of Location_Industrial	-0.413289	-0.086657	-0.042851	-0.230044	-0.016275	0.000756	0.016034	-0.028532	0.012034	0.008327	.
Type of Location_Residential	0.413289	0.086657	0.042851	0.230044	0.016275	-0.000756	-0.016034	0.028532	-0.012034	-0.008327	.

349 rows × 349 columns

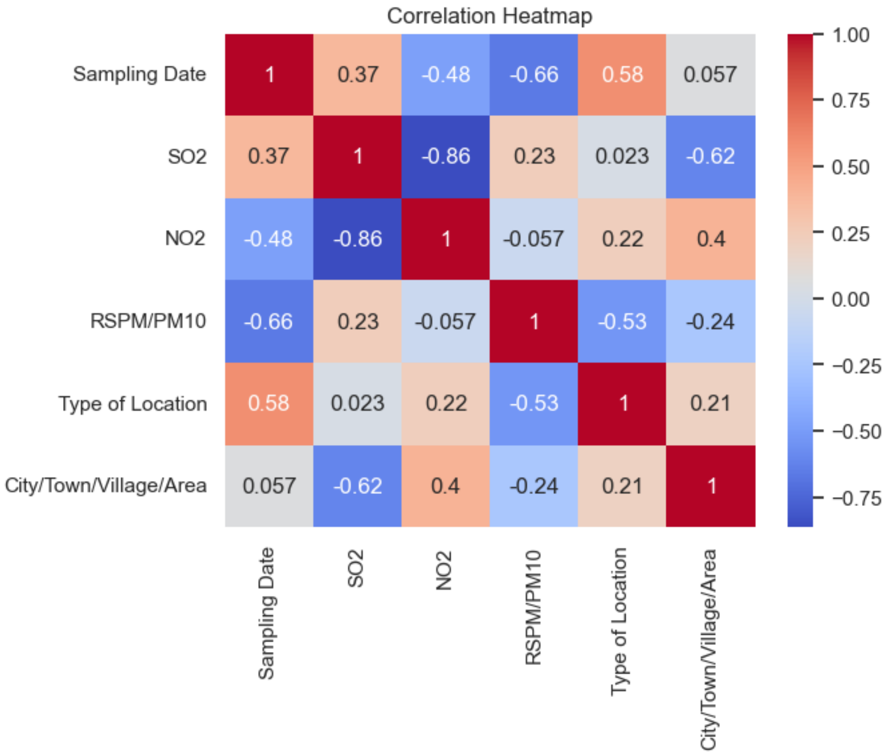
In [171]:

```
xlabs = ["Sampling Date", "SO2", "NO2", "RSPM/PM10",
         "Type of Location", "City/Town/Village/Area"]
ylabs = ["Sampling Date", "SO2", "NO2", "RSPM/PM10",
         "Type of Location", "City/Town/Village/Area"]
one_hot_encoded_data = pd.DataFrame(np.random.random((6,6)), columns=["Sampling Date","SO2","NO2","RSPM/PM1

# Default heatmap

p1 = sns.heatmap(one_hot_encoded_data)
```



In [172]:

```
#Heatmap

correlation_matrix = one_hot_encoded_data.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



In [173]:

```
sorted_data = one_hot_encoded_data.sort_values(by=['Sampling Date','SO2','NO2','RSPM/PM10'], ascending=True)
```

In [174]:

```
sorted_data
```

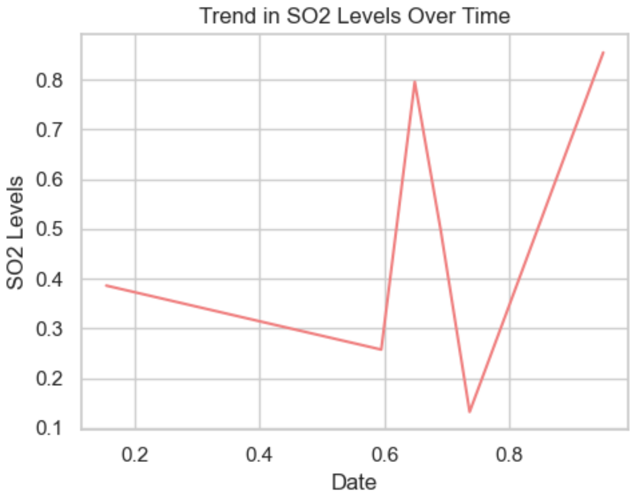
Out [174]:

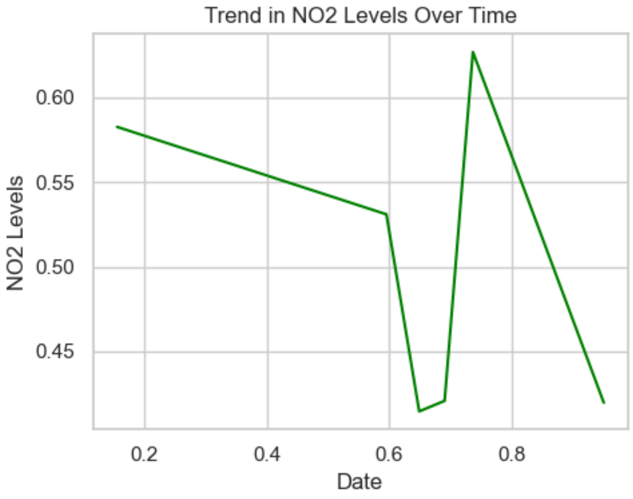
```
   Sampling Date       SO2       NO2  RSPM/PM10  Type of Location  City/Town/Village/Area
2       0.154934  0.386195  0.582729   0.887238          0.114952                0.257591
0       0.594932  0.257594  0.531061   0.184740          0.018519                0.133460
1       0.648539  0.795445  0.414623   0.973575          0.087590                0.065091
3       0.690076  0.497327  0.420869   0.369925          0.039192                0.820846
5       0.736292  0.132414  0.626984   0.236556          0.882963                0.960921
4       0.950121  0.854591  0.419746   0.106897          0.859297                0.010103
```

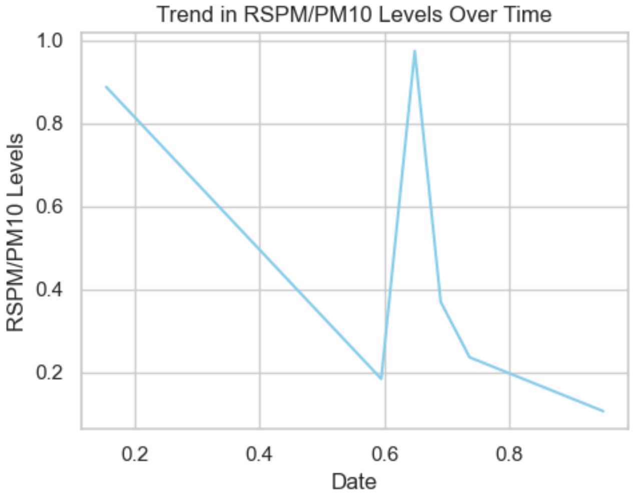
In [175]:

```
#lineplot for SO2
dates = sorted_data['Sampling Date']
so2_data = sorted_data['SO2']
plt.figure(figsize=(5,4))
plt.plot(dates, so2_data, label='SO2', color='lightcoral', linestyle='-')
plt.title('Trend in SO2 Levels Over Time')
plt.xlabel('Date')
plt.ylabel('SO2 Levels')
plt.xticks(rotation=0)
plt.grid(True)
plt.tight_layout()
plt.show()

#lineplot for NO2
dates = sorted_data['Sampling Date']
no2_data = sorted_data['NO2']
plt.figure(figsize=(5,4))
plt.plot(dates, no2_data, label='NO2', color='green', linestyle='-')
plt.title('Trend in NO2 Levels Over Time')
plt.xlabel('Date')
plt.ylabel('NO2 Levels')
plt.xticks(rotation=0)
plt.grid(True)
plt.tight_layout()
plt.show()


#lineplot for RSPM/PM10
dates = sorted_data['Sampling Date']
rspm_pm10_data = sorted_data['RSPM/PM10']
```

```
plt.figure(figsize=(5,4))
plt.plot(dates, rspm_pm10_data, label='RSPM/PM10', color='skyblue', linestyle='-')
plt.title('Trend in RSPM/PM10 Levels Over Time')
plt.xlabel('Date')
plt.ylabel('RSPM/PM10 Levels')
plt.xticks(rotation=0)
plt.grid(True)
plt.tight_layout()
plt.show()
```

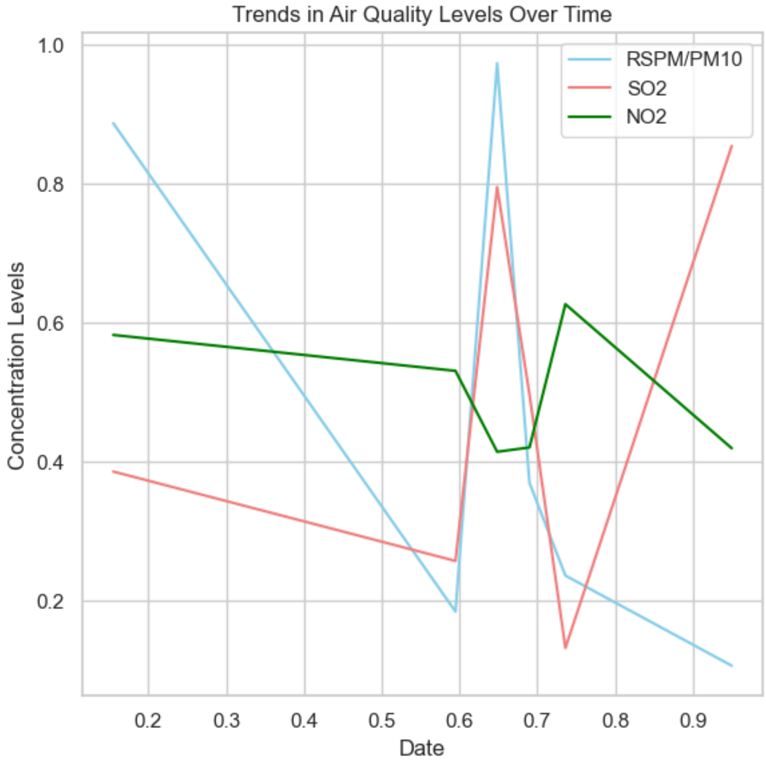






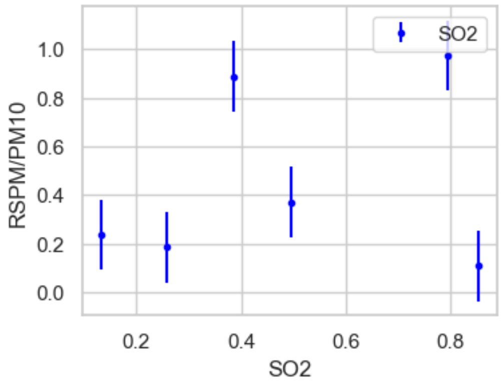
In [176]:

```
dates = sorted_data['Sampling Date']
rspm_pm10_data = sorted_data['RSPM/PM10']
so2_data = sorted_data['SO2']
no2_data = sorted_data['NO2']

plt.figure(figsize=(6, 6))
plt.plot(dates, rspm_pm10_data, label='RSPM/PM10', color='skyblue', linestyle='-')
plt.plot(dates, so2_data, label='SO2', color='lightcoral', linestyle='-')
plt.plot(dates, no2_data, label='NO2', color='green', linestyle='-')

plt.title('Trends in Air Quality Levels Over Time')
plt.xlabel('Date')
plt.ylabel('Concentration Levels')

plt.legend()
plt.xticks(rotation=0)
plt.grid(True)
plt.tight_layout()
plt.show()
```



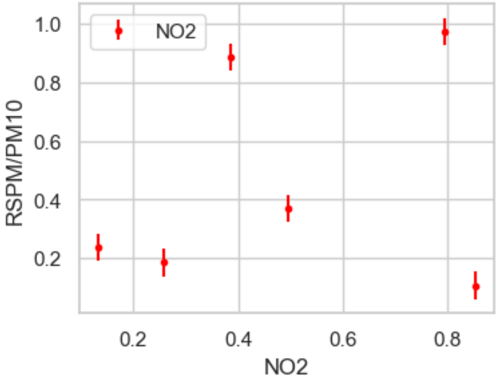
In [177]:

```
plt.figure(figsize=(4,3))
# Calculate standard deviation of data(SO2 and RSPM/PM10)
std = one_hot_encoded_data['SO2'].std()
plt.errorbar(one_hot_encoded_data['SO2'], one_hot_encoded_data['RSPM/PM10'], yerr=std/2, fmt='.',label='SO2'
plt.xlabel('SO2')
plt.ylabel('RSPM/PM10')
plt.legend()
plt.show()
```



In [178]:

```
# Calculate standard deviation of data(NO2 and RSPM/PM10)
plt.figure(figsize=(4,3))
std = one_hot_encoded_data['NO2'].std()

# Add error bars to plot
plt.errorbar(one_hot_encoded_data['SO2'], one_hot_encoded_data['RSPM/PM10'], yerr=std/2, fmt='.',label='NO2'
plt.xlabel('NO2')
plt.ylabel('RSPM/PM10')
plt.legend()
plt.show()
```



In [179]:

```
X_train.dtypes
```

Out [179]:

```
SO2    int64
NO2    int64
dtype: object
```

In [180]:

```
from sklearn.model_selection import train_test_split
from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Split your data into features (X) and target (y)
X = air_quality_data[['SO2', 'NO2']]
y = air_quality_data['RSPM/PM10']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train an SVM regression model
svm_model = SVR(kernel='linear')  # You can choose different kernels (linear, rbf, etc.)
svm_model.fit(X_train, y_train)

# Make predictions
y_pred = svm_model.predict(X_test)

# Calculate accuracy metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print the accuracy metrics
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R-squared (R^2): {r2:.2f}")
```

```
Mean Absolute Error (MAE): 0.20
Mean Squared Error (MSE): 0.04
R-squared (R^2): 0.99
```

In [181]:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Load your air quality data into a pandas DataFrame


# Split your data into features (X) and target (y)
X = one_hot_encoded_data[['SO2', 'NO2']]
y = one_hot_encoded_data['RSPM/PM10']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Initialize and train a Decision Tree regression model
decision_tree_model = DecisionTreeRegressor(random_state=42)
decision_tree_model.fit(X_train, y_train)
```

```
# Make predictions
y_pred = decision_tree_model.predict(X_test)

# Calculate accuracy metrics
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print the accuracy metrics
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"R-squared (R^2): {r2:.2f}")
```

```
Mean Absolute Error (MAE): 0.78
Mean Squared Error (MSE): 0.62
R-squared (R^2): -3.00
```

In [ ]:

In [ ]: